Hindawi
Discrete Dynamics in Nature and Society
Volume 2023, Article ID 9964744, 10 pages
https://doi.org/10.1155/2023/9964744

Research Article

A Comparative Study of Particle Swarm Optimization and Artificial Bee Colony Algorithm for Numerical Analysis of Fisher's Equation

Geeta Arora,[1] **Kiran Bala**,[1] **Homan Emadifar**,[2,3] **and Masoumeh Khademi**[2]

[1]*Department of Mathematics, Lovely Professional University, Jalandhar, Punjab, India*
[2]*Department of Mathematics, Hamedan Branch, Islamic Azad University, Hamedan, Iran*
[3]*MEU Research Unit, Middle East University, Amman, Jordan*

Correspondence should be addressed to Homan Emadifar; homan_emadi@yahoo.com

Received 1 August 2023; Revised 6 October 2023; Accepted 9 October 2023; Published 18 October 2023

Academic Editor: Rodica Luca

The aim of this research work is to obtain the numerical solution of Fisher's equation using the radial basis function (RBF) with pseudospectral method (RBF-PS). The two optimization techniques, namely, particle swarm optimization (PSO) and artificial bee colony (ABC), have been compared for the numerical results in terms of errors, which are employed to find the shape parameter of the RBF. Two problems of Fisher's equation are presented to test the accuracy of the method, and the obtained numerical results are compared to verify the effectiveness of this novel approach. The calculation of the error norms leads to the conclusion that the performance of PSO is better than the ABC algorithm to minimize the error for the shape parameter in a given range.

1. Introduction

To obtain numerical solutions with the optimized results in a variety of scientific and engineering disciplines, researchers have developed various methods. Algorithms based on swarm intelligence have great potential in the field of numerical optimization, according to researchers Yagmahan and Yenisey [1], Eberhart and Kennedy [2], Price et al. [3], and Vesterstrom and Thomsen [4]. Swarm intelligence-based algorithms [5] and evolution [6] are two significant categories of population-based algorithms in the field of optimization. Optimization is the process of increasing the advantages of a mathematical model or function while minimizing its disadvantages. It is a combination of techniques that enables us to improve the output of the system. The primary goal of optimization is to find an optimal or nearly ideal solution with the least amount of computing work. The key to find the solution is to optimize parameters connected to a mathematical model.

Several research fields adopt optimization approaches for numerical simulation of various linear and nonlinear partial differential equations (PDEs) and also for optimizing the parameters related to problematic models. There are some well-known approaches of optimization such as ant colony optimization (ACO) [7] is one of the optimization algorithms that is a meta-heuristic algorithm inspired by the foraging behaviour of ants and how they find the shortest path between their nest and a food source. While it is commonly used for combinatorial optimization problems, it can also be adopted for numerical solutions of PDEs. Particle swarm optimization (PSO) [2] is also a heuristic algorithm inspired by the social behaviour of birds and fish, where individuals in a group (particles) cooperate and communicate to find optimal solutions to a problem. Bacteria foraging optimization (BFO) [8] is stimulated by the foraging behaviour of Escherichia coli (E. coli) bacteria that mimics the way bacteria forage for nutrients in their environment to find the optimal solution for a given optimization problem. When adapting BFO for the numerical solution of PDEs, it can effectively explore the solution space for parameter settings that yield accurate and efficient numerical solutions to PDEs. These nature-inspired meta-

heuristic optimization algorithms have recently gained popularity for developing an effective search algorithm.

Exploration and exploitation are two major determinants for the development of successful optimization algorithms for search mechanisms. A meta-heuristic optimization algorithm effectively explores the solution space, balancing between exploration (global exploration) and exploitation (local refinement) to find near-optimal or optimal solutions for a variety of optimization problems. Exploration involves the search for new, unexplored regions of the solution space. It aims to discover potential solutions that might be superior to the current ones. Exploitation involves focusing on known promising regions of the solution space to improve the quality of solutions. It aims to refine and optimize the current solutions based on the information available. Researchers are motivated to develop such population-based optimization algorithms because of the abundance of natural resources. These population-based optimization methods assess fitness and provide almost perfect solutions to complex optimization problems.

