Enhancing Global Optimization Performance of Arithmetic Optimization Algorithm with a Modified Population Initialization Scheme

Tin Chang Ting[1], Hameedur Rahman[2], Meng Choung Chiong[1], Mohamed Khan Afthab Ahamed Khan[1], Cik Suhana Hassan[1], Farah Adilah Jamaludin[1], Sew Sun Tiang[1*], Wei Hong Lim[1*]

[1]*Faculty of Engineering, Technology and Built Environment, UCSI University, Kuala Lumpur 56000, Malaysia*

[2]*Faculty of Computing and Artificial Intelligence, Air University, Islamabad Capital Territory 44000, Pakistan*

Email: 1002058096@ucsiuniversity.edu.my, rhameedur@gmail.com, ChiongMC@ucsiuniversity.edu.my, mohamedkhan@ucsiuniversity.edu.my, suhana@ucsiuniversity.edu.my, FarahAdilah@ucsiuniversity.edu.my, tiangss@ucsiuniversity.edu.my, limwh@ucsiuniversity.edu.my

Abstract

Arithmetic Optimization Algorithm (AOA) is widely used to solve global optimization problems. However, it often faces premature convergence challenges in complex optimization scenarios. A key factor affecting AOA's performance is the solution quality of the initial population. The conventional initialization scheme, despite its prevalence, lacks reliability in ensuring high-quality solutions due to inherent stochastic processes. To address this issue, we propose a modified initialization scheme that improves initial population quality by integrating chaotic maps and oppositional-based learning. Through extensive simulation studies, we demonstrate that the enhanced AOA, equipped with this new initialization scheme, exhibits superior performance in solving a range of benchmark functions with improved accuracy.

Keywords: Arithmetic optimization algorithm, Population initialization, Chaotic map, Oppositional-based learning

1. Introduction

Optimization is pivotal in real-life engineering design, where it seeks the most effective solutions while accommodating various stakeholders' criteria. Accurate problem definition and modeling are essential for resolving these design challenges. This process includes establishing clear objectives, identifying both technical and non-technical constraints, and optimizing decision variables. Nonetheless, real-world engineering design problems frequently present complexity and challenges due to factors like high dimensionality, numerous constraints, conflicting objectives, and data uncertainty.

In the era of Industrial Revolution 4.0 (IR 4.0), engineering systems have become more intricate, often involving non-differentiable, nonlinear, multimodal, and non-continuous functions. While traditional optimization methods such as Newton's method are prevalent, they face limitations in addressing these complex, real-world engineering design challenges. These methods often depend heavily on initial solutions and are typically suited to specific problem types, limiting their scalability for diverse, complicated optimization tasks [1]. Most traditional methods also rely on gradient information to identify optimal solutions, which is not always feasible in real-world scenarios that frequently present as black box functions [2]. Furthermore, the limited global search capability of traditional optimization methods increases the risk of premature convergence.

Recognizing the limitations of traditional optimization methods, there is an urgent need in the Industrial Revolution 4.0 era to develop more intelligent and robust optimization algorithms. These algorithms must be capable of providing efficient solutions to a broad spectrum of increasingly complex optimization problems. Metaheuristic Search Algorithms (MSAs) have emerged as effective solutions, drawing on search mechanisms inspired by natural phenomena. Existing MSAs fall into four categories based on their natural inspirations [3]: evolutionary algorithms, swarm intelligence, human-based algorithms, and physics-based algorithms. Compared to traditional methods, MSAs offer several advantages, including potent global search capabilities, straightforward implementation, and enhanced scalability. They exploit the unique strengths of their respective inspirations, enabling them to effectively address a variety of complex optimization challenges as outlined in [4], [5], [6], [7], [8], [9], [10], [11], [12], [13].

