IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 14, No. 5, October 2025, pp. 3790~3801
ISSN: 2252-8938, DOI: 10.11591/ijai.v14.i5.pp3790-3801

Arithmetic artificial bee colony optimization algorithm with flexible manipulator system

Mohd Ruzaini Hashim[1], Ahmad Fitri Mazlan[2], Mohammad Osman Tokhi[3]
[1]Department of Electrical Engineering, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[2]Master of Mechatronics Engineering, Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia
[3]School of Engineering, London South Bank University, London, United Kingdom

Article Info

Article history:

Received Dec 29, 2023
Revised Jul 24, 2025
Accepted Aug 6, 2025

Keywords:

Algorithm
Arithmetic
Artificial bee colony
Error criteria
Flexible manipulator system
Hub angle
Optimization algorithm

ABSTRACT

The artificial bee colony (ABC) algorithm, a well-known swarm intelligence-based metaheuristic inspired by the food foraging behavior of honeybees, has been widely applied to solve complex optimization problems. Despite its effectiveness, the standard ABC algorithm suffers from drawbacks such as slow convergence rates, limited balance between exploration and exploitation, and a tendency to get stuck in local optima, thereby hindering its overall performance. This study introduces an enhanced variant of the ABC algorithm, integrating the exploration strategy of the arithmetic optimization algorithm (AOA) to overcome these limitations. The enhanced algorithm is thoroughly tested on a set of benchmark functions as well as a flexible manipulator system model. Comprehensive statistical analyses are employed to evaluate and compare the performance of the improved algorithm against the original ABC. The results demonstrate that the enhanced ABC algorithm delivers superior performance in both benchmark scenarios and the flexible manipulator application.

Corresponding Author:

Mohd Ruzaini Hashim
Department of Electrical Engineering, Faculty of Electrical Technology and Engineering
Universiti Teknikal Malaysia Melaka
Hang Tuah Jaya, 76100 Durian Tunggal, Melaka, Malaysia
Email: ruzaini@utem.edu.my

1. INTRODUCTION

In recent years, metaheuristic algorithms based on swarm intelligence have emerged as powerful tools for addressing complex optimization problems, primarily due to their ease of implementation, flexibility, resilience, and computational efficiency. These algorithms are modeled on the collective behaviors exhibited by social organisms, such as food searching, nest building, and group navigation. Motivated by these biological phenomena, researchers have introduced numerous optimization methods grounded in swarm intelligence principles. Noteworthy approaches in this category include the particle swarm algorithm (PSA) [1], ant-based optimization (ABO) [2], bee-inspired search algorithm (BSA) [3], cuckoo-inspired optimization (CIO) [4], and the echolocation-based bat strategy (EBBS) [5]. While these algorithms have achieved notable success, many studies have pointed out their limitations when applied to different types of optimization problems. This observation supports the "No free lunch" theorem [6], [7], which states that no single algorithm performs optimally across all problem domains. Consequently, the field continues to evolve, with ongoing research focused on developing and refining algorithms tailored to the unique demands of specific optimization challenges, thereby enriching the landscape of available techniques.

Originally proposed by Karaboga [8] in 2005, the artificial bee colony (ABC) algorithm is modeled after the foraging patterns of honeybee colonies [9]. Its simplicity, minimal parameter tuning, and strong performance across diverse optimization problems have made it a popular choice among researchers [10]. In a comparative study involving 50 benchmark functions, Kaya *et al.* [11] demonstrated ABC's superior performance over other well-established algorithms such as the genetic algorithm (GA), differential evolution (DE), evolutionary strategies (ES), and particle swarm optimization (PSO), reporting consistently better objective values and lower standard deviations. Additional validation was provided by Khosravanian *et al.* [12], who found ABC to be more effective than harmony search (HS), ant colony optimization (ACO), and GA in optimizing oil-well designs. Similarly, Agarwal *et al.* [13] showed that ABC outperformed the firefly algorithm in solving the Rastrigin function, highlighting its faster convergence. These studies affirm ABC's robustness [11], computational efficiency [14], and reliable performance despite having relatively few control parameters. To further enhance ABC's capabilities, Lee and Hashim [15] introduced the hybrid ABC algorithm with artificial rabbit algorithm, which accelerates convergence by refining the structure of searching bee phase. Numerous enhancements since then have focused on improving the algorithm's balance between exploration and exploitation [16]. Building on these developments, the current study presents a new modification that integrates the arithmetic optimization algorithm (AOA) [17] into the ABC structure. This hybrid, termed the arithmetic artificial bee colony (AABC) algorithm, is designed to strengthen both global exploration and local exploitation efficiency.

The proposed AABC algorithm enhances the exploration capability of the standard ABC approach by embedding the search dynamics of the AOA into the employed bee phase. To strengthen exploitation, the onlooker bee phase is refined with innovative strategies, including leveraging the global best solution as a guiding reference and implementing a newly designed step-size control mechanism. The algorithm's effectiveness is thoroughly evaluated using a set of ten benchmark functions. Moreover, the AABC is applied to a flexible manipulator system (FMS) to assess its performance in regulating the hub angle within a real-world control context. A detailed comparative study between the proposed AABC and the original ABC algorithm is conducted to demonstrate the performance gains achieved through the introduced enhancements.

The structure of the paper is as follows: section 2 presents the fundamentals of the ABC algorithm. Section 3 discusses the AOA algorithm and the FMS model, followed by section 4, which details the latter. Section 5 describes the formulation and components of the AABC algorithm. Section 6 reports the results of numerical experiments on benchmark functions and the application of AABC to the FMS. Section 7 concludes with a summary of key findings and recommendations for future research.

