https://doi.org/10.31436/iiumej.v26i3.3800

OPTIMIZING GENERATION COST AND REDUCING GAS EMISSIONS IN POWER GENERATION USING THE ARTIFICIAL BEE RABBIT OPTIMIZATION ALGORITHM

WEI WEN LEE, MOHD RUZAINI BIN HASHIM*, CHIN KIM GAN

Faculty of Electrical Technology and Engineering, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia

**Corresponding author: ruzaini@utem.edu.my*

(Received: 20 June 2025; Accepted: 09 August 2025; Published online: 9 September 2025)

ABSTRACT: This study develops a hybrid metaheuristic optimization algorithm named Artificial Bee Rabbit Optimization (ABRO) to improve generation cost efficiency and reduce gas emissions in power generation systems. By integrating the strengths of the Artificial Bee Colony (ABC) and Artificial Rabbits Optimization (ARO) algorithms, ABRO aims to overcome issues such as premature convergence and slow convergence speed commonly observed in ABC and ARO. This paper evaluates and compares the ABRO algorithm against a collection of optimization algorithms, such as ABC, ARO, the Crow Search (CSA) algorithm, and the Artificial Jellyfish Search (JS) algorithm. The evaluation covers four benchmark functions and extends to engineering applications, specifically in solving the economic dispatch, emission dispatch, and an integrated objective that considers financial and emission dispatch aspects for the IEEE 26-bus system. The simulation results show that ABRO generally outperforms the competing algorithms tested in solving various benchmark functions. ABRO consistently achieved the lowest mean, standard deviation, and minimum values, demonstrating superior convergence speed, robustness, and accuracy. Furthermore, the ABRO algorithm effectively enhances optimization regarding generation cost, generation emission, and an integrated objective that considers both economic and emission dispatch aspects for the IEEE 26-bus system.

ABSTRAK: Kajian ini membangunkan satu algoritma pengoptimuman metaheuristik hibrid yang dinamakan Pengoptimuman Buatan Bee Rabbit (ABRO) bagi meningkatkan kecekapan kos penjanaan dan mengurangkan pelepasan gas dalam sistem penjanaan tenaga. Gabungan kekuatan antara Koloni Buatan Bee (ABC) dan Pengoptimuman Buatan Rabbit (ARO), menghasilkan ABRO yang bertujuan mengatasi isu penumpuan pramatang dan kelajuan penumpuan perlahan, ysng sering berlaku pada ABC dan ARO. Kajian ini menilai algoritma ABRO dan beberapa algoritma pengoptimuman lain seperti ABC, ARO, algoritma Pencarian Crow (CSA), dan pengoptimum Pencarian Buatan Jellyfish (JS). Penilaian meliputi empat fungsi penanda aras dan diperluas kepada aplikasi kejuruteraan, khususnya dalam penyelesaian masalah pengagihan ekonomi, pengagihan pelepasan, serta objektif bersepadu yang memgambil kira kedua-dua aspek ekonomi dan pelepasan bagi sistem IEEE 26-bas. Dapatan simulasi menunjukkan bahawa algoritma ABRO secara amnya mengatasi prestasi algoritma lain yang diuji dalam menyelesaikan pelbagai fungsi penanda aras. ABRO secara konsisten mencapai nilai min, sisihan piawai, dan nilai minimum terendah, sekali gus membuktikan kelajuan penumpuan, keteguhan, dan ketepatan terbaik. Tambahan pula, algoritma ABRO mampu meningkatkan pengoptimuman dari segi kos penjanaan, pelepasan penjanaan, serta objektif bersepadu yang mempertimbangkan kedua-dua aspek ekonomi dan pelepasan bagi sistem IEEE 26-bas.

KEYWORDS: *Artificial Bee Colony, Optimization algorithm, Hybrid algorithm, Economic dispatch, Emission dispatch*

1. INTRODUCTION

Optimization is finding the optimal solution from a set of potential solutions. It either minimizes or maximizes the value of the objective function subject to some constraints [1]. Optimization algorithms are search methods that are designed to iteratively improve upon an initial solution until an optimal or near-optimal result is achieved [2],[3]. Artificial Bee Colony (ABC) [4], Artificial Jellyfish Search (JS) optimizer [5] and Artificial Rabbits Optimization (ARO) [6] are some examples of optimization algorithms that have been widely used in various fields for solving optimization problems.

According to the No-Free Lunch (NFL) theorem [7], there is no single optimization algorithm that can solve all problems well. ABC algorithm is an effective optimization approach for solving several practical issues, such as those in medical image processing [8], power dispatch [9] and wireless sensor networks [10]. However, the basic ABC algorithm undergoes a slow convergence rate and easily traps the local optima in specific scenarios [11]. Therefore, researchers have modified and adapted the ABC algorithm to suit specific problems and enhance its performance. Some of the variants of the ABC algorithm include hybrid ABC [12], adaptive ABC [13] and multiobjective ABC [14].

One of the recent optimization algorithms, called the Artificial Bee Rabbit Optimization (ABRO) algorithm [15], was first proposed in 2023 during a conference. The ABRO algorithm is a hybrid algorithm inspired by the ABC algorithm, which simulates the foraging behavior of honeybees and rabbits adapting to their natural environment. ABRO algorithm synthesizes these two models, aiming to achieve higher accuracy and convergence speed. The paper [15] was primarily focused on proposing the ABRO algorithm and comparing it against its predecessor algorithms, ABC and ARO, without employing statistical tools to evaluate its performance.

In this study, the ABRO algorithm is evaluated against other competitor algorithms, such as ABC, ARO, and the Crow Search Algorithm (CSA) [16], and Artificial Jellyfish Search (JS) optimizer [5]. The evaluation includes four benchmark functions with different landscapes and complexities. Moreover, these algorithms are used in the IEEE 26-bus system, and their performances for tackling the economic dispatch [17], emission dispatch [18] and combined economic with emission dispatch problems [19] are evaluated. Moreover, the ABRO algorithm has been validated statistically using the Wilcoxon Signed-Rank Test. This study aims to demonstrate that the ABRO algorithm, developed through hybridization, performs better in optimization, particularly in solving economic and emission dispatch issues in the IEEE 26-bus system.

The subsequent sections of the paper are structured as follows: Section 2 briefly explains the ABC and ARO algorithms. Section 3 highlights the inspiration, mathematical works, and flows of the ABRO algorithm. Section 4 presents ABRO's comparative results with other algorithms in optimizing four benchmark functions and the IEEE 26-bus system. Finally, Section 5 concludes the study's findings.

2. ARTIFICIAL BEE COLONY AND ARTIFICIAL RABBITS OPTIMIZATION ALGORITHMS

The Artificial Bee Colony (ABC) algorithm is well-known for its simplicity and strong global search capabilities, and has been successfully applied to solve problems across various fields. However, several studies have highlighted that ABC is poor in exploitation and suffers from issues such as premature convergence, slow convergence, and getting trapped in local optima [11], [12]. On the other hand, the Artificial Rabbits Optimization (ARO) algorithm is a newer technique designed to improve optimization performance. Despite its innovative design, the ARO still exhibits limitations in its exploration capabilities, which may lead to suboptimal solutions and stagnation [20]. Therefore, the hybridization technique is introduced to combine the strengths of both algorithms and enhance performance.