The Arithmetic Optimization Algorithm (AOA) [14], introduced in 2021, is a physics-based algorithm with search mechanisms inspired by the four basic arithmetic operations: division, multiplication, addition, and subtraction. These operations vary in their exploration and exploitation strengths, essential for solving optimization problems. Like other MSAs, achieving a proper balance between exploration and exploitation is crucial for enhancing AOA's performance. Consequently, several enhancement schemes, including hybridization

[15], [16], [17], the introduction of new learning mechanisms [18], [19], [20] and others, have been developed to improve the balance in AOA variants. Since its introduction, both the original AOA and its enhanced variants have been successfully applied to a wide array of engineering optimization problems [15], [16], [19], [21], [22], [23], [24], [25].

The quality of the initial population is a key factor in determining the performance of MSAs. Despite numerous variants of the AOA being introduced in recent years, many still rely on conventional methods to randomly generate their initial populations. This conventional approach, while simple to implement, does not intelligently leverage environmental information around the solution regions during initialization [3]. As a result, it often mistakenly places some solutions in local or suboptimal regions, leading to premature convergence. Additionally, if initial solutions are far from the global optimum, the algorithm's convergence speed may be compromised. These drawbacks can significantly impact the robustness and effectiveness of AOA in addressing complex real-world optimization challenges.

This paper presents the Multi-Chaotic Dynamic Oppositional Learning (MCDOL) module as an enhancement to the AOA's population initialization scheme, culminating in a new variant named MCDOL-AOA. The MDCOL module is designed to produce an initial population with enhanced solution quality, both in terms of accuracy and convergence rate, by harnessing the strengths of multiple chaotic maps and the dynamic oppositional-based learning (DOL) mechanism. Notably, the diverse ergodic properties of various chaotic maps are employed for a comprehensive solution space search, enhancing population diversity and mitigating the risk of initializing solutions in local optima. Additionally, the inclusion of DOL in MCDOL module promotes more rapid global optimum identification by expanding the exploration of solution regions. The effectiveness of MCDOL-AOA is evaluated through benchmarking against the original AOA across a range of benchmark functions with varying characteristics.

2. Methodology

2.1. *Proposed MCDOL module*

In this subsection, we detail how the proposed MCDOL module generates an initial population of superior quality. The MCDOL module leverages multiple chaotic maps to create a diverse set of initial solutions, forming a chaotic population. This approach contrasts with conventional initialization schemes, which often result in poorly distributed initial solutions. The deterministic nature of chaotic maps, coupled with their ability to exhibit stochastic behavior through ergodic properties, allows for a more comprehensive exploration of the solution space. Unlike some prior studies [26], [27], [28] that rely on a single chaotic map, this paper explores the benefits of using multiple chaotic maps for population initialization. We hypothesize that each map's unique ergodic characteristics can enhance the algorithm's performance for specific optimization problems. Therefore, the synergistic integration of multiple chaotic maps promises to significantly improve the algorithm's robustness in addressing various complex problems.

The MCDOL module incorporates five distinct chaotic maps — Circle, Logistic, Piecewise, Sine, and Tent — for generating a chaotic population. Each chaotic map is randomly chosen to generate different dimensions of the initial solutions within this population. To facilitate this, define $\boldsymbol{\vartheta_t}$ as the output of a chaotic variable at the t-th iteration, where $\boldsymbol{t = 1, \dots, T}$. The population initialization in the MCDOL module, using these multiple chaotic maps, follows five criteria: (a) the Circle map is used if $\mathbf{0} \le \boldsymbol{\vartheta_0} < \mathbf{0.2}$, (b) the Logistic map for $\mathbf{0.2} \le \boldsymbol{\vartheta_0} < \mathbf{0.4}$, (c) the Piecewise map for $\mathbf{0.4} \le \boldsymbol{\vartheta_0} < \mathbf{0.6}$, (d) the Sine map for $\mathbf{0.6} \le \boldsymbol{\vartheta_0} < \mathbf{0.8}$, and (e) the Tent map for $\mathbf{0.8} \le \boldsymbol{\vartheta_0} \le \mathbf{1.0}$. The specific equations for these five chaotic maps, Circle, Logistic, Piecewise, Sine, and Tent, are detailed in Eqs. (1) to (5).