2. ARTIFICIAL BEE COLONY ALGORITHM

Inspired by the intelligent foraging dynamics of honeybee swarms, the ABC algorithm, developed by Karaboga [8], transforms the optimization process into a metaphorical search for nectar. In this nature-inspired framework, each food source symbolizes a potential solution, scattered across a virtual landscape representing the problem's search space. The algorithm simulates the coordinated efforts of three types of bees: employed bees, onlooker bees, and scout bees, each contributing distinctively to the balance between exploration (searching new regions) and exploitation (refining known good areas). The journey begins with a random initialization phase, where a swarm of solution candidates is dispersed throughout the search domain. This initial population, typically represented by SN, mirrors the number of employed bees and is positioned using (1) to seed the algorithm's first steps.

$$x_{i,j} = x_{min,j} + rand(0,1)\left(x_{max,j} - x_{min,j}\right) \tag{1}$$

Where $x_{i,j}$ represents the solution in $i\ th$ food source in $j\ th$ dimension, D, in which $i = 1,2,3,\dots,SN$ and $j = 1,2,3,4,\dots,D$. Each of the food source is randomly assigned to the *SN* number of employed bees for the purpose of food quality evaluation.

In employed bee phase, the information of the current food source is used by the employed bee to adjust themselves to another random food source to improve the food source quality. The new solution, $v_{i,j}$ is generated using (2).

$$v_{i,j} = x_{i,j} + rand(-1,1)\left(x_{i,j} - x_{k,j}\right) \tag{2}$$

Where $x_{i,j}$ is randomly chosen solution. $x_{k,j}$ is randomly chosen neighbor partner solution in which $k = 1,2,3, \dots, SN$ and k must be different from i. The employed bees will then compare the quality of new solution, $v_{i,j}$ and the previous solution, $x_{i,j}$. If the newly discovered solution is superior to the previous one, the employed bee replaces the old solution with the new one. The fitness of this updated solution is then evaluated using (3).

$$fit_i = \begin{cases} \frac{1}{1+f(v_{i,j})} & f(v_{i,j}) \geq 0 \\ 1 + |f(v_{i,j})| & f(v_{i,j}) \leq 0 \end{cases} \quad (3)$$

In this context, $f(v_{i,j})$ represents the objective value of the newly generated solution. After completing their search, employed bees communicate the quality of the food sources to the onlooker bees through a mechanism akin to the waggle dance observed in nature. The quality of the waggle dance reflects the fitness of the corresponding food source—the more favorable the food source, the more expressive the dance. This quality assessment is quantified using (4).

$$P_i = \frac{fit_i}{\sum_{i=1}^{SN} fit_i} \quad (4)$$

In this context, the probability value P_i signifies the quality of the food source. Onlooker bees then randomly choose a food source based on its associated probability value P_i. Following the selection of a food source, onlooker bees refine the solution using (2). This selection mechanism mirrors a roulette wheel in the ABC algorithm. In instances where a food source fails to exhibit improvement within a specified timeframe, as indicated by trial limits, the said food source is deemed unsuccessful and abandoned. Employed bees then transition into scout bees, tasked with exploring for new food sources, a process facilitated by (1).

3. ARITHMETIC OPTIMIZATION ALGORITHM

Introduced by Hu *et al.* [17], the AOA is built upon fundamental arithmetic principles: division (D), multiplication (M), subtraction (S), and addition (A), which form the core of conventional mathematical problem-solving. Like other metaheuristic algorithms, AOA is designed to effectively balance exploration and exploitation to locate the global optimum. The algorithm follows the BODMAS principle (brackets, orders, division/multiplication, addition/subtraction), prioritizing division and multiplication over addition and subtraction to ensure a logically coherent execution of mathematical operations. This hierarchy ensures that arithmetic operations are executed in a logical and structured sequence when multiple operations are present in a computation.

AOA begins by initializing a population of candidate solutions (referred to as food sources) randomly across the search space. The algorithm then employs a dynamic mathematical function, known as the math optimizer acceleration (MOA), to govern the transition between exploration and exploitation phases. This switching behavior is formulated in (5), which plays a critical role in controlling the algorithm's convergence dynamics.

$$MOA(C_{iter}) = Min + C_{iter} \times \left(\frac{Max - Min}{Miter}\right) \quad (5)$$

Here, $Miter$ represents the maximum number of iterations, C_{iter} is the current iteration count, Max is the maximum value of MOA, and Min is the minimum value of MOA. The term $MOA(C_{iter})$ corresponds to the MOA value at the current iteration. The decision to switch between the exploration and exploitation processes is determined by comparing a random number r_1 with the current MOA value. If $MOA(C_{iter}) < r_1$, the exploration process unfolds, involving the use of division and multiplication operators. Conversely, if $MOA(C_{iter}) \geq r_1$, the exploitation process takes place, employing subtraction and addition operators.

The exploration phase in the AOA leverages multiplication and division operators due to their strong scattering characteristics, which enable broad coverage of the search space. These operations facilitate the generation of diverse candidate solutions across wide regions. The updated positions of new candidate solutions during exploration are calculated using (6) and (7).

$$v_{i,j} = best(x_j) \div (MOP + \epsilon) \left((UB_j - LB_j)\mu + LB_j\right), r_2 > 0.5 \quad (6)$$

$$v_{i,j} = best(x_j) \times MOP \times \left((UB_j - LB_j)\mu + LB_j\right), otherwise \quad (7)$$

In this context, $best(x_j)$ denotes the best solution found so far in the $j\,th$ dimension. The term μ represents a small floating-point constant introduced to prevent division by zero or singularity. The parameter μ serves as a control factor for adjusting the search behavior. UB_j and LB_j correspond to the upper and lower bounds of the search space, respectively, while r_2 is a randomly generated number within the interval [0, 1]. The math optimizer probability (MOP) is computed using (8).

$$MOP(C_{iter}) = 1 - \left(\frac{C_{iter}}{M_{iter}}\right)^{\frac{1}{\alpha}} \quad (8)$$

Here, the parameter α plays a pivotal role, dynamically tuning the precision of the exploitation process as the algorithm progresses through its iterations. The MOP, evaluated at the current iteration $MOP(C_{iter})$ further influences this balance. When the randomly generated value $r_2 > 0.5$, the algorithm ventures into exploration using the division operator; otherwise, it opts for the multiplication operator, both known for their broad search dispersion, enabling the discovery of diverse regions in the solution space.