Hybridization in optimization algorithms involves combining multiple techniques, such as combining different metaheuristic algorithms or integrating machine learning methods [21]. The goal of the hybridization technique is to develop a more powerful and efficient optimization approach. The Artificial Bee Rabbit Optimization (ABRO) algorithm was developed based on this hybridization concept. To gain insight into the development of the proposed ABRO algorithm, it is essential to study the prior algorithms, namely the ABC and ARO algorithms.

2.1. Artificial Bee Colony Algorithm

The Artificial Bee Colony Algorithm (ABC) algorithm [4] was developed by Dervis Karaboga in 2005, which draws inspiration from the activity of honey bees when searching for food. In ABC, food sources are representative of solutions to the addressed problem. Each food source produces nectar proportionally to fit the solution to the problem. The ABC algorithm comprises three kinds of searching agents, such as employed bees (EBs), onlooker bees (OBs), and scout bees (SBs). EBs are in charge of searching for food sources around their hives, OBs select food based on the information they share by EBs, and SBs will explore new food sources if the EBs' solution has not improved for a specific iteration or exceeds the predefined limit. The ABC algorithm terminates once it reaches the predefined number of iterations or discovers a satisfied solution [22]. The ABC algorithm has demonstrated its applicability in various engineering applications, such as path planning of mobile robots, designing of optimized PID controllers, and MPPT power extraction [11]. The ABC algorithm is characterized by its simplicity, with few control parameters; however, it has limitations, including slow convergence and susceptibility to premature convergence. Hence, many researchers modify the ABC algorithm to enhance its performance, overcome its limitations, and adapt it to specific problems [12],[23].

Recent studies have shown that hybridizing the ABC algorithm with other techniques enhances its performance. For example, Yan et al. (2020) proposed a hybrid ABC with Tabu-Neighborhood search to improve route optimization in logistics [24], while Mala et al. (2023) introduced ABC variants (NMABC and LHABC) enhanced with neural network strategies and stochastic gradient descent for improved classification accuracy [25]. Moreover, Kumar et al. (2022) integrated ABC with Differential Evolution and Whale Optimization Algorithm (ABC-DE-WOA) to accelerate convergence [26]. Ramachandran et al. (2022) developed FRCSA-ABC by merging ABC with a fuzzy logic-enhanced Crow Search Algorithm for economic dispatch problems [27]; and Ustun et al. (2022) incorporated DE's mutation and crossover into a modified ABC for higher precision [28]. These works illustrate the growing interest in hybridizing ABC to expand its applicability and effectiveness.

https://doi.org/10.31436/iiumej.v26i3.3800

2.2. Artificial Rabbits Optimization

The Artificial Rabbits Optimization (ARO) algorithm [6] is inspired by the survival tactics employed by rabbits. The survival strategies of rabbits include energy shrink strategies, detour foraging techniques, and random hiding techniques. The detour foraging acts as the exploration phase, where the rabbits seek food near other rabbits' nests, which can prevent predators from discovering their nests. In contrast, random hiding acts as the exploitation phase, where the rabbits will randomly pick one of their burrows to hide in to avoid being captured by predators. The energy shrink approach relies on the energy factor, which leads to a transition from the detour foraging approach to the random hiding approach. The ARO algorithm continues to iterate until the stopping criterion is met, such as maximum iterations or the desired fitness value is achieved. The ARO algorithm has been extensively applied in various areas, such as PV allocation [29], stock price prediction [30] and water productivity prediction [31].

The ARO algorithm is a modern optimization approach that is easy to implement and requires minimal control parameters [6]. However, on the contrary, the ARO suffers from several shortcomings, including premature convergence, a dearth of population diversity, and an imbalance between exploitation and exploration [32]. For instance, to improve the efficiency of the ARO algorithm, researchers [32] have combined the ARO algorithm with the dimension learning-based hunting technique and have employed it to investigate optimal sizing of a stand-alone hybrid system. Recently, Li et al. (2024) proposed a hybrid algorithm combining ARO with the Bottlenose Dolphin Optimizer (BDO) to enhance feature selection in network intrusion detection tasks. Their modified algorithm, LBARO, incorporates multiple improvements such as an adaptive switching mechanism, Levy flight strategy, and dynamic lens imaging to overcome ARO's stagnation and improve exploration and exploitation. [33]. The success of LBARO across benchmark functions and classification datasets demonstrates the growing interest and effectiveness of hybrid ARO strategies in complex optimization problems.

3. METHODOLOGY

3.1. Fundamentals of Artificial Bee Rabbit Optimization

The Artificial Bee Rabbit Optimization (ABRO) algorithm [15] is an optimization algorithm that combines the techniques from two other algorithms, namely the ABC algorithm [4] and the ARO algorithm [6]. The ABRO algorithm is a metaheuristic approach that iteratively generates and evaluates candidate solutions to guide the search toward better solutions for various optimization problems. The ABRO algorithm was developed based on the observations made on two existing algorithms: ABC and ARO. ABC shows strong exploration and requires few control parameters, but it struggles with exploitation and converges slowly [34]–[36]. In contrast, ARO demonstrates rapid convergence and strong exploitation capabilities [6], but it is a relatively new algorithm proposed in 2022 and might have limitations in terms of performance, efficiency, and robustness.

Drawing inspiration from the strengths and limitations of both ABC and ARO, the ABRO algorithm has been developed. The aim is to achieve a balanced compromise between exploration and exploitation, using the strength of both algorithms and aiming to achieve better optimal outcomes. The ABRO algorithm combines the two key approaches from its predecessor algorithm. It involves the employed bee phase of the ABC and the random hiding strategy from the ARO, as shown in Figure 1.

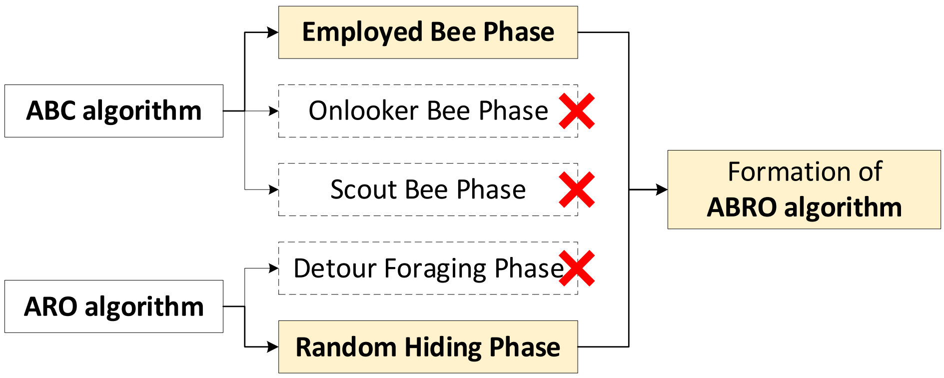

Figure 1. The formation of the ABRO algorithm

3.2. Mathematical Models of Artificial Bee Rabbit Optimization

The overview and the mathematical models of the ABRO are outlined. The ABRO algorithm consists of the employed bee stage and the random hiding stage. The total population is denoted NP refers to the combined count of search agents that are actively engaged in both stages. The term "food sources" (FS) relates to half of the population and refers to the search agents involved in either the employed bee phase or the random hiding phase. Each search agent is restricted to operating with a single food source or solution at a time. D represents the dimension or the number of input variables the optimization task takes.