$$\vartheta_{t+1} = \mathrm{mod}\left(\vartheta_t + 0.2 - \left(\frac{0.5}{2\pi}\right)\sin(2\pi\vartheta_t), 1\right) \tag{1}$$

$$\vartheta_{t+1} = 4\,\vartheta_t(1-\vartheta_t) \tag{2}$$

$$\vartheta_{t+1} = \begin{cases} \dfrac{\vartheta_t}{P}, & 0 \le \vartheta_t < P \\ \dfrac{\vartheta_t - P}{0.5 - P}, & P \le \vartheta_t < 0.5 \\ \dfrac{1 - P - \vartheta_t}{0.5 - P}, & 0.5 \le \vartheta_t < 1 - P \\ \dfrac{1 - \vartheta_t}{P}, & 1 - P \le \vartheta_t < 1 \end{cases} \tag{3}$$

$$\vartheta_{t+1} = \sin(\pi\vartheta_t) \tag{4}$$

$$\vartheta_{t+1} = \begin{cases} \dfrac{\vartheta_t}{0.7} & \vartheta_t < 0.7 \\ \dfrac{10}{3}(1-\vartheta_t) & \vartheta_t \ge 0.7 \end{cases} \tag{5}$$

Let $\boldsymbol{X_d^L}$ and $\boldsymbol{X_d^U}$ denote the lower and upper bounds of the d-th dimensional decision variable, respectively, for $\boldsymbol{d = 1, \dots, D}$. At the final iteration $\boldsymbol{t = T}$, a chaotic value $\boldsymbol{\vartheta_T}$ is generated from a randomly selected chaotic map. This value initializes the d-th dimension of each n-th chaotic solution, as outlined in Eq. (6). The resulting chaotic population, denoted as $\mathbf{P^C} = [\boldsymbol{X_1^C}, \dots, \boldsymbol{X_n^C}, \dots \boldsymbol{X_N^C}]$, encompasses all solution members formed using multiple chaotic maps.

$$X_{n,d}^C = X_d^L + \vartheta_T\,(X_d^U - X_d^L) \tag{6}$$

While various chaotic maps exhibit differing levels of robustness to local optima, they may still generate initial solutions distant from the global optimum, potentially slowing the algorithm's convergence. To address this issue, a DOL operator is applied to the chaotic population $\mathbf{P^C}$, generating opposite solutions for each n-th chaotic solution. This DOL operator is expected to broaden the initial population's coverage of the solution space, increasing the likelihood of identifying fitter solutions. The opposite solution for each dimension, denoted as $\boldsymbol{X^O_{n,d}}$, is determined using Eq. (7), corresponding to $\boldsymbol{X^C_{n,d}}$, with $\boldsymbol{r_1}, \boldsymbol{r_2} \in [\mathbf{0}, \mathbf{1}]$. Consequently, this generates an opposition population, $\mathbf{P^O} = [\boldsymbol{X^O_1}, \dots, \boldsymbol{X^O_n}, \dots \boldsymbol{X^O_N}]$.

$$X^O_{n,d} = X^C_{n,d} + r_1 \left[r_2 (X^U_d + X^L_d - X^C_{n,d}) - X^C_{n,d} \right] \quad (7)$$

The two populations, one generated using multiple chaotic maps ($\mathbf{P^C}$) and the other using the DOL operator ($\mathbf{P^O}$), are merged to form a combined population set $\mathbf{P^C} \cup \mathbf{P^O}$ with a total size of $2N$. The fitness of each solution member in this merged population set is evaluated according to the predefined objective function. These solution members are then ordered based on their fitness values, from the best to worst performing ones. The top N solution members from this ordered population set $\mathbf{P^C} \cup \mathbf{P^O}$ are chosen as the initial population for the proposed MCDOL-AOA algorithm, represented as $\mathbf{P} = [\boldsymbol{X_1}, \dots, \boldsymbol{X_n}, \dots, \boldsymbol{X_N}]$.

2.2. *Iterative Search Processes of MCDOL-AOA*

After generating an initial population $\mathbf{P}$ of superior quality using the MCDOL module, each n-th solution in MCDOL-AOA is iteratively updated through search mechanisms similar to those in the original AOA.