Acknowledging that division and multiplication operators possess high dispersion characteristics—which can hinder the algorithm's ability to converge toward the optimal solution in later stages—AOA strategically utilizes subtraction and addition operations to enhance the exploitation phase. This localized search process is governed by (9) and (10).

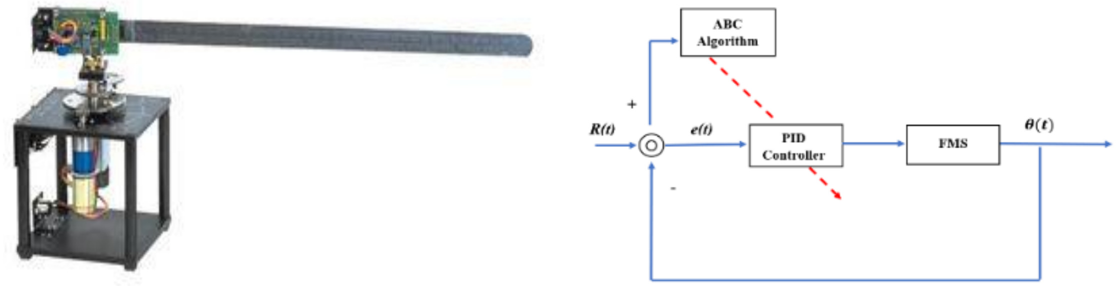
$$v_{i,j} = best(x_j) - MOP \times \left((UB_j - LB_j)\mu + LB_j\right), r_3 > 0.5 \quad (9)$$

$$v_{i,j} = best(x_j) + MOP \times \left((UB_j - LB_j)\mu + LB_j\right), otherwise \quad (10)$$

Here, r_3 represents a randomly generated number within the range [0, 1]. If $r_3 < 0.5$, the subtraction operator is applied during the exploitation phase; otherwise, the addition operator is used. These operators introduce only minor adjustments to the solution's position, allowing the algorithm to maintain focus on promising regions of the search space and reducing the risk of drifting away from potential optima.

4. FLEXIBLE MANIPULATOR SYSTEM

A manipulator is a mechanical assembly consisting of multiple interconnected links, designed to execute a diverse array of tasks across different application fields [18]. Its segmented structure draws inspiration from the versatility and precision of the human arm, which enables complex and coordinated motion. Traditional manipulator designs emphasize high structural stiffness to reduce system vibrations and enhance positional accuracy [19]. This rigidity is typically achieved through the use of dense and heavy materials. While effective in damping oscillations, this approach introduces significant drawbacks, including increased system weight, limited maneuverability, the necessity for larger actuators, higher energy demands, and elevated operational costs [20]. To mitigate these issues, the FMS has emerged as a promising alternative. An FMS generally comprises key components such as a strain gauge, shaft encoder, accelerometer, tachogenerator, reduction gearbox, and direct current (DC) motor [21]. In comparison to rigid manipulators, FMSs offer several benefits: lighter overall weight, improved mobility, smaller actuator size, lower power consumption, reduced manufacturing costs, a higher payload-to-weight ratio, and enhanced safety, particularly in collaborative environments involving human interaction [21]. However, due to their inherently lightweight and compliant structure, FMSs are more prone to vibrations when subjected to external forces or disturbances [22]. These oscillations can compromise the system's precision and control accuracy. To address this challenge, numerous control techniques have been investigated to suppress vibrations in FMSs. Among the most notable are the proportional-integral-derivative (PID) control [23], nonlinear adaptive control [24], time-delay control [25], linear quadratic regulator (LQR) [26], and input shaping [27]. Of these, PID control remains the most commonly employed due to its ease of implementation and robust performance. In the present study, both the modified optimization algorithm and the standard ABC algorithm are applied to fine-tune the PID controller parameters for the FMS, as illustrated in Figure 1. Figure 1(a) depicts the schematic diagram of the single-link flexible manipulator, while Figure 1(b) shows the control system architecture used for the FMS.



(a) (b)

Figure 1. Flexible manipulator system: (a) single link manipulator system [25] and (b) FMS block diagram

5. ARITHMETIC ARTIFICIAL BEE COLONY ALGORITHM

The AABC algorithm was developed by integrating the AOA into the framework of the original ABC algorithm. This hybridization is motivated by the need to overcome several key limitations of the conventional ABC algorithm, including its inadequate exploration capability, slow convergence speed, tendency to become trapped in local optima, and weak exploitation performance. In the standard ABC algorithm, only one employed bee is responsible for discovering a potential solution, which is then communicated to the onlooker bees. The search movement is controlled by (2), which relies on a randomly chosen solution as a reference point. However, the absence of a guiding mechanism in this process results in erratic exploration, hindering the algorithm’s ability to effectively focus on promising regions and thus slowing down convergence.

Furthermore, without a structured selection approach, the algorithm has an equal probability of choosing either the best or worst reference solutions, reducing its effectiveness in converging toward optimal solutions and increasing the likelihood of becoming trapped in local optima. The use of the addition operator in (2) also limits the step size of the search agents, thereby restricting their capacity to explore distant or boundary regions of the search space. This limitation diminishes the algorithm’s exploratory strength. In terms of exploitation, the original ABC algorithm lacks a mechanism to adaptively control the step size, which can lead to inefficient local searches and missed opportunities in promising areas. Additionally, the algorithm exhibits an imbalance between exploration and exploitation agents, with a greater emphasis on exploration (employed and scout bees) compared to exploitation (onlooker bees), further weakening its local search performance.