The term Sol is an abbreviation used to denote a candidate solution or a feasible solution. During the initialization phase, these solutions are distributed at random throughout D-dimensional search space. Equation (1) is used to execute the initialization process.

$$Sol_{i,j} = rand(0,1) \times \left(ub_{i,j} - lb_{i,j}\right) + lb_{i,j} \tag{1}$$

$Sol_{i,j}$ represents the solution randomly found within the population, where i refers to the i^{th} food source with $i = 1,2,3,\dots,FS$ and j refers to the j^{th} dimensional vector with $j = 1,2,\dots,D$. The term $rand(0,1)$ means a pseudorandom number is generated in the interval of 0 to 1. The "ub" indicates the upper limit, where the function's largest possible value can attain. Conversely, the "lb" referred to as the lower bound, represents the minimum possible value that the function can take.

After the initialization process, the main loop of the ABRO algorithm starts. The method initiates with the employed bee stage. At this stage, each search agent visits its solution $Sol_{i,j}$ and evaluates its objective function value, which is denoted as $f(Sol_{i,j})$. During the employed bee phase, the search agent generates new solutions, $NSol_{i,j}$ based on its current candidate solution using equation (2) and evaluates its fitness using equation (3).

$$NSol_{\ i,j} = Sol_{\ i,j} + \left(Sol_{\ i,j} - Sol_{\ k,j}\right) \times \emptyset \tag{2}$$

where the k and j are random selection index, $k = 1,2,\dots,FS$ but the k must be different from i whereas $j = 1,2,\dots,D$. $\emptyset$ refers to the random number that is generated between -1 to 1. The fitness equation described in (3) is used to direct the search process. Xs appears as the variable or placeholder, which is used generically to represent either $Sol_{\ i,j}$ or $NSol_{\ i}$ can be inserted into the equation (3) depending on the context.

$$Fit(Xs_i) = \begin{cases} \frac{1}{1+f(Xs_i)} & ; \quad f(Xs_i) \geq 0 \\ 1 + |\, f(Xs_i)\,| & ; \quad f(Xs_i) \leq 0 \end{cases} \tag{3}$$

During the search process, it is essential to ensure that the solutions generated by the algorithm remain within the defined search space. Hence, the new solution is checked against the defined ub and lb by using the equation in (4). If the generated solution's parameter exceeds these boundaries, it will be updated to fit within the limits.

$$NSol_{i,j} = \begin{cases} ub_{i,j} & ; \quad NSol_{i,j} > ub_{i,j} \\ lb_{i,j} & ; \quad NSol_{i,j} < lb_{i,j} \\ NSol_{i,j} & ; \quad lb_{i,j} < NSol_{i,j} < ub_{i,j} \end{cases} \tag{4}$$

Then, the search agents compare the fitness of the solutions and perform greedy selection using the equation as shown in (5). If the newly generated solution, $NSol_i$ exhibits greater fitness (Fit) than the existing candidate solution ($Sol_{i,j}$), the new solution, $NSol_i$ will be substituted and memorized for the next stage. If not, the existing candidate solution Sol_i remains.

$$Sol_i = \begin{cases} NSol_i & ; \quad Fit(NSol_i) > Fit(Sol_i) \\ Sol_i & ; \quad Fit(NSol_i) \leq Fit(Sol_i) \end{cases} \tag{5}$$

The subsequent phase is the random hiding phase. The variable "motion length" (L) represents the step length and can be calculated using equation (6). L is varied and will be generated from the higher to the lower steps along with the iterations. Higher steps enable the search agent to search in further locations, while the lower step is vice versa. T refers to the maximum number of cycles and $iter$ represents the iteration at a particular time.

$$L = \left(e - e^{\left(\frac{iter-1}{T}\right)^2} \right) \times \, sin(2\pi \, \times rand(0,1)) \tag{6}$$

The vector g as shown in equation (7) illustrates the random ordering of rd integers selected randomly from 1 to D without repeating. The term "$ceil$" in equation (8) represents the ceiling function that rounds the specific element to the nearest integer larger than or equal to that element. The g creates a random permutation of rd unique integers selected randomly from 1 to D.

$$g = randperm(D, rd) \tag{7}$$

$$rd = ceil(rand(0,1) \times D) \tag{8}$$

The vector c acts as the mapping vector and is used to randomly select the number of search agents' variables for mutation. The equation in (9) can be used to determine the vector c. The v represents the selected dimension vectors with possibly $v = 1, \dots, [random(0,1) \times D]$ whereas p represents the p^{th} element of the search agents with $p = 1, \dots, D$. R is the running operator that controls the running characteristic of search agents, and it can be calculated using equation (10).

$$c\,(p) = \begin{cases} 1 & if \; p == g(v) \\ 0 & else \end{cases} \tag{9}$$

$$R = L \, \times c \tag{10}$$

The random hiding stage is inspired by the survival technique of rabbits, digging different burrows and randomly choosing one of them as their nest for hiding, with the hope of confusing the predator and reducing the chances of being caught. Equation (11) selects which burrow/dimension vector to mutate randomly. Each search agent has several dimensional vectors, also known as variables. By adopting the hiding technique of rabbits, one of the dimensional vectors is selected for mutation.

https://doi.org/10.31436/iiumej.v26i3.3800

$$gr(q) = \begin{cases} 1 & if\ q == ceil(rand(0,1) \times D) \\ 0 & else \end{cases} \tag{11}$$

where q represents the only chosen dimension vector with possibly $q = 1, \ldots, D$. gr represents the search agent's dimension vectors (burrows), if $gr = 1$ represents the selected burrow/ dimension vector to mutate and if $gr = 0$ represents the selected dimension vector is not affected.

H as shown in equation (12) represents the hiding parameter, and it will reduce linearly from 1 to $1\ /\ T$ iteratively. The search agent starts searching in broader neighbourhoods and narrows down as iterations increase. b as shown in equation (13) mutate the randomly selected dimension vector/burrow.

$$H = (T - iter + 1)/T \tag{12}$$

$$b_i = Sol_{\,i} + H\ \times gr \times Sol_{\,i} \tag{13}$$

Subsequently, by using equation (14), the newly generated solution $NSol_{\,i}$ is generated. The i^{th} search agent intends to vary its current solution toward the randomly selected dimension vector (burrow) where i= $1, \ldots, FS$. The boundaries of $NSol_{\,i}$ are checked to make sure they are within the search space.

$$NSol_{\,i} = Sol_{\,i} + R \times (\,rand(0,1) \times b_i - Sol_{\,i}) \tag{14}$$

Finally, the solution update for the ith search agent is determined using equation (15). The selection is based on the objective function value $f()$. In the minimization problem, if the newly generated solution $NSol_{\,i}$ has a lesser objective function value than the existing solution $Sol_{\,i}$, the solution will be updated to the newly generated solution $NSol_{\,i}$.

$$Sol_{\,i} = \begin{cases} NSol_{\,i} & ; \quad f(Sol_{\,i}) > f(NSol_{\,i}) \\ Sol_{\,i} & ; \quad f(Sol_{\,i}) \leq f(NSol_{\,i}) \end{cases} \tag{15}$$

Table 1. Pseudocode of ABRO algorithm

ABRO Algorithm

Set the parameters. NP, FS, D, T, ub, lb
Initialize the population of candidate solutions of FS using (1)
$iter$=1
While $iter \leq T$ **do**
Employed Bee Stage:
for all *i = 1: FS* ***do***
Utilize (2) to create new solutions
Use (3) to perform a fitness evaluation for the solutions
Analyse and update the solutions using (5)
end for
Random Hiding Stage:
for all *i = 1: FS* **do**
Determine running operator R using (6 – 10)
Calculate b using (11 – 13)
Generate a new solution using (14)
Evaluate the objective function and do selection using (15)
Update the search agent's solution
end for
Memorize the best solution and the best function value
Iteration = Iteration+1
Until ($iter = T$)

The ABRO algorithm repeats the process by iteratively refining its solution in the search space until it meets the optimal solution or reaches the predefined maximum iterations. Table 1 shows the pseudocode of the ABRO algorithm. The pseudocode includes key components such as the initialization, iteration steps, employed bee stage, random hiding stage, and termination conditions.