During each iteration, the Math Optimizer Accelerated (MOA) function value is adjusted in MCDOL-AOA to alternate between exploration and exploitation phases:

$$MOA(C_{Iter}) = Min + C_{Iter} \left(\frac{Max - Min}{M_{Iter}} \right) \quad (8)$$

where C_{Iter} and M_{Iter} represent the current and maximum iteration counts; Min and Max are the minimum and maximum values of MOA. Simultaneously, the Math Optimizer Probability (MP) function, which dictates the search range for each solution and is influenced by the critical parameter θ for exploitation efficiency, is updated as:

$$MP(C_{Iter}) = 1 - \left(\frac{C_{Iter}}{M_{Iter}} \right)^{1/\theta} \quad (9)$$

A random number $\boldsymbol{rand1}$ determines the search strategy (exploration or exploitation) at each iteration for updating the d-th dimension of the n-th solution, $\boldsymbol{X_{n,d}}$. During the exploration phase ($\boldsymbol{rand1} > \boldsymbol{MOA}$), either the Multiplication or Division operator is selected:

$$X_{n,d}(C_{Iter}+1) = \begin{cases} best_d \div (MP + \varepsilon) \times [(X^U_d - X^L_d)\mu + X^L_d], & rand2 < 0.5 \\ best_d \times MP \times [(X^U_d - X^L_d)\mu + X^L_d], & \text{Otherwise} \end{cases} \quad (10)$$

where $rand2$ is a random number between 0 and 1; $best_d$ denotes the d-th dimension of the current best solution; ε is a small positive number to prevent division by zero, and μ is a control parameter.

In the exploitation phase ($rand1 \leq MOA$), either the Addition or Subtraction operator is used to update $X_{n,d}$:

$$X_{n,d}(C_{Iter}+1) = \begin{cases} best_d - MP \times [(X^U_d - X^L_d)\mu + X^L_d], & rand2 < 0.5 \\ best_d + MP \times [(X^U_d - X^L_d)\mu + X^L_d], & \text{Otherwise} \end{cases} \quad (11)$$

As illustrated in Fig. 1, MCDOL-AOA continues this iterative search, following Eqs. (8) to (11), until pre-set termination criteria are fulfilled. Upon completion, optimal decision variables in the best solution are decoded to solve the specific optimization problems.

MCDOL-AOA for Global Optimization

Inputs: D, N, M_{Iter}, T, Max, Min, θ
Initialize $\mathbf{P^C} \leftarrow \emptyset$, $\mathbf{P^O} \leftarrow \emptyset$ and $C_{Iter} \leftarrow 0$;
for each n-th solution **do**
  for each d-th dimension **do**
    Randomly initialize $\vartheta_t \in [0,1]$, where $t = 0$;
    if $0 \leq \vartheta_0 < 0.2$ **then**
      Select Circle map in Eq. (1);
    elseif $\mathbf{0.2} \leq \boldsymbol{\vartheta_0} < \mathbf{0.4}$ **then**
      Select Logistic map in Eq. (2);
    elseif $\mathbf{0}.4 \leq \vartheta_0 < 0.6$ **then**
      Select Piecewise map in Eq. (3);
    elseif $\mathbf{0}.6 \leq \vartheta_0 < 0.8$ **then**
      Select Sine map in Eq. (4);
    elseif $0.8 \leq \vartheta_0 \leq 1.0$ **then**
      Select Tent map in Eq. (5);
    end if
    while $t \leq T$ **do**
      Update ϑ_t with the selected chaotic map;
      $t \leftarrow t + 1$;
    end while
    Generate $X^C_{n,d}$ using Eq. (6);
    Generate $X^O_{n,d}$ using Eq. (7);
  end for
  Update $\mathbf{P^C} \leftarrow \mathbf{P^C} \cup X^C_{n,d}$ and $\mathbf{P^O} \leftarrow \mathbf{P^O} \cup X^O_{n,d}$;
end for
Merge the two population sets as $\mathbf{P^C} \cup \mathbf{P^O}$;
Evaluate the fitness values of all solutions stored within $\mathbf{P^C} \cup \mathbf{P^O}$;
Rearrange the solutions stored within $\mathbf{P^C} \cup \mathbf{P^O}$ from best to worst based on their fitness values;
Select the top N solutions from the sorted $\mathbf{P^C} \cup \mathbf{P^O}$ as the initial population, i.e., $\mathbf{P} = [X_1, \dots, X_n, \dots, X_N]$.
Assign the first solution of $\mathbf{P}$ and its fitness as *best* and $f(best)$, respectively;
while $C_{Iter} \leq M_{Iter}$ **do**
  Update *MOA* and *MP* with Eqs. (8) and (9);
  for each n-th solution **do**
    if $rand1 > MOA$ **then** /**Exploration**/
      Update $X_{n,d}(C_{Iter}+1)$ with Eq. (10);
    else /**Exploitation**/
      Update $X_{n,d}(C_{Iter}+1)$ with Eq. (11);
    end if
    Fitness evaluation of $X_n(C_{Iter}+1)$;
    Update the X_n, $f(X_n)$, $best$ and $f(best)$, with greedy selection method;
  end for
  $C_{Iter} \leftarrow C_{Iter} + 1$;
end while
Output: $best$ and $f(best)$