To address the limitations of the original ABC algorithm, several enhancement strategies have been introduced in the proposed AABC algorithm. In this improved version, the exploration phase traditionally performed by the employed bees in ABC is replaced with the exploration mechanisms from the AOA. This substitution leverages AOA’s strong exploratory capabilities through its division and multiplication operators, enabling a dual-mode search that enhances solution diversity. Additionally, the onlooker bee phase is split into two distinct stages. The first stage, referred to as the baron onlooker bee phase, adopts AOA’s exploitation strategies, utilizing addition and subtraction operators to allow bidirectional search movement. The second stage, termed the duke onlooker bee phase, applies a modified version of the standard onlooker bee equation, as detailed in (11).

$$v_{i,j} = best(x_j) + rand(-1,1)\left(x_{i,j} - x_{k,j}\right)b \qquad (11)$$

The best-so-far solution, denoted as $best(x_j)$, is utilized as the reference point for the search process within the jjj-th dimension. To enhance the accuracy of the exploitation phase, a step size control parameter, b, is introduced. This parameter governs the magnitude of positional updates near the optimal solution, thereby improving local search refinement. The value of b is computed using (12).

$$b\ (C_{iter}) = \frac{1}{e^{\left(1-\left(\frac{M_{iter}-C_{iter}}{C_{iter}}\right)\right)}} \qquad (12)$$

Where $b\ (C_{iter})$ represents the step size coefficient at the current iteration, with the step size value decreasing exponentially as the iteration numbers increase. The selection of the exponential function for the step size parameter is motivated by its high rate of change. The baron onlooker bee evaluates feasible solutions without using a probability equation, while the duke onlooker bee's evaluation of feasible solutions is determined by the probability value calculated using (4). The duke onlooker bee only evaluates the highest-quality food based on the probability of the food. The onlooker bee phase in the proposed AABC algorithm is divided into two categories—baron and duke onlooker bees—to further strengthen its exploitation capability. To ensure diversity in the search process, the original scout bee equation from the ABC algorithm is retained in AABC for generating new feasible solutions once the current solution has been fully exploited. Both the exploration and exploitation phases in AABC are guided by the best solution identified so far, which serves as a reference point to direct the search toward the global optimum. Moreover, the proposed algorithm maintains an equal number of exploration and exploitation agents, thereby achieving a balanced search process. The overall steps of the AABC algorithm are illustrated in Figure 1, while its detailed framework is presented in Algorithm 1.

Algorithm 1: Arithmetic artificial bee colony algorithm

```
Initialize population FS using (1)
Evaluate fitness of each candidate solution using (3)
Identify best solution so far
Compute MOA and MOP using (5) and (8)
Set iteration counter C_iter ← 0
while C_iter < M_iter do
   Generate random numbers r1, r2, r3
   for each candidate solution in FS do
      if r1 > MOA then
         if r2 > 0.5 then
            Update solution using Equation (6)
         else
            Update solution using Equation (7)
         end if
      else
         if r3 > 0.5 then
            Update solution using Equation (9)
         else
            Update solution using Equation (10)
         end if
         Evaluate fitness of updated solution
         Apply greedy selection
         Memorize best solution
         trial_iter ← trial_iter + 1
      end if
   end for
   Calculate selection probability Pi using Equation (4)
   Set iter ← 1, t ← 0
   while t < FS do
      if rand() < Pi then
         Update step size using Equation (12)
         Update solution using Equation (11)
         Evaluate fitness of updated solution
         Apply greedy selection
         Memorize best solution
         trial_iter ← trial_iter + 1
         t ← t + 1
      end if
      if trial_iter > limit then
         Replace solution using Equation (1)
      end if
      Memorize best solution
      C_iter ← C_iter + 1
   end while
end while
Return best solution found
```

6. RESULTS AND DISCUSSION

The proposed AABC algorithm is comprehensively evaluated by conducting experiments using five widely recognized benchmark functions, each featuring distinct and diverse landscape characteristics as referenced in [28]. Additionally, the algorithm is applied to a real-world application involving a single-link manipulator system, as detailed in [18]. The primary objective of these experiments is to thoroughly assess the algorithm's performance in terms of its convergence speed, robustness against variations in input or conditions, and overall accuracy in reaching optimal solutions.

6.1. Arithmetic artificial bee colony evaluation using benchmark functions

To assess the performance of the proposed AABC algorithm, a set of five widely recognized benchmark functions is employed—these functions are commonly utilized in optimization research [28]. The algorithm is tested on both 10-dimensional and 100-dimensional versions of the benchmark functions to evaluate its scalability and effectiveness. For this experiment, the population size is set to 80, with a trial limit of 50, and the stopping condition is defined as a maximum of 3,000 cycles. The comparative results for both the original ABC algorithm and the proposed AABC algorithm across the 10D and 100D test cases are summarized in Table 1.

Table 1. Comparison of results obtained by ABC and AABC on benchmark problems with 10 and 100-dimensions

Function	Algorithm	10 Dimension		100 Dimension	
		Average	STD	Average	STD
F1 Sphere	ABC	8.44E-17	1.85E-17	1.20E-11	1.02E-11
	AABC	0	0	5.03E-15	1.03E-14
F2 Ackley	ABC	9.41E-15	2.57E-15	7.11E-05	3.25E-05
	AABC	8.88E-16	8.88E-16	1.58E-13	5.32E-14
F3 Rosenbrock	ABC	2.64E-02	1.80E-02	2.11E+00	1.58E+00
	AABC	2.45E-03	2.78E-03	1.48E+02	6.62E+01
F4 Griewank	ABC	1.48E-11	5.81E-11	2.47E-07	1.10E-06
	AABC	0.00E+00	0.00E+00	2.34E-15	2.60E-15
F5 Rastrigin	ABC	0.00E+00	0.00E+00	3.40E+00	1.91E+00
	AABC	0.00E+00	0.00E+00	5.00E-12	7.59E-12

The tables present the best, average, median, worst, and standard deviation values of the benchmark functions across 30 independent runs. Values that show superior performance for either algorithm are highlighted in bold. The benchmark set includes functions F1 and F3, which are unimodal with a single global minimum. These functions are specifically designed to evaluate the algorithms' exploitation capabilities and convergence speed.