Swarm intelligence (SI) is a field of study inspired by the collective behaviour of social insect colonies and other animal societies. It explores the principles and models of behaviour that emerge from the interactions of simple individuals within a group. The connection between SI and optimization lies in leveraging the collective behaviour observed in natural swarms to create effective optimization algorithms and strategies. SI uses social insect behaviour to create algorithms or distributed problem-solving tools, according to Bonabeau et al. [9]. Bonabeau studied only social insects such as termites, bees, wasps, and ants. Social species first developed swarm intelligence through trial and error. It simulates self-organizing swarms of interacting agents. An immune system, ant colony, or bird flock are swarm systems. Bees swarming around their hives illustrate swarm intelligence. Based on honey bee swarm social intelligence, the artificial bee colony (ABC) algorithm first described by Karaboga [10] in 2005, and in 1995 [2], Kennedy and Russell proposed PSO for solving numerical optimization problems. Honey bees' search for nutritious food influenced the procedure of ABC, and the process of PSO was attracted by the behaviour of social animals. These population-based stochastic search methods are simple and fast. These methods also solve complex, continuous, and unbounded optimization problems with multimodal or unimodal issues.

SI-based meta-heuristic algorithms are popular for solving various optimization models as in [11] employed a novel adaptive artificial bee colony (A-ABC) algorithm that can select the best search equation based on the current situation in order to more precisely predict the transport energy demand (TED), in [12] four different meta heuristic algorithms used for natural gas demand forecasting based on meteorological indicators in Turkey, [13] proposed a new modified artificial bee colony (M-ABC) method that can more precisely calculate Turkey's energy usage by adaptively choosing an optimal search equation and many more examples are there in biology, physics, evolution, and human behaviour that inspire nature-inspired algorithms that include ant colony optimization, artificial bee colony, the firefly method, particle swarm optimization, brain storm optimization, sine and cosine algorithms, and genetic algorithms. With inspiration from SI, many researchers applied these meta-heuristic algorithms for the numerical simulation of various PDEs by optimizing the solution space.

For numerically simulating ordinary and partial differential equations, RBF has proven to be a useful basis function. Numerical solutions to a nonlinear partial differential equation are found in this study by employing a mesh-free method based on radial basis functions (RBFs) with ABC and PSO optimization techniques. Both optimization strategies are used to determine the shape parameter (ϵ) related to RBF.

The reaction-diffusion equation is one of the most intriguing equations in physical processes. We concentrate on the form of reaction-diffusion, which is known as Fisher's equation.

$$v_t - \mathrm{sv}_{xx} = F(v), \tag{1}$$

whose boundary and initial conditions are as follows:

$$\begin{aligned} &v(x,0) = v_0(x) \in [0,1], x \in (-\infty, \infty), \\ &\lim_{x \longrightarrow -\infty} v(x,t) = 1, \lim_{x \longrightarrow \infty} v(x,t) = 0, \\ &\lim_{x \longrightarrow \pm\infty} v(x,t) = 0. \end{aligned} \tag{2}$$

Many chemical and biological processes use $F(v) = v(1 - v)$. Fisher [14] introduced this equation to demonstrate a beneficent gene's kinetic advance rate. Fisher's equation shows population evolution through opposing physical phenomena. Fisher's equation dominates genetics, tissue engineering, growth models, and more in science and engineering. Fisher's equation was first simulated using the pseudospectral method developed by Gazdag and Canosa in 1974 [15]. Since then, many different approaches have been developed to solve it, such as the Petrov–Galerkin finite element method processed by Tang and Weber [16], the Tanh method by Wazwaz [17], and the homotopy analysis method proposed by Tan et al. [18]. Other methods that have been used to solve this problem include the alternating iterative method by Sahimi and Evans [19], the central finite difference algorithm by Hagstrom and Keller [20], the explicit and implicit finite difference algorithms by Parekh and Puri [21], the collocation of cubic B-splines by Mittal and Arora [22], and the pseudo spectral approach by Bhatia and Arora [23].

In this paper, the ABC and PSO algorithms with RBF applied to Fisher's partial differential equation are used to find the best shape parameter of RBF by minimizing the error, and the RBF-PS method is used for numerical simulation of Fisher's equation by converting it into an ordinary differential equations (ODEs) system. MATLAB is used for optimizing the parameter ε and for numerical approximation of Fisher's equation. In this work, the results are obtained by the present hybrid approach in the form of error norms-L_∞, L_2, and L_{rms} and shape parameter values at different time intervals, which are more comparable to the

results available in the literature. The errors obtained by PSO are less as compared to the errors obtained by the ABC algorithm; thus, PSO values of shape parameter are good in comparison to the ABC's results.

The structure of the paper is as follows. Section 2 describes the ABC algorithm in detail, with pseudocodes for all the phases, and also presents the complete process of ABC. Section 3 presents the explanation of the PSO algorithm with pseudocode, and the obtained results of the two problems of Fisher's equation by the novel hybrid approach are discussed and compared in Section 4. Section 5 concludes the present article with the details of key findings and the future scope.