Fig.1 Workflow of proposed MCDOL-AOA in solving the global optimization problems.

3. Results and Discussions

3.1. *Simulations settings*

In this section, we compare the performance of MCDOL-AOA with the original AOA using 23 benchmark functions, each with distinct characteristics as outlined in [17]. Functions F1 to F7 are scalable unimodal functions, while F8 to F13 are scalable multimodal functions, all set at a dimension size of D = 100. Functions F14 to F23, on the other hand, are fixed-dimension multimodal functions with D ranging from 2 to 6. Both MCDOL-AOA and the original AOA were implemented on MATLAB 2021a, running on a personal computer equipped with an Intel® Core™ i7-HQ CPU at 2.50 GHz and 16 GB RAM. For both algorithms, the population size and maximum number of iterations are set at 30 and 1000, respectively.

3.2. *Performance analysis*

The evaluation of MCDOL-AOA's performance, and its comparison with the original AOA across all 23 benchmark functions, is detailed in Table 1. We employ two performance metrics: mean error (E_{mean}) and standard deviation (SD), to assess the algorithms' accuracy and consistency in solving these functions. Lower values of E_{mean} and SD are preferable, indicating the algorithm's consistent and accurate resolution of the given benchmark functions.

Table 1. Performance comparison of MCDOL-AOA and original AOA using 23 benchmark functions with different characteristics.

Fun	Original AOA		MCDOL-AOA	
	E_{mean}	SD	E_{mean}	SD
F1	2.048e-04	5.662e-05	**1.938e-04**	5.371e-05
F2	0.0112	0.0115	**0.0107**	0.0011
F3	**0.0915**	0.0229	0.0934	0.0202
F4	0.0523	0.0076	**0.0516**	0.0089
F5	98.090	0.0899	**98.060**	0.1158
F6	15.131	0.7747	**14.930**	0.8160
F7	**2.609e-05**	3.060e-05	3.574e-05	3.667e-05
F8	**-1.53e+04**	646.631	**-1.53e+04**	631.509
F9	6.244e-05	1.478e-05	**6.195e-05**	1.080e-05
F10	**0.0016**	1.452e-04	**0.0016**	1.767e-04
F11	**0.0085**	0.0249	0.0176	0.0519
F12	0.891	0.0655	**0.0888**	0.0740
F13	9.906	0.0030	**9.904**	0.0030
F14	**9.864**	4.317	11.192	**2.979**
F15	0.0055	0.0125	**0.0028**	0.0075
F16	**-1.032**	**5.375e-12**	**-1.032**	8.016e-12
F17	**0.398**	5.620e-07	**0.398**	**3.516e-07**
F18	18.3	28.080	**16.5**	25.318
F19	**-3.863**	5.441e-05	**-3.863**	**4.105e-06**
F20	**-3.294**	**0.0511**	-3.274	0.0592
F21	**-8.554**	**2.522**	-8.141	2.979
F22	-8.077	3.155	**-8.350**	3.267
F23	-8.590	3.094	**-8.751**	3.091