Table 2 demonstrates that the AABC algorithm outperforms the original ABC algorithm on the F1 benchmark function. AABC successfully reaches the global minimum of F1 (i.e., 0), consistently across multiple runs. This superior performance is also observed in the 30-dimensional version of F1, where AABC achieves better average and worst-case results compared to the original ABC. Furthermore, AABC exhibits greater stability, as indicated by its lower standard deviation. This improvement in exploitation precision can be attributed to the step size control mechanism embedded in the duke onlooker bee phase. For the F3 benchmark function, AABC surpasses ABC in the 10-dimensional case, delivering a better average performance. The improved results can be linked to the relatively low complexity of the problem in lower dimensions, which makes it easier for algorithms to converge to optimal solutions. However, as the problem dimension increases, the complexity grows—due to transformations such as rotation and convolution—making it more challenging. In the 100-dimensional F3 function, ABC outperforms AABC. The decline in AABC's performance at higher dimensions is primarily due to its exploitation mechanism, which uses the global best solution as a reference point. While this approach effectively guides the search toward promising regions, it also reduces population diversity. This loss in diversity is further exacerbated by the roulette wheel selection mechanism, which tends to favor 'super individuals'—i.e., highly fit solutions—causing the search to converge prematurely around limited areas of the solution space. In contrast, the ABC algorithm, despite using the same selection strategy, maintains greater diversity by referencing randomly selected solutions during the search process, which contributes to its better performance in high-dimensional problems.

Table 2. Statistical results of Wilcoxon signed-rank test for 10- and 100-dimensional benchmark problems

Problem complexity	10-dimensional		100-dimensional	
Function	Sign	p-value	Sign	p-value
F1	+	2.00E-06	+	2.00E-06
F2	+	7.75E-07	+	2.00E-06
F3	+	2.00E-06	-	2.00E-06
F4	+	2.00E-06	+	2.00E-06
F5	=	1.00E+00	+	2.00E-06
Overall outcome +/-/=	4/0/1		4/1/0	

The performance of the proposed AABC algorithm is further evaluated using multimodal benchmark functions F2, F4, and F5. Function F2 presents a challenging landscape, characterized by a nearly flat outer region and a large central basin. This structure increases the risk of algorithms getting trapped in local minima. Despite this, AABC demonstrates strong performance on F2, achieving better objective values than the original ABC algorithm. Similarly, F5 poses a high risk due to its numerous local optima. On this function, AABC outperforms ABC in terms of average, worst-case, and standard deviation values, indicating greater robustness and consistency. The improved performance of AABC on these complex landscapes can be attributed to the enhanced exploration capability introduced by the division and multiplication operators. Randomly alternating between these operators in each iteration helps maintain population diversity and reduces the risk of premature convergence. For function F4, which contains numerous widely scattered local minima, AABC consistently produces better median values than ABC. The ABC algorithm relies solely on the addition operator during its search process, which limits its ability to escape local optima due to insufficient directional guidance. In contrast, AABC shows significantly improved results, particularly in higher-dimensional instances of F4, where it achieves superior average and best objective values. This highlights the algorithm's enhanced exploration ability when tackling complex multimodal problems. Overall, the comparison results presented in Table 1 clearly demonstrate that the AABC algorithm exhibits superior search capabilities compared to the original ABC. The modifications introduced in AABC lead to a more effective balance between exploration and exploitation, enabling it to navigate complex search spaces more efficiently.

To verify the significance of the performance differences between the proposed AABC algorithm and the original ABC algorithm, a statistical significance test is conducted. The Wilcoxon signed-rank test—a widely used non-parametric method—is selected due to its effectiveness in handling data sets that do not require assumptions about normality. The test is performed at a 5% significance level to ensure a reliable comparison. Table 2 presents the corresponding p-values and outcomes of the test. In this table, the symbols '+', '−', and '=' represent cases where AABC is statistically superior, inferior, or comparable to ABC, respectively. This statistical analysis serves as a rigorous validation of the improvements introduced in the AABC algorithm.

6.2. Arithmetic artificial bee colony evaluation for flexible manipulator system controller

The AABC algorithm is employed to optimize the trajectory of FMS. In this application, the gain values obtained from the AABC algorithm are integrated into a PID controller to improve the system's dynamic response. The effectiveness of the algorithm is assessed using several error-based performance metrics. The evaluation process comprises a series of simulation experiments, including error minimization, transient response analysis, hub angle performance assessment, and single-objective optimization involving multiple control parameters. For all simulations, the population size is set to 30, with a problem dimensionality of 3. The algorithm is executed for 100 iterations, with a trial limit of 50. The search space for the optimization process is constrained within the range of [0, 10].

To evaluate the performance of the PID controller, three error metrics are employed: integral time absolute error (ITAE), integral absolute error (IAE), and integral square error (ISE). The proposed AABC algorithm is executed over 10 independent trials to determine the optimal values of the proportional gain (K_P) integral gain (K_I), and derivative gain (K_D) of the PID controller. For each error metric, the best result among the 10 trials is selected for performance evaluation. Additionally, the computational time required by the algorithm is recorded to assess its efficiency in minimizing the error criteria.

The results, summarized in Table 3 and illustrated in Figure 2, compare the performance of the AABC algorithm with the original ABC algorithm. In the table, boldface values indicate the lowest error achieved for each metric. The findings clearly show that AABC outperforms ABC across all three error criteria. Furthermore, AABC demonstrates shorter computational times, indicating improved efficiency. The superior performance of AABC can be attributed to its enhanced exploration capability, which allows the search agents to explore the solution space more broadly, thus increasing population diversity. At the same time, AABC exhibits strong exploitation capabilities, enabling it to converge to lower error values compared to ABC. While computational time tends to increase with population size due to the higher number of candidate solutions being evaluated, AABC still maintains faster performance under the given settings. In terms of PID tuning, AABC consistently produces smaller gain values than ABC, resulting in improved error metrics for the PID controller. The convergence plots in Figure 2 further confirm AABC's advantage, showing that it starts from a lower initial error and converges more rapidly. This is due to AABC's guided search mechanism, which selects the best-performing individual in the population as a reference point from the very first iteration.