4. RESULTS AND DISCUSSION

Based on previous studies, the ABC algorithm has limitations, including slow convergence and a tendency to get stuck at local optima. To address these shortcomings, the ABRO algorithm has been developed to improve ABC's convergence speed and accuracy. To evaluate the algorithm's effectiveness, the ABRO algorithm's results are compared with those of four existing algorithms: ABC, ARO, JS, and CSA, across four benchmark functions and in solving economic and emission dispatch problems for the IEEE 26-bus system. The ABRO algorithm demonstrates faster convergence and superior performance compared to the others. Further details are provided in this section.

4.1. Benchmark Functions

The ABRO algorithm has been evaluated with four benchmark functions, including the Sum Squares function, Rastrigin function, Ackley function, and Zakharov function that are taken from the literature [5]. Benchmark functions are sets of functions suitable for evaluating the performance of any optimization tasks. Several existing optimizers, such as ABC, ARO, CSA, and JS, are used to compare with ABRO. Each optimizer is run for 30 independent runs. To maintain equitable conditions for each algorithm, the number of search agents (NP) is adjusted to 50, whereas the maximum iteration (T) is selected to 100. Besides NP and T, ABC and CSA algorithms require additional parameter settings. The limit of ABC is set to $(NP/2) \times D$, while CSA requires setting the awareness probability to 0.1 and its flight length to 2. The optimal solution ($Best$), mean ($Mean$) and standard deviation (SD) of all algorithms are recorded in Table 2. Based on Table 2, the ABRO algorithm outperforms the other algorithms as it can achieve the best minimum value, the lowest SD and lowest $Mean$ in every selected benchmark function.

Wilcoxon Signed-Rank Test (WSRT) [37] has been utilized to validate the proposed ABRO's performance compared to competitor algorithms such as ABC, ARO, CSA, and JS across four benchmark functions. This statistical test is well-suited for assessing the significance of differences between paired data sets. The null hypothesis is that the median of differences between competitor algorithms and the proposed ABRO equals 0. The p-value reflects the probability of the null hypothesis being true. A significant level of 0.05 has been chosen. When the p-value is less than 0.05, it indicates statistically significant differences, leading to the rejection of the null hypothesis. "$+$" indicates significant improvement, "$=$" means no difference, and "$-$" refers to significant worsening in the ABRO algorithm versus competitors. Referring to Table 3, ABRO significantly improves over every benchmark function compared to other studied algorithms.

Figure 2 shows the convergence plots. ABRO converges faster than other algorithms in every benchmark function, as ABRO requires fewer iterations to find the optimal solution. Faster convergence is often desirable in optimization algorithms, especially when time and resources are valuable considerations. This efficiency can significantly reduce computational time and resources.

Table 2. Optimization of benchmark mathematical functions with different algorithms

FUNCTION	INDEX	ABRO	ABC	ARO	JS	CSA
$f_1(x)$	*Mean*	**5.73E-17**	1.41E+02	2.47E-08	4.14E-04	7.75E+01
	SD	**1.72E-16**	1.51E+02	1.28E-07	2.11E-04	1.56E+01
	Best	**6.65E-30**	5.46E+00	2.60E-13	1.04E-04	4.74E+01
$f_2(x)$	*Mean*	**1.78E-15**	9.18E+01	1.11E-08	1.55E+02	1.16E+02
	SD	**4.83E-15**	1.56E+01	2.55E-08	1.81E+01	2.18E+01
	Best	**0.00E+00**	4.93E+01	7.46E-14	1.04E+02	7.71E+01
$f_3(x)$	*Mean*	**3.94E-09**	1.36E+01	1.41E-05	1.44E-02	6.38E+00
	SD	**9.28E-09**	1.16E+00	2.88E-05	2.51E-03	5.48E-01
	Best	**4.00E-15**	1.10E+01	3.73E-07	9.45E-03	5.33E+00
$f_4(x)$	*Mean*	**4.08E-15**	3.30E+01	1.23E-08	7.80E-02	5.42E+00
	SD	**1.54E-14**	1.50E+01	3.36E-08	6.13E-02	4.88E+00
	Best	**3.95E-22**	5.47E+00	4.80E-13	1.01E-02	7.22E-01