The results in Table 1 showcase MCDOL-AOA's impressive accuracy, outperforming or matching the original AOA in 16 of the 23 benchmark functions, as evidenced by lower E_{mean} values. In the scalable unimodal function's category (F1 to F7), MCDOL-AOA surpasses the original AOA in 5 of the 7 functions based on E_{mean}. For the 6 scalable multimodal functions (F8 to F13), MCDOL-AOA demonstrates superior E_{mean} values in 4 functions (F8, F9, F12, and F13) and equals the original AOA in one function (F10). When dealing with the fixed-dimension multimodal functions (F14 to F23), MCDOL-AOA outperforms the original AOA in 4 functions (F15, F18, F22, and F23) and equals its performance in 3 other functions (F16, F17, and F19).

Analysis of the simulation results in Table 1 indicates that MCDOL-AOA significantly outperforms the original AOA in solving larger-scale benchmark functions (F1 to F13) with D = 100, evidenced by lower E_{mean} values in 9 out of 13 functions. However, the performance of MCDOL-AOA is relatively on par with the original AOA in fixed-dimension problems (F14 to F23) with smaller dimensions (D = 2, 3, 4, and 6). These findings suggest that the MCDOL module in MCDOL-AOA effectively produces initial populations with superior fitness and diversity compared to the original AOA, yielding enhanced optimization results. The benefits of the MCDOL module are particularly notable in larger dimensional sizes (e.g., D = 100), as utilized in this study. The multiple chaotic maps' non-repetitive and ergodic properties within the MCDOL module facilitate a more exhaustive exploration of the solution space across various problem types, thereby minimizing the risk of local optima entrapment and premature convergence. Additionally, the DOL mechanism within the MCDOL module contributes to faster algorithmic convergence by broadening solution space exploration through the generation of opposite solutions from those initiated by the chaotic maps.

4. Conclusion

In this paper, we introduce an enhanced version of the AOA, termed MCDOL-AOA, designed to solve complex global optimization problems with improved accuracy. The novelty of MCDOL-AOA resides in its integration of multiple chaotic maps and DOL mechanisms into a modified initialization scheme, the MCDOL module. This module aims to generate an initial population of superior quality, focusing on fitness and diversity. Simulation results demonstrate that MCDOL-AOA, benefiting from the enhanced initial population quality provided by the MCDOL module, surpasses the original AOA in solving 23 distinct benchmark functions. Notably, the performance gains of MCDOL-AOA are more pronounced in solving functions with larger dimensions, specifically D = 100. This underscores the efficacy of the multiple chaotic maps and DOL mechanisms within the MCDOL module in boosting the algorithm's robustness against premature convergence and enhancing its convergence speed. Future work will explore the application of MCDOL-AOA to real-world engineering optimization problems, including machine learning model training and scheduling optimization, to assess its practicality and effectiveness.

Acknowledgements

This work was supported by UCSI University's Research Excellence & Innovation Grant (REIG) with project code of REIG-FETBE-2022/038 and Billion Prima Sdn. Bhd.'s Industry Research Grant with project code of IND-FETBE-2023/006.