Under the IAE criterion, the ABC algorithm exhibits a shorter settling time (T_S) compared to the AABC algorithm. This is primarily due to the influence of the integral gain, which plays a significant role in the accumulation of error over time. As the integral component increases in response to even small errors, the system tends to take longer to stabilize. The ABC algorithm features a higher integral gain than AABC, contributing to this faster settling time. However, ABC also incorporates a higher derivative gain, which counteracts the effects of its larger integral gain. Additionally, because the IAE criterion inherently leads to slower system responses, the inclusion of a stronger derivative component helps mitigate this drawback by accelerating system correction. In terms of overshoot (OS), the AABC algorithm outperforms ABC by producing a lower OS value. This improvement is attributed to AABC's smaller proportional and integral gains, which help suppress oscillatory behavior and enhance system stability. When evaluating the rise time (T_r), AABC demonstrates a faster response compared to ABC, indicating that it reaches the desired output level more quickly.

While for ISE criteria, the two algorithms perform equally well in settling time because they have the lowest values of integral and derivative gain. However, ABC algorithm beats AABC algorithm in OS because AABC algorithm does not have enough integral gain to balance the high proportional gain. AABC algorithm perform better where it reacts faster in term of rise time than original ABC algorithm. For the ITAE criteria, AABC algorithm outperforms ABC algorithm in settling time and OS. But AABC algorithm has the slowest rise time because it starts from the highest value compared to ABC algorithm, which hinders its convergence. This is because ITAE has the most sluggish initial response.

Table 3. Outcomes for output response flexible manipulator system experiment

Error criteria	IAE		ISE		ITAE	
Parameter	ABC	AABC	ABC	AABC	ABC	AABC
K_P	2.818	1.887	9.377	9.421	5.631	5.186
K_I	1.836E-08	0.000	1.081E-06	0.000	1.052E-07	0.000
K_D	0.953	0.667	2.586	2.589	1.863	1.730
Error	1.215E+02	1.197E+02	8.123E+03	8.122E+03	4.365E+02	4.360E+02
T_{s1} (s)	0.200	0.798	0.000	0.000	0.000	0.000
T_{r1} (ms)	536.529	535.132	450.495	448.971	852.510	862.595
OS_1 (%)	0.085	0.000	5.178	5.376	0.404	0.373
T_{s2} (s)	0.487	1.022	0.000	0.000	0.000	0.000
T_{r2} (ms)	648.342	663.730	599.739	598.440	643.301	645.828
OS_2 (%)	9.863	7.265	32.610	33.040	10.050	9.054
T_{s3} (s)	2.501	2.698	0.000	0.000	0.000	0.525
T_{r3} (ms)	914.849	590.738	520.771	516.829	883.428	887.224
OS_3 (s)	0.000	0.000	0.891	0.905	0.159	0.113

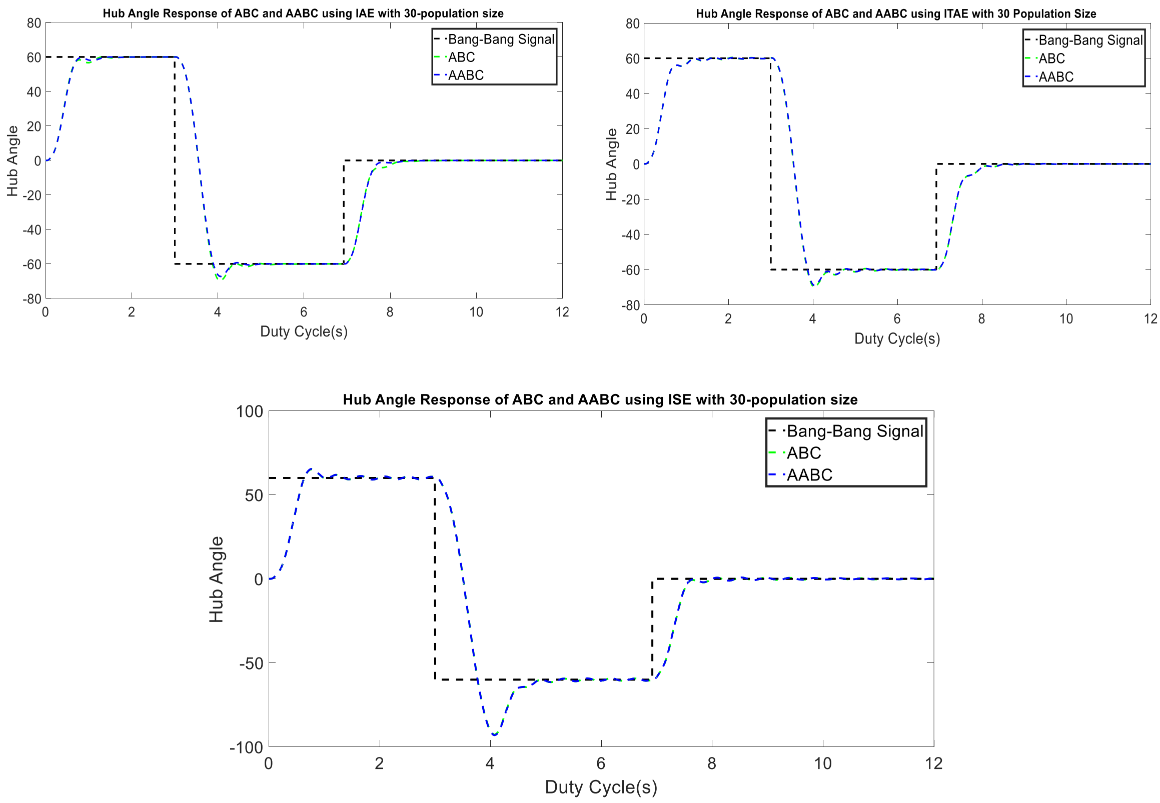


Figure 2. Output response flexible manipulator system experiment with different error criteria