2. Artificial Bee Colony (ABC) Algorithm

The artificial bee colony (ABC) algorithm was invented by Karaboga [10] in 2005. The algorithm seeks the nectar-rich flower region (optimal solution). Employed bees (busy), onlookers, and scouts structured the swarms in the ABC algorithm. The swarm that finds the best food supply is more likely to be followed by the others. ABC remembers a user's best location, as in the process of PSO. The bee travels to a new location and evaluates it in comparison to its current favourite place. If the new site is better, the old one is forgotten and the new one is remembered. The recollection remains unchanged. The ABC algorithm begins with deploying bees in the beginning and dispersing them in different locations. All employed bees are those actively foraging for nectar or pollen. Employed bees bring food information back to the hive and share it with curious bees. Onlooker bees wait in the hive for scout bees to report new food sources. To share food supply information, employed bees danced in a designated area. The dancing bee's dance depends on the food source's nectar. Onlooker bees monitor the dance and choose a food source depending on its trustworthiness. Before returning to the beehive, employed bees communicate the information with onlooker bees, which chose the most likely to follow. So, better food sources attract more bees than poor ones. When a food supply is exhausted, all employed bees become scouts. Scout and employed bees pursue exploitation and exploration processes.

In this algorithm, each food source is a potential solution to the problem, and its nectar amount indicates the fitness value's estimate of its quality. For each possible food source, there is exactly one busy bee, and the total number of food sources is equal to the total number of employed bees. Based on the following related probability value p_i defined, an observer bee selects a food source.

$$p_i = \frac{\text{fit}_i}{\sum_{t=1}^{\text{Np}} \text{fit}_t}, \tag{3}$$

where the fitness function (objective function value) of i^{th} solution is fit_i evaluated by employed bees that are proportional to the optimal solution and Np is the number of food sources. Due to the process, employed bees share the whole information with the onlooker bees. Utilizing the probability values of the employed bees, a roulette wheel selection method is used. The likelihood of being chosen by onlooker bees increases with the amount of nectar a worker bee shares. With the aid of the chosen employed bee, the onlooker bee travels to a new site V_i using the following formula:

$$V_i = X_i + \varphi_i\left(X_i - X_j\right). \tag{4}$$

Here, the present position is denoted by X_i, employed bee selection is represented by X_j, and φ_i is selected randomly from −1 to 1 for finding the food sources in the region of X_j. Any bee that is not capable of finding a better food source after several iterations is replaced with a scout bee X_k. The scout bee summoned to replace the unsuccessful bee flies about any random or uncharted area to investigate its surroundings using the following equation:

$$X_k = \text{lb} + \varphi_i\,(\text{ub} - \text{lb}). \tag{5}$$

Here, ub and lb are upper and lower bounds, respectively, and randomness lies between 0 and 1. Again, the process is repeated with employed bees. The parameter limit (the number of cycles) is used for a location that cannot be enhanced during the fixed cycles. These three steps of the algorithm ABC, with their pseudocodes, are as follows.

2.1. Employed Bee Phase. For the generation of new solutions in the employed bee phase, the following are some points to be remembered:

(1) The number of employed bees is equal to the number of food sources

(2) There is an opportunity for all of the solutions

(3) A partner is randomly selected for the generation of a new solution

(4) The current solution and partner should not be the same

(5) Modification of a randomly selected variable is important for the generation of a new solution

$$X_{\text{new}}^j = X^j + \varnothing\left(X^j - X_p^j\right). \tag{6}$$

where $X_p^j = j^{\text{th}}$ variable of p^{th} solution
$X_{\text{new}}^j = j^{\text{th}}$ variable of a new solution
$\varnothing$ = the random variable lies between −1 to 1
$X^j = j^{\text{th}}$ variable of the current solution

(6) Boundedness of the newly generated solution

$$\begin{aligned} X_{\text{new}}^j &= \text{lb if } X_{\text{new}}^j < \text{lb}, \\ X_{\text{new}}^j &= \text{ub if } X_{\text{new}}^j > \text{ub}. \end{aligned} \tag{7}$$

After generating a new solution within the boundaries, we must assess the objective function value and find its fitness via the relation. To update the current solution, we greedily select the newly generated solution. We track trial failures for each solution. If the new solution is worse, we

increase the trial by one; if it is better, we reset it. Now, this better solution counts in the population. The pseudocode for the employed bee phase is given as follows:

Input = objective function, p, fit, trial, lb, ub, Np = s/2 = no. of food source/employed or onlooker bees, s = swarm size

for $i = 1$ to Np

Selected a partner (p) randomly such that i is not equal to p

Selected a variable (j) randomly and update j^{th} variable

$X^j_{\text{new}} = X^j + \varnothing(X^j - X^j_p)$

Bound modified j (x^j_{new})

Evaluate objective function (f_{new}) and fitness (fit_{new})

Accept X^{new}, if fit_{new} greater then fitness and Set trial = 0. Else increase trial by one.