References

1. J. C. Nash, On best practice optimization methods in R, Journal of Statistical Software 60(2), 2014, pp. 1-14.
2. B. Petrongolo and A. P. Chirstopher, Looking into the black box: A survey of the matching function, Journal of Economic Literature 39(2), 2001, pp. 390-431.
3. M. F. Ahamad, N. A. Mat Isa, W. H. Lim and K. M. Ang, Differential evolution with modified initialization scheme using chaotic oppositional based learning strategy, Alexandria Engineering Journal 61(12), 2022, pp. 11835-11858.
4. E. S. Ghith and F. A. A. Tolba, Real-time implementation of an enhanced proportional-integral-derivative controller based on sparrow search algorithm for micro-robotics system, IAES International Journal of Artificial Intelligence 11(4), 2022, pp. 1395-1404.
5. E. S. Ghith and F. A. A. Tolba, LabVIEW implementation of tuning PID controller using advanced control optimization techniques for micro-robotics system. International Journal of Mechanical Engineering and Robotics Research 11(9), 2022, pp. 653-661.
6. E. S. Ghith and F. A. Tolba, Real-time implementation of an enhanced PID controller based on marine predator algorithm (MPA) for micro-robotics system, 2022 3rd International Conference on Artificial Intelligence, Robotics and Control (AIRC), Cairo, Egypt, 2022.
7. C. Hassan, S. Sapuan, N. Abd. Aziz and M. Z. Mohamed Yusof, Effect of chemical treatment on the tensile properties of single oil palm empty fruit bunch (OPEFB) fibre. Trends in Textile Engineering & Fashion Technology 3(2), 2018, pp. 1-7.
8. E. Natarajan, C. Hassan, C. K. Ang, M. S. Santhosh, S. Ramesh and R. Sasikumar, Modeling of multiwall carbon nanotubes reinforced natural rubber for soft robotic applications - A comprehensive presentation 46(9), 2021, pp. 3251-3258.
9. C. Hassan C. W. Yeo, B. Shari, M. S. Salit and N. Abdul Aziz, Mechanical properties of unidirectional oil palm empty fruit bunch (OPEFB) fiber reinforced epoxy composite. IOP Conference Series: Materials Science and Engineering 206(1), 2017, pp 012045.
10. L. Yao *et al.*, Demand bidding optimization for an aggregator with genetic algorithm. Energies 11(1), 2018, pp. 1-22.
11. K. M. Ang *et al.*, Modified teaching-learning-based optimization and applications in multi-response machining processes, Computers & Industrial Engineering 174, 2022, pp. 108719.
12. A. Sharma *et al.*, Identification of photovoltaic module parameters by implementing a novel teaching learning based optimization with unique exemplar generation scheme (TLBO-UEGS), Energy Report 10, 2023, pp. 1485-1506.
13. A. Sharma *et al.*, An effective method for parameter estimation of solar PV cell using grey-wolf optimization technique, International Journal of Mathematical, Engineering and Management Science 6(3), 2021, pp. 911-931.
14. L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz and A. H. Gandomi, The arithmetic optimization algorithm, Computer Methods in Applied Mechanics and Engineering 376, 2021, pp. 113609.
15. A. A. Mohamed *et al.*, A novel hybrid arithmetic optimization algorithm and salp swarm algorithm for data placement in cloud computing, Soft Computing, 27(9), 2023, pp. 5769-5780.
16. M. Issa, Enhanced arithmetic optimization algorithm for parameter estimation of PID controller, Arabian Journal for Science and Engineering, 48(2), 2023, pp. 2191-2205.
17. L. Abualigah and A. Diabat, Improved multi-core arithmetic optimization algorithm-based ensemble mutation for multidisciplinary applications, Journal of Intelligent Manufacturing 34(4), 2023, pp. 1833-1874.
18. S. B. Aydemir, A novel arithmetic optimization algorithm based on chaotic maps for global optimization, Evolutionary Intelligence 16(3), 2023, pp. 981-996.
19. W.-K. Hao, J.-S. Wang, X.-D. Li, H.-M. Song and Y.-Y. Bao, Probability distribution arithmetic optimization algorithm based on variable order penalty functions to solve combined economic emission dispatch problem, Applied Energy, 316, 2022, pp. 119061.
20. Y.-J. Zhang, Y.-F. Wang, Y.-X. Yan, J. Zhao and Z.-M. Gao, LMRAOA: An improved arithmetic optimization algorithm with multi-leader and high-speed jumping based on opposition-based learning solving engineering and numerical problems, Alexandria Engineering Journal 61(12), 2022, pp. 12367-12403.
21. E. S. Ghith and F. A. A. Tolba, Tuning PID controllers based on hybrid arithmetic optimization algorithm and artificial gorilla troop optimization for micro-robotic systems. IEEE Access 11, 2023, pp. 27138-27154.
22. S. Rajput *et al.*, A novel opposition-based arithmetic optimization algorithm for parameter extraction of PEM fuel cell. Electronics 10(2), 2021, pp. 2834.
23. K. M. Ang *et al.*, Training feedforward neural networks using arithmetic optimization algorithm for medical classification, in Advances in Intelligent Manufacturing and Mechatronics. Lecture Notes in Electrical Engineering, eds. M. A. Abdullah *et al.* Springer, Singapore, vol. 988, 2023.
24. X. Shi, X. Yu, and M. Esmaeili-Falak, Improved arithmetic optimization algorithm and its application to carbon fiber reinforced polymer-steel bond strength estimation, Composite Structures 306, 2023, pp. 116599.
25. A. Sharma *et al.*, Performance investigation of state-of-the-art metaheuristic techniques for parameter extraction of solar cells/module, Scientific Reports 13, 2023, pp. 11134.
26. Y. Wang, S. Gao, Y. Yu, Z. Wang, J. Cheng and T. Yuki, A gravitational search algorithm with chaotic neural oscillators, IEEE Access 8, 2020, pp. 25938-25948.
27. Z. Xu, S. Gao, H. Yang and Z. Lei, SCJADE: Yet another state-of-the-art differential evolution algorithm, IEEJ Transactions on Electrical and Electronic Engineering 16(4), 2021, pp. 644-646.
28. A.B. Ozer, CIDE: chaotically initialized differential evolution, Expert Systems with Applications 37(6), 2010, pp. 4632-4641.