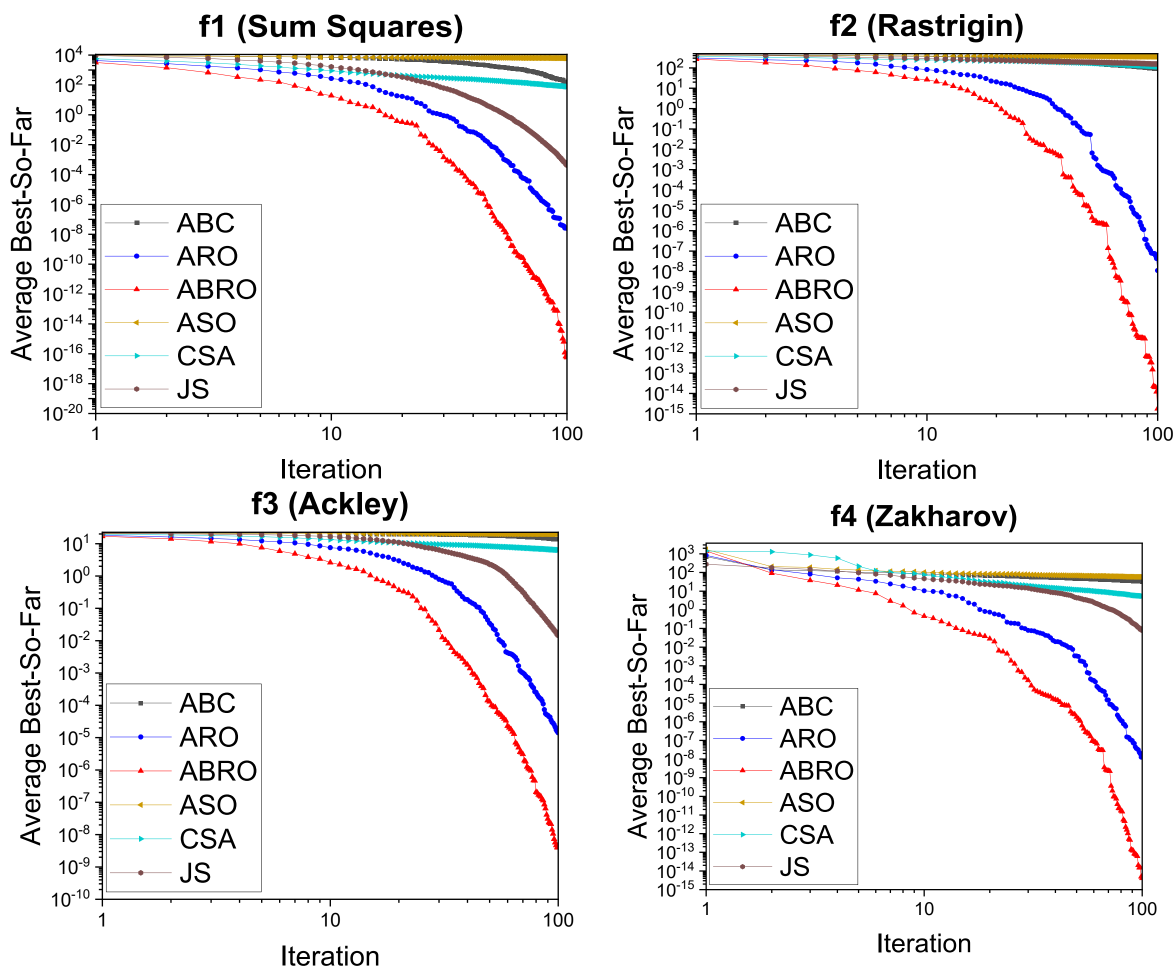

Figure 2. Comparisons of convergence curves

Table 3. Significant comparison of WSRT for ABRO versus other competitors

Function	ABC versus ABRO				ARO versus ABRO			
	p-value	Median		Improve	p-value	Median		Improve
		ABC	ABRO			ARO	ABRO	
$f_1(x)$	2.00E-6	1.02E02	2.89E-19	+	2.00E-06	2.18E-10	2.89E-19	+
$f_2(x)$	2.00E-6	9.19E+01	0.00E+00	+	2.00E-06	1.21E-10	0.00E+00	+
$f_3(x)$	2.00E-6	1.39E+01	4.68E-10	+	2.00E-06	4.10E-06	4.68E-10	+
$f_4(x)$	2.00E-6	3.35E+01	8.83E-17	+	2.00E-06	4.68E-10	8.83E-17	+

Function	JS versus ABRO				CSA versus ABRO			
	p-value	Median		Improve	p-value	Median		Improve
		JS	ABRO			CSA	ABRO	
$f_1(x)$	2.00E-06	3.64E-04	2.89E-19	+	2.00E-06	7.88E+01	2.89E-19	+
$f_2(x)$	2.00E-06	1.61E+02	0.00E+00	+	2.00E-06	1.16E+02	0.00E+00	+
$f_3(x)$	2.00E-06	1.37E-02	4.68E-10	+	2.00E-06	6.37E+00	4.68E-10	+
$f_4(x)$	2.00E-06	5.94E-02	8.83E-17	+	2.00E-06	3.83E+00	8.83E-17	+

4.2. EEE 26-Bus System

Next, the ABRO algorithm and four other algorithms are implemented on the IEEE 26 bus system [38] to evaluate their applicability in optimizing the economic dispatch, emission dispatch, and combined economic and emission dispatch problems. The IEEE 26-bus system is a widely used test system in power systems research and optimization. The IEEE 26-bus system is a small-scale system that comprises five generators and one slack bus [38]. The generators are located respectively at buses 2, 3, 4, 5, and 26, with different power generation ranges [39] as described in Table 4. The system's power demand is 1,263MW. Each generator in the IEEE 26-bus system has its unique cost and emission coefficients, as listed in Table 4 [40]. Optimization's aim in this context is typically to determine the optimal allocation of power generation among accessible generating units in a power system to meet power demand, considering various constraints and objectives.

This study will primarily concentrate on three cases: economic dispatch, emission dispatch, and the combined optimization of financial and emission dispatch. Each algorithm has been run 30 times for every case to evaluate the consistency. Also, each algorithm has been configured with an NP value of 10 and a T value of 100. The recorded metrics include the mean, standard deviation, minimum, and maximum value.

Table 4. Generator limits, cost coefficients, and emission coefficients for the IEEE 26-bus system

Generator P_{Gi}	Power limits megawatt (MW)	Cost Coefficient			Emission Coefficient				
		a_i	b_i	c_i	α_i	β_i	γ_i	ε_i	λ_i
P_{G1} (Slack)	$100 \leq P_{G1} \leq 500$	0.0070	7.0	240	4.091	-5.543	6.490	2.0e-4	2.857
P_{G2}	$50 \leq P_{G2} \leq 200$	0.0095	10.0	200	2.543	-6.047	5.638	5.0e-4	3.333
P_{G3}	$80 \leq P_{G3} \leq 300$	0.0090	8.5	220	4.258	-5.094	4.586	1.0e-6	8.000
P_{G4}	$50 \leq P_{G4} \leq 150$	0.0090	11.0	200	5.326	-3.550	3.380	2.0e-3	2.000
P_{G5}	$50 \leq P_{G5} \leq 200$	0.0080	10.5	220	4.258	-5.094	4.586	1.0e-6	8.000
P_{G26}	$50 \leq P_{G26} \leq 120$	0.0075	12.0	190	6.131	-5.555	5.151	1.0e-5	6.667

https://doi.org/10.31436/iiumej.v26i3.3800

4.2.1. Case 1: Economic Dispatch

Economic dispatch aims to minimize the overall cost of generation while meeting the load demand by identifying the optimal schedule of all generating units. Economic dispatch is essential because it can help reduce the operating costs of power systems, leading to lower electricity prices for consumers. Its mathematical model is formulated in equation (16). In a power system with N generators are actively engaged in operation, the total generation cost can be expressed as FC, where FC is the sum of the individual generation costs of each generator. P_{Gi} represents the output power from the generator, while a_i, b_i, and c_i are the cost coefficients of the i^{th} generation unit in the IEEE 26-bus system.

$$FC(\vec{P}_G) = \sum_{i=1}^{N} a_i P_{Gi}^2 + b_i P_{Gi} + c_i \qquad dollar/hr \tag{16}$$

Table 5 lists the algorithms' performances in reducing the generation cost for the IEEE 26-bus system. ABRO consistently exhibits the lowest mean and the smallest standard deviation and attains the lowest minimum values compared to the other algorithms. ABRO can achieve mean generation costs of \$ 15446.7425 per hour. Also, the convergence curve of all the algorithms for minimizing the generation costs is shown in Figure 3.

Table 5. Comparisons of Generation costs for the IEEE 26-bus (case 1)

Algorithm	Cost (dollars/hr)			
	Mean	SD	Min (Best)	Max (Worst)
ABRO	**15446.7425**	**0.0008**	**15446.7407**	**15446.7437**
ABC	15447.0838	0.4662	15446.7894	15448.6256
ARO	15447.1746	0.1897	15446.9767	15447.8997
JS	15446.8690	0.1229	15446.7697	15447.2949
CSA	15449.9793	1.8234	15448.5454	15455.4806

Furthermore, the WSRT has been run, and the output indicates that the ABRO algorithm significantly outperforms the other competitors for economic dispatch in the IEEE 26-bus system. To illustrate this point, the median ABRO, $Mdn = 15446.74$ is statistically significantly lower than the median ABC, $Mdn = 15446.84$, $z = -4.78$, $p = 2.00E - 06$. Besides, the median ABRO, $Mdn = 15446.74$ is statistically significantly lower than the median ARO, $Mdn = 15447.14$, $z = -4.78$, $p = 2.00E - 06$. Also, the median ABRO, $Mdn = 15446.74$ is statistically significantly lower than the median JS, $Mdn = 15446.82$, $z = -4.78$, $p = 2.00E - 06$. Lastly, the median ABRO, $Mdn = 15446.74$ is statistically significantly lower than the median CSA, $Mdn = 15449.30$, $z = -4.78$, $p = 2.00E - 06$.