7. CONCLUSION

The AABC algorithm is a refined variant of the standard ABC algorithm, designed specifically to enhance the optimization performance of FMS. This improvement aims to overcome key shortcomings of the original ABC, particularly by strengthening global exploration and reducing the risk of premature convergence to suboptimal solutions. In this modified approach, the employed bee phase is replaced with the exploration mechanism derived from the AOA, introducing a more systematic and efficient search methodology. Additionally, the AABC algorithm incorporates the global best solution as a dynamic reference point throughout the optimization process to accelerate convergence. To enhance local exploitation, the onlooker bee phase is restructured into two distinct components, coupled with a step-size control mechanism that enables precise fine-tuning during the local search phase. These enhancements collectively result in a more balanced and effective search strategy compared to the original ABC algorithm. The performance of AABC has been rigorously validated through comprehensive testing using ten widely accepted benchmark functions. Across these functions, AABC consistently demonstrated superior results in terms of convergence speed, solution accuracy, and result stability. Further insights into its convergence dynamics and statistical performance metrics affirm its effectiveness. To examine the algorithm's practical relevance, the AABC algorithm was applied to optimize the control parameters of FMS. Experimental evaluations using various performance indicators revealed that AABC outperformed the conventional ABC algorithm, delivering notable improvements in control precision. Comparative assessments with other ABC-based variants also confirmed the AABC’s competitive edge. These outcomes highlight the algorithm's robustness and efficiency as an optimization tool. Future research may explore hybridizing AABC with other metaheuristic techniques or intelligent systems—such as GA, PSO, or neural networks—to extend its capabilities for addressing more complex and high-dimensional optimization problems.

ACKNOWLEDGMENTS

The authors would like to express their sincere appreciation to the Centre for Research and Innovation Management (CRIM), the Faculty of Electrical Technology and Engineering (FTKE), and Universiti Teknikal Malaysia Melaka (UTeM) for their continuous support.

FUNDING INFORMATION

This research was funded by the Ministry of Higher Education Malaysia under the Fundamental Research Grant Scheme (FRGS/1/2022/TK07/UTEM/02/63).

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Mohd Ruzaini Hashim	✓	✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Ahmad Fitri Mazlan	✓	✓	✓			✓		✓	✓	✓	✓			
Mohammad Osman Tokhi	✓			✓		✓		✓		✓			✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

INFORMED CONSENT

Not applicable.

ETHICAL APPROVAL

Not applicable.

DATA AVAILABILITY
Data availability is not applicable to this paper as no new data were created or analyzed in this study.

REFERENCES

[1] D. Song, H. Pang, J. Yang, J. Zhou, Y. Liu, and H. Miao, "Research on harmonic elimination method for permanent magnet synchronous motors based on improved particle swarm optimization algorithm," in *2024 5th International Conference on Mechatronics Technology and Intelligent Manufacturing (ICMTIM)*, 2024, pp. 803–809, doi: 10.1109/ICMTIM62047.2024.10629360.

[2] M. Geranmehr, M. R. Nikoo, G. Al-Rawas, K. Al-Jabri, and A. H. Gandomi, "Application of metaheuristic algorithms in optimal design of sewer collection systems," in *Comprehensive Metaheuristics*, 2023, pp. 153–161, doi: 10.1016/B978-0-323-91781-0.00008-9.

[3] K. C. Chaudhary, "A modified version of the ABC algorithm and evaluation of its performance," *Heliyon*, vol. 9, no. 5, 2023, doi: 10.1016/j.heliyon.2023.e16086.

[4] R. D. A. Almunem, D. H. Muhsen, H. T. Haider, and T. Khatib, "A novel method for modeling of photovoltaic modules based on arithmetic optimization algorithm and cuckoo search," *Optik*, vol. 298, 2024, doi: 10.1016/j.ijleo.2023.171591.

[5] A. R. Abbasi and M. Zadehbagheri, "Expansion planning of hybrid electrical and thermal systems using reconfiguration and adaptive bat algorithm," *Heliyon*, vol. 10, no. 16, 2024, doi: 10.1016/j.heliyon.2024.e36054.

[6] S. P. Adam, S. A. N. Alexandropoulos, P. M. Pardalos, and M. N. Vrahatis, "No free lunch theorem: A review," in *Approximation and Optimization: Algorithms, Complexity and Applications*, vol. 145, Cham, Switzerland: Springer, 2019, pp. 57–82, doi: 10.1007/978-3-030-12767-1_5.

[7] T. C. Service, "A no free lunch theorem for multi-objective optimization," *Information Processing Letters*, vol. 110, no. 21, pp. 917–923, 2010, doi: 10.1016/j.ipl.2010.07.026.

[8] D. Karaboga, "An idea based on honeybee swarm for numerical optimization," *Technical Report*. Erciyes University, pp. 1–10, 2005.

[9] M. R. Hashim, A. K. Hyreil, and M. O. Tokhi, "Optimal tuning of PD controllers using modified artificial bee colony algorithm," *Journal of Telecommunication, Electronic and Computer Engineering*, vol. 10, no. 2–8, pp. 67–70, 2018.

[10] A. Rajasekhar, M. Pant, and A. Abraham, "Cauchy movements for artificial bees for finding better food sources," in *2011 Third World Congress on Nature and Biologically Inspired Computing*, 2011, pp. 279–284, doi: 10.1109/NaBIC.2011.6089608.

[11] E. Kaya, B. Gorkemli, B. Akay, and D. Karaboga, "A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 115, 2022, doi: 10.1016/j.engappai.2022.105311.

[12] R. Khosravanian, V. Mansouri, D. A. Wood, and M. R. Alipour, "A comparative study of several metaheuristic algorithms for optimizing complex 3-D well-path designs," *Journal of Petroleum Exploration and Production Technology*, vol. 8, no. 4, pp. 1487–1503, 2018, doi: 10.1007/s13202-018-0447-2.

[13] S. Agarwal, A. P. Singh, and N. Anand, "Evaluation performance study of firefly algorithm, particle swarm optimization and artificial bee colony algorithm for non-linear mathematical optimization functions," in *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, 2013, pp. 1–8, doi: 10.1109/ICCCNT.2013.6726474.