Authors Introduction

Mr. Tin Chang Ting

He is currently pursuing the Bachelor of Mechatronics Engineering with Honours in Faculty of Engineering, Technology and Built Environment, UCSI University, Malaysia. His research interests are machine learning, deep learning, and optimization algorithm.

Dr. Hameedur Rahman

He is an Associate Professor in Faculty of Computing and AI at AIR University in Pakistan. He received his PhD in Computer Science from Universiti Kebangsaan Malaysia in 2018. His research interests are Virual/Augmented Rality, Image Processing, Data Mining, Artificial Intelligence, Natural Language Processing and CyberSecurity.

Dr. Meng Choung Chiong

He is an Assistant Professor in Faculty of Engineering at UCSI University in Malaysia. He received his PhD in Chemical Enigneering from Universiti Teknologi Malaysia. His research interests are artificial intelligence, mechatronics and alternative fuels combustion.

Dr. Mohamed Khan Afthab Ahmed Khan

He is an Assistant Professor in Faculty of Engineering at UCSI University in Malaysia. His research interests are artificial intelligence, robotics, control, medical rehabilitation and power electroncs.

Dr. Cik Suhana Hassan

She is an Assistant Professor in Faculty of Engineering at UCSI University in Malaysia. She obtained her PhD in PhD in Mechanical Engineering from Universiti Putra Malaysia. Her research interests are artifical intelligence, bio composite and finite element analysis.

Dr. Farah Adilah Binti Jamaluddin

She is an Assistant Professor in Faculty of Engineering at UCSI University in Malaysia. She obtained her PhD in PhD in Eletrical Power Engineering from Universiti Putra Malaysia. Her research interests are artifical intelligence, electrical insulation and high voltage engineering.

Dr. Sew Sun Tiang

She is an Assistant Professor in Faculty of Engineering at UCSI University in Malaysia. She received her PhD in Electrical and Electronic Engineering from Universiti Sains Malaysia in 2014. Her research interests are optimization and antenna design.

Dr. Wei Hong Lim

He is an Associate Professor in Faculty of Engineering at UCSI University in Malaysia. He received his PhD in Computational Intelligence from Universiti Sains Malaysia in 2014. His research interests are optimization and artificial intelligence.