4.2.2. Case 2: Emission Dispatch

The second case comprises emission dispatch. Emission dispatch focuses on environmental factors and aims to minimize the emission of pollutants, such as nitrogen oxides, sulphur dioxide, and carbon dioxide. By scheduling the generating units effectively, emission dispatch can help reduce emissions ton/hr while maintaining the power supply. Its mathematical model is represented in equation (17). γ_i, β_i, α_i, ε_i and λ_i are the emission coefficients of atmospheric pollutants.

$$EM(\vec{P}_G) = \sum_{i=1}^{N} 10^{-2} \left(\gamma_i P_{Gi}^2 + \beta_i P_{Gi} + \alpha_i \right) + \varepsilon_i e^{\lambda_i P_{Gi}} \qquad ton/hr \tag{17}$$

Table 6 lists the algorithms' performance in reducing the generation emission for the IEEE 26-bus system. In Table 6, ABRO consistently demonstrates the lowest mean and the smallest standard deviation, achieving the best minimum value when compared to the other algorithms. The ABRO algorithm can achieve an average generation emission of 15869.7847 tons/hr. In addition, the convergence curve of all the algorithms for generation emission in the IEEE 26 bus system is illustrated in Figure 3.

Table 6. Comparisons of generation emission for the IEEE 26-bus (case 2)

Algorithm	Emissions (ton/hr)			
	Mean	SD	Min (Best)	Max (Worst)
ABRO	**15869.7847**	**0.0001**	**15869.7846**	**15869.7852**
ABC	15870.0631	1.5252	15869.7846	15878.1385
ARO	15977.2962	68.2082	15896.8664	16142.6559
JS	15881.3237	3.9165	15871.8956	15887.0627
CSA	16704.0726	297.5323	15982.4369	17159.0742

Moreover, the WSRT has been run, and the output indicates that the ABRO algorithm significantly outperforms the other competitors except the ABC algorithm for emission dispatch in the IEEE 26-bus system. To illustrate this point, the median ABRO, Mdn=15869.78 is statistically significantly lower than the median ARO, $Mdn = 15955.43$, $z = -4.78$, $p = 2.00E - 06$. Also, the median ABRO, $Mdn = 15869.78$ is statistically significantly lower than the median JS, $Mdn = 15881.50$, $z = -4.78$, $p = 2.00E - 06$. In addition, the median ABRO, $Mdn = 15869.78$ is statistically significantly lower than the median CSA, $Mdn = 16754.61$, $z = -4.78$, $p = 2.00E - 06$. Lastly, although ABRO's emission results are better on average than ABC's, the WSRT reveals no significant difference (p-value = 0.6). This suggests that while ABRO generally performs better, the difference is not statistically significant at the 0.05 level.

4.2.3. Case 3: Economic with Emission Dispatch

The third case is about solving the objective of encompassing the economic and emission dispatch problem. The combined optimization of economic and emission dispatch is a power system optimization problem that determines the optimal generation schedule for generators in the system, minimizing both the total generation cost and total emissions of pollutants. To address the dual objectives of cost minimization and emission reduction into a single objective, the weighted sum approach is used [41]. The weighted sum method can be represented as in (18). F refers to the objective function, p refers to the number of objectives ($p = 1,2, \ldots, k$), w_p refer as the real-value weighting with $0 \leq w_p \leq 1$ and $\sum w_p = 1$.

$$F = \sum_{p=1}^{k} w_p F_p \tag{18}$$

The weighted sum-based method converts the multi-objective optimization problem into a single-objective optimization problem. In this method, weights are assigned to each objective, and the optimal solution is obtained by linearly combining the respective objective function values. In the third case, two primary goals are considered: reducing generation costs and reducing generation emissions. Both objectives are considered equally important; hence, each is allocated a weight of 0.5, essentially treating them equally.

Table 7. Comparisons of generation costs with emissions for the IEEE 26-bus (case 3)

Algorithm	Costs with Emissions			
	Mean	SD	Min (Best)	Max (Worst)
ABRO	**15749.0905**	**9.25E-12**	**15749.0905**	**15749.0905**
ABC	15749.4444	1.85E+00	15749.0905	15759.2313
ARO	15797.1685	2.88E+01	15762.7517	15876.3179
JS	15761.5552	1.02E+01	15751.6845	15801.6833
CSA	16128.5023	1.16E+02	15898.6975	16318.4441