[14] L. Wang, L. Xie, and C. Xi, "Improved artificial bee colony algorithm for pressure source parameter inversion of Sakurajima volcano from InSAR data," *Geodesy and Geodynamics*, vol. 15, no. 6, pp. 635–641, 2024, doi: 10.1016/j.geog.2024.05.004.

[15] W. W. Lee and M. R. B. Hashim, "A hybrid algorithm based on artificial bee colony and artificial rabbits optimization for solving economic dispatch problem," in *2023 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, 2023, pp. 298–303, doi: 10.1109/I2CACIS57635.2023.10193351.

[16] S. Zhu, C. M. Pun, H. Zhu, S. Li, X. Huang, and H. Gao, "An artificial bee colony algorithm with a balance strategy for wireless sensor network," *Applied Soft Computing*, vol. 136, 2023, doi: 10.1016/j.asoc.2023.110083.

[17] G. Hu, J. Zhong, B. Du, and G. Wei, "An enhanced hybrid arithmetic optimization algorithm for engineering applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 394, 2022, doi: 10.1016/j.cma.2022.114901.

[18] H. M. Yatim and I. Z. M. Darus, "Development of an experimental single-link flexible manipulator system," *International Journal of Engineering and Technology (UAE)*, vol. 7, no. 2, pp. 7–12, 2018, doi: 10.14419/ijet.v7i2.29.13116.

[19] M. O. Tokhi, Z. Mohamed, and M. H. Shaheed, "Dynamic modelling of a flexible manipulator system incorporating payload: Theory and experiments," *Journal of Low Frequency Noise Vibration and Active Control*, vol. 19, no. 4, pp. 209–2289, 2000, doi: 10.1260/0263092001493056.

[20] A. F. V. Pedroso *et al.*, "An overview on the recent advances in robot-assisted compensation methods used in machining lightweight materials," *Robotics and Computer-Integrated Manufacturing*, vol. 91, 2025, doi: 10.1016/j.rcim.2024.102844.

[21] Q. X. Meng, X. Z. Lai, Y. W. Wang, and M. Wu, "A fast stable control strategy based on system energy for a planar single-link flexible manipulator," *Nonlinear Dynamics*, vol. 94, no. 1, pp. 615–626, 2018, doi: 10.1007/s11071-018-4380-1.

[22] Z. Mohamed, J. M. Martins, M. O. Tokhi, J. Sá da Costa, and M. A. Botto, "Vibration control of a very flexible manipulator system," *Control Engineering Practice*, vol. 13, no. 3, pp. 267–277, 2005, doi: 10.1016/j.conengprac.2003.11.014.

[23] B. A. M. Zain, M. O. Tokhi, and S. F. Toha, "PID-based control of a single-link flexible manipulator in vertical motion with genetic optimisation," in *2009 Third UKSim European Symposium on Computer Modeling and Simulation*, 2009, pp. 355–360, doi: 10.1109/EMS.2009.86.

[24] M. R. M. Nawawi, C. S. Horng, M. R. Hashim, and A. N. Hanaffi, "Development of output feedback sliding mode control for nonlinear system with disturbance," in *2011 IEEE International Conference on Control System, Computing and Engineering*, 2011, pp. 495–499, doi: 10.1109/ICCSCE.2011.6190576.

[25] D. Meng, H. Xu, H. Xu, H. Sun, and B. Liang, "Trajectory tracking control for a cable-driven space manipulator using time-delay estimation and nonsingular terminal sliding mode," *Control Engineering Practice*, vol. 139, 2023, doi: 10.1016/j.conengprac.2023.105649.

[26] S. Amertet, G. Gebresenbet, and H. M. Alwan, "Optimizing the performance of a wheeled mobile robots for use in agriculture using a linear-quadratic regulator," *Robotics and Autonomous Systems*, vol. 174, 2024, doi: 10.1016/j.robot.2024.104642.

[27] D. Newman and J. Vaughan, "Concurrent design of linear control with input shaping for a two-link flexible manipulator arm," *IFAC-PapersOnLine*, vol. 51, no. 14, pp. 66–71, 2018, doi: 10.1016/j.ifacol.2018.07.200.

[28] L. Wang, Q. Cao, Z. Zhang, S. Mirjalili, and W. Zhao, "Artificial rabbits optimization: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems," *Engineering Applications of Artificial Intelligence*, vol. 114, 2022, doi: 10.1016/j.engappai.2022.105082.

BIOGRAPHIES OF AUTHORS

Mohd Ruzaini Hashim obtained his B.Sc. (Hons.) in Electrical (Electronic) from Universiti Teknologi Mara (UiTM), Malaysia, his M.Sc. in Electrical Electronic from University of Leeds, U.K., and his Ph.D. in automation, control system engineering from University of Sheffield, U.K., respectively. He is now a senior lecturer at Universiti Teknikal Malaysia Melaka, Malaysia, where he conducts technical research on metaheuristic optimization algorithm computation approaches and their various applications. He can be contacted at email: ruzaini@utem.edu.my.

Ahmad Fitri Mazlan obtained his B.Eng. (Hons.) in Mechatronic Engineering from Universiti Malaysia Pahang (UMOP), Malaysia, and his M.Sc. in Mechatronic Engineering from Universiti Teknikal Malaysia Melaka, Malaysia. His research interests are in technical research on optimization algorithms and mechatronic system design. He can be contacted at email: m012020004@student.utem.edu.my.

Mohammad Osman Tokhi is a Professor at the School of Engineering, London South Bank University, U.K. He holds a B.Sc. in Electrical Engineering from Kabul University (1978) and a Ph.D. from Heriot-Watt University, U.K. (1988). His research focuses on active noise and vibration control, intelligent control, assistive robotics, and bio-inspired computing. A chartered engineer, he is a Fellow of the IET and a member of IIAV and CLAWAR. He is a life senior member of IEEE. He can be contacted at email: tokhim@lsbu.ac.uk.