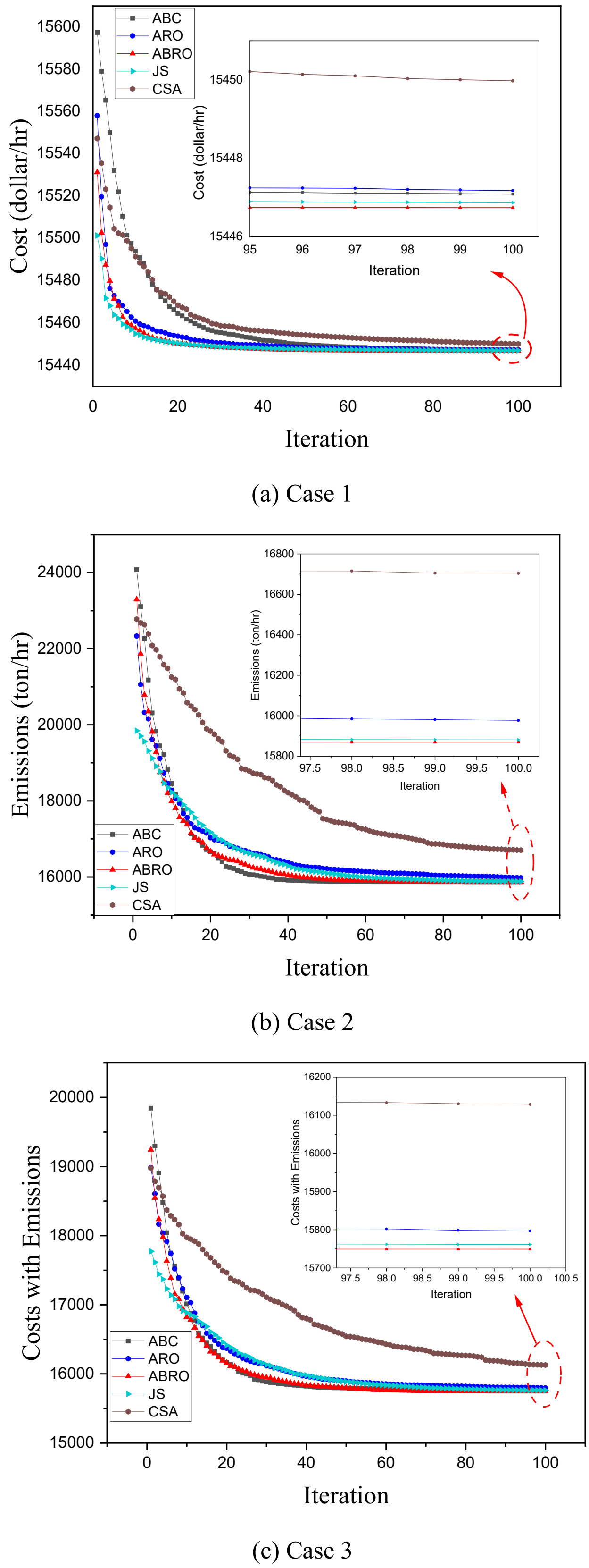

Figure 3. Convergence curves for the IEEE 26-bus system

The outcomes of addressing economic and emission considerations using a weighted sum approach are presented in Table 7. In Table 7, ABRO consistently demonstrates the lowest mean (15749.0905) and the smallest SD, and successfully finds the best Min compared to the other algorithms in optimizing the generation costs with emission. The WSRT is conducted, revealing that the ABRO algorithm significantly outperforms its competitors for weighted sum economic and emission dispatch on the IEEE 26-bus system. Specifically, the median ABRO, $Mdn = 15749.09$ is statistically significantly lower than the median ARO, $Mdn = 15793.84$, $z = -4.78$, $p = 2.00E - 06$. Besides, the median ABRO, $Mdn = 15749.09$ is statistically significantly lower than the median JS, $Mdn = 15759.08$, $z = -4.78$, $p = 2.00E - 06$. Also, the median ABRO, $Mdn = 15749.09$ is statistically significantly lower than the median CSA, $Mdn = 16156.96$, $z = -4.78$, $p = 2.00E - 06$. Lastly, the ABRO is statistically significantly lower than CSA with $z = -4.78$ and $p = 2.00E - 06$. Figure 3 illustrates the convergence curves for three cases: generation costs, generation emissions, and combined generation costs and emissions. As shown in Figure 3, ABRO demonstrates its superior convergence performance and achieves the lowest mean value at iteration 100 for these three cases.

5. CONCLUSION

This study introduces ABRO, a hybrid algorithm that combines the employed bee and random hiding phases from ABC and ARO. Hybridization significantly improved ABRO's performance, where ABRO demonstrated improved accuracy and convergence speed in finding solutions. This research evaluated ABRO's performance against ABC, ARO, CSA, and JS across four benchmark functions and the IEEE 26-bus system. ABRO consistently outperformed the others, achieving the lowest mean, standard deviation, and minimum values, demonstrating its robustness and efficacy. Applied to the IEEE 26-bus system, ABRO reduced generation costs, emissions, and their combined impact, surpassing other algorithms. Using the ABRO algorithm, this study achieved an average rate of 15,446.7425 dollars/hr for economic dispatch, 15,869.7847 tons/hr for emission dispatch, and 15,749.0905 for combined economic and emission dispatch. WSRT confirmed ABRO's superiority except in emission dispatch, where it did not significantly outperform ABC. However, this research was limited to four benchmark functions and the IEEE 26-bus system. Future studies could explore additional benchmark functions and applications. Moreover, integrating Pareto optimization techniques into ABRO could enhance its ability to solve multi-objective problems, resulting in a set of non-dominated solutions representing the trade-offs between conflicting objectives.

ACKNOWLEDGEMENT

The authors would like to express their sincere gratitude to the Centre for Research and Innovation Management (CRIM), UTeM, and Universiti Teknikal Malaysia Melaka (UTeM) for funding this research paper.

REFERENCES

[1] Mahmoud M. El-Halwagi, Ed., "Overview of optimization," in process system engineering, vol. 7, Academic Press, 2006, pp. 285–314. doi: 10.1016/S1874-5970(06)80012-3.

[2] C. Peel and T. K. Moon, "Algorithms for Optimization [Bookshelf]," IEEE Control Syst., vol. 40, no. 2, pp. 92–94, Apr. 2020, doi: 10.1109/MCS.2019.2961589.

[3] A. P. Engelbrecht, Computational Intelligence, Second. Wiley, 2007. doi: 10.1002/9780470512517.

[4] D. Karaboga, "An idea based on honey bee swarm for numerical optimization., Technical Report - TR06" in Erciyes University, 2005, pp. 1–10.

https://doi.org/10.31436/iiumej.v26i3.3800

[5] J.-S. Chou and D.-N. Truong, "A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean," Appl. Math. Comput., vol. 389, p. 125535, Jan. 2021, doi: 10.1016/j.amc.2020.125535.

[6] 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," Eng. Appl. Artif. Intell., vol. 114, p. 105082, Sep. 2022, doi: 10.1016/j.engappai.2022.105082.

[7] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, Apr. 1997, doi: 10.1109/4235.585893.

[8] Ş. Öztürk, R. Ahmad, and N. Akhtar, "Variants of Artificial Bee Colony algorithm and its applications in medical image processing," Appl. Soft Comput., vol. 97, p. 106799, Dec. 2020, doi: 10.1016/j.asoc.2020.106799.

[9] M. Ettappan, V. Vimala, S. Ramesh, and V. T. Kesavan, "Optimal reactive power dispatch for real power loss minimization and voltage stability enhancement using Artificial Bee Colony Algorithm," Microprocess. Microsyst., vol. 76, p. 103085, Jul. 2020, doi:

[10] Z. Wang, H. Ding, B. Li, L. Bao, and Z. Yang, "An Energy Efficient Routing Protocol Based on Improved Artificial Bee Colony Algorithm for Wireless Sensor Networks," IEEE Access, vol. 8, pp. 133577–133596, 2020, doi: 10.1109/ACCESS.2020.3010313.

[11] A. Sharma, A. Sharma, S. Choudhary, R. K. Pachauri, A. Shrivastava, and D. Kumar, "A review on artificial bee colony and its engineering applications," J. Crit. Rev., vol. 7, no. 11, pp. 4097–4107, 2020, doi: 10.31838/jcr.07.11.558.

[12] M. Zhang, Y. Pan, J. Zhu, and G. Chen, "ABC-TLBO: A Hybrid Algorithm Based on Artificial Bee Colony and Teaching-Learning-Based Optimization," in 2018 37th Chinese Control Conference (CCC), IEEE, Jul. 2018, pp. 2410–2417. doi: 10.23919/ChiCC.2018.8483829.

[13] A. Yurtkuran and E. Emel, "An adaptive artificial bee colony algorithm for global optimization," Appl. Math. Comput., vol. 271, pp. 1004–1023, Nov. 2015, doi: 10.1016/j.amc.2015.09.064.

[14] J. Luo, Q. Liu, Y. Yang, X. Li, M. Chen, and W. Cao, "An artificial bee colony algorithm for multi-objective optimisation," Appl. Soft Comput., vol. 50, pp. 235–251, Jan. 2017, doi: 10.1016/j.asoc.2016.11.014.

[15] W. W. Lee and M. R. Bin 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), IEEE, Jun. 2023, pp. 298–303. doi: 10.1109/I2CACIS57635.2023.10193351.

[16] A. Askarzadeh, "A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm," Comput. Struct., vol. 169, pp. 1–12, Jun. 2016, doi: 10.1016/j.compstruc.2016.03.001.

[17] M. A. Bagherian and K. Mehranzamir, "A comprehensive review on renewable energy integration for combined heat and power production," Energy Convers. Manag., vol. 224, p. 113454, Nov. 2020, doi: 10.1016/j.enconman.2020.113454.

[18] M. Gent and J. Lamont, "Minimum-Emission Dispatch," IEEE Trans. Power Appar. Syst., vol. PAS-90, no. 6, pp. 2650–2660, Nov. 1971, doi: 10.1109/TPAS.1971.292918.

[19] B. Dey, S. K. Roy, and B. Bhattacharyya, "Solving multi-objective economic emission dispatch of a renewable integrated microgrid using latest bio-inspired algorithms," Eng. Sci. Technol. an Int. J., vol. 22, no. 1, pp. 55–66, Feb. 2019, doi: 10.1016/j.jestch.2018.10.001.

[20] H. Jia et al., "Improved artificial rabbits algorithm for global optimization and multi-level thresholding color image segmentation," Artif. Intell. Rev., vol. 58, no. 2, 2025, doi: 10.1007/s10462-024-11035-3.

[21] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic Algorithms: A Comprehensive Review," in Computational Intelligence for Multimedia Big Data on the Cloud

with Engineering Applications, Elsevier, 2018, pp. 185–231. doi: 10.1016/B978-0-12-813314-9.00010-4.

[22] J. Yao Lee, M. Ruzaini Hashim, and M. O. Tokhi, “Artificial Bee Colony Optimization Algorithm with Flexible Manipulator System,” vol. 3, no. 2, pp. 75–82, 2020.

[23] N. B and J. J, “Hybrid ABC-BAT Optimization Algorithm for Localization in HWSN,” Microprocess. Microsyst., p. 104024, Jan. 2021, doi: 10.1016/j.micpro.2021.104024.

[24] Y. Yan, J. Chen, H. Huang, and S. Yang, “A hybrid algorithm for three-dimensional loading capacitated vehicle routing problems with time windows,” in Proceedings - 2020 7th International Conference on Information Science and Control Engineering, ICISCE 2020, L. S., D. Y., M. J., and C. Y., Eds., Xiamen University, School of Informatics, Xiamen City, China: Institute of Electrical and Electronics Engineers Inc., 2020, pp. 1534–1540. doi: 10.1109/ICISCE50968.2020.00304.

[25] C. Mala, V. Deepak, S. Prakash, and S. Lashmi Srinivasan, “A Hybrid Artificial Bee Colony Algorithmic Approach for Classification Using Neural Networks,” in EAI/Springer Innovations in Communication and Computing, H. A., R. A., M. S., and C. M.-Y., Eds., National Institute of Technology, Tiruchirappalli, Tamil Nadu, India: Springer Science and Business Media Deutschland GmbH, 2021, pp. 339–359. doi: 10.1007/978-3-030-47560-4_28.

[26] P. Kumar, B. J. Sowmya, A. Kanavalli, and A. Vincent, “ABC-DE-WOA: A New Hybrid Algorithm for Optimization Problems,” in 2022 4th International Conference on Circuits, Control, Communication and Computing (I4C), 2022, pp. 501–506.

[27] M. Ramachandran, S. Mirjalili, M. Malli Ramalingam, C. A. R. Charles Gnanakkan, D. S. Parvathysankar, and A. Sundaram, “A ranking-based fuzzy adaptive hybrid crow search algorithm for combined heat and power economic dispatch,” Expert Syst. Appl., vol. 197, 2022, doi: 10.1016/j.eswa.2022.116625.

[28] D. Ustun, A. Toktas, U. Erkan, and A. Akdagli, “Modified artificial bee colony algorithm with differential evolution to enhance precision and convergence performance,” Expert Syst. Appl., vol. 198, 2022, doi: 10.1016/j.eswa.2022.116930.

[29] M. Elshahed, M. A. Tolba, A. M. El-Rifaie, A. Ginidi, A. Shaheen, and S. A. Mohamed, “An Artificial Rabbits’ Optimization to Allocate PVSTATCOM for Ancillary Service Provision in Distribution Systems,” Mathematics, vol. 11, no. 2, p. 339, Jan. 2023, doi: 10.3390/math11020339.

[30] B. Gülmez, “Stock price prediction with optimized deep LSTM network with artificial rabbits optimization algorithm,” Expert Syst. Appl., vol. 227, p. 120346, Oct. 2023, doi: 10.1016/j.eswa.2023.120346.

[31] A. O. Alsaiari, E. B. Moustafa, H. Alhumade, H. Abulkhair, and A. Elsheikh, “A coupled artificial neural network with artificial rabbits optimizer for predicting water productivity of different designs of solar stills,” Adv. Eng. Softw., vol. 175, p. 103315, Jan. 2023, doi: 10.1016/j.advengsoft.2022.103315.

[32] A. G. Hussien, H. A. El-Sattar, F. A. Hashim, and S. Kamel, “Enhancing optimal sizing of stand-alone hybrid systems with energy storage considering techno-economic criteria based on a modified artificial rabbits optimizer,” J. Energy Storage, vol. 78, no. October 2023, p. 109974, Feb. 2024, doi: 10.1016/j.est.2023.109974.

[33] F. Li, H. Xu, and F. Qiu, “Modified artificial rabbits optimization combined with bottlenose dolphin optimizer in feature selection of network intrusion detection,” Electron. Res. Arch., vol. 32, no. 3, pp. 1770–1800, 2024, doi: 10.3934/ERA.2024081.

[34] X. Zhou, J. Lu, J. Huang, M. Zhong, and M. Wang, “Enhancing artificial bee colony algorithm with multi-elite guidance,” Inf. Sci. (Ny)., vol. 543, pp. 242–258, Jan. 2021, doi: 10.1016/j.ins.2020.07.037.

[35] D. Bajer and B. Zorić, “An effective refined artificial bee colony algorithm for numerical optimisation,” Inf. Sci. (Ny)., vol. 504, pp. 221–275, Dec. 2019, doi: 10.1016/j.ins.2019.07.022.

https://doi.org/10.31436/iiumej.v26i3.3800

[36] D. Kumar and K. K. Mishra, “Co-variance guided Artificial Bee Colony,” Appl. Soft Comput., vol. 70, pp. 86–107, Sep. 2018, doi: 10.1016/j.asoc.2018.04.050.

[37] S. W. Scheff, “Nonparametric Statistics,” in Fundamental Statistical Principles for the Neurobiologist, Academic Press, 2016, pp. 157–182. doi: 10.1016/B978-0-12-804753-8.00008-7.

[38] D. V. Tien, P. Hawliczek, R. Gono, and Z. Leonowicz, “Analysis and modeling of STATCOM for regulate the voltage in power systems,” in 2017 18th International Scientific Conference on Electric Power Engineering (EPE), IEEE, May 2017, pp. 1–4. doi: 10.1109/EPE.2017.7967358.

[39] P. P. S. Saputra, F. D. Murdianto, R. Firmansyah, and K. Widarsono, “Economic Dispatch in IEEE 26 Bus System using Quantum Behaved Particle Swarm Optimization,” in 2020 International Conference on Applied Science and Technology (ICAST), IEEE, 2020, pp. 54–58. doi: 10.1109/iCAST51016.2020.9557625.

[40] M. R. M. Ridzuan, E. E. Hassan, A. R. Abdullah, and A. F. A. Kadir, “Sustainable environmental economic dispatch optimization with hybrid metaheuristic modification,” Indones. J. Electr. Eng. Comput. Sci., vol. 11, no. 1, pp. 161–168, 2018, doi: 10.11591/ijeecs.v11.i1.pp161-168.

[41] M. R. Hashim, “Improved spiral dynamics and artificial bee colony algorithms with application to engineering problems,” University of Sheffield, 2018.