End.

2.2. Onlooker Bee Phase. In this phase, there is a condition of probability for bees to exploit a particular food source. As we know the fitness of each food source, for each food source, we calculate the probability which is determined as follows:

$$p_i = \frac{\text{fit}_i}{\sum_{t=1}^{\text{Np}} \text{fit}_t}, \tag{8}$$

where fit_i and prob_i are fitness and probability of the i^{th} solution, respectively. A solution with a higher fitness value will have a higher probability. The pseudocode for the onlooker bee phase is as follows:

Input = obj. Function, f, p, fit, trial, lb, ub, Np = s/2, prob.

Set $m = 0$ & $n = 1$(m is onlooker bee & n is food source)

While m less then Np

Generate random no. r

If r less then prob.

Randomly select a partner (p) s.t. n is not equal to p

Selected a variable (j) randomly and update j^{th} variable

$X^j_{\text{new}} = X^j + \varnothing(X^j - X^j_p)$.

Bound modified j (x^j_{new})

Evaluate objective function (f_{new}) and fitness (fit_{new})

Accept X^{new}, if fit_{new} greater then fitness and Set trial = 0. Else increase trial by one

$m = m + 1$

End

$n = n + 1$

Modify $n = 1$ (if $n >$ Np)

End.

2.3. Scout Bee Phase. As every solution is associated with an individual trial, we need to specify a parameter limit that is a user-specified integer value. For entering the solution in this phase, the value of trial should be greater than the limit and trial of the abandoned solution is reset to zero. Not every solution passes through the scout phase. The limit of solution can be calculated as Np $*$ d because of d-dimensional problem space. Scout phase can occur only when trial counter of at least one solution is greater than the limit. The pseudocode for the scout bee phase is as follows:

Input = obj. Function, p, fit, trial, lb, ub, limit.

Identify food source (t) whose trial > limit.

Replace X_t with p as $X_t = \text{lb} + \varphi_i(\text{ub} - \text{lb})$

Evaluate the objective function (f_t) and assign fitness (fit_t)

2.4. Complete Pseudocode for the ABC Algorithm. ABC initializes bee swarms and repeats until stopping criteria are met. ABC optimizes iteratively. Employed and onlooker bees agree on exploitation, while the process of exploration is performed with scout bees.

Initialisation of input parameters = fit, t, lb, ub, limit, Np.

(1) Initialization of population parameter (p) randomly.

(2) Calculate the objective function value (f) and the fitness value (fit).

(3) Locate trial counter = 0.

(4) for $i = 1$ to t

Evaluate the employed bee stage.

Determined probability

Apply the onlooker bee phase for generating food sources.

Memorise the best food source.

If trial > limit

Enter into Scout bee stage

End

End

3. Particle Swarm Algorithm (PSO)

Kennedy and Eberhart [4] invented PSO as a popular swarm intelligence technique in 1995. PSO is effective in solving optimization problems by changing the paths of particles. Particle movement is mostly stochastic and deterministic. Social animals, flocks of birds, marine animal communities, and swarms influence this optimization strategy. Swarms are population particles that transfer information to improve the search solution and discover the global optimum in this nature-based swarm optimization technique. Each particle's position is their best experience. Particles' worldwide best position is their finest experience. When it finds a target better than all others, a particle alters its best position. Each

n-particle has a new best solution during iterations. This method finds the best solution among all possible solutions. This process continues until the set iteration or the goal is not met. In this algorithm, the velocity U_i^{k+1} and the position X_i^{k+1} of the i^{th} particle are updated as follows:

$$\begin{aligned} V_i^{k+1} &= V_i^k + a_1 * \text{random} * \left(P_{\text{best}_i} - X_i^k\right) + a_2 * \text{random} * \left(G_{\text{best}} - X_i^k\right), \\ X_i^{k+1} &= X_i^k + V_i^{k+1}, \end{aligned} \tag{9}$$

where V_i^k is the velocity of particle i at k^{th} iteration, a_1, a_2 are real parameters, random is a random number, whose value lies between 0 and 1, X_i^k is the i^{th}-particle position at k^{th}-iteration, P_{best_i} is the personal best position of particle i, G_{best} is the global best position of whole search space.

3.1. Pseudocode for the PSO Algorithm. Enter values of parameters: a_1, a_2, fitness, lb, ub, Np, and t.

(1) Initialization of population (P) randomly and velocity U_i of particle i.
(2) Calculate the objective function (f).
(3) Assign P_{best_i} as P and f_{best} as f.
(4) Evaluate finest fitness solution and allocate the solution to G_{best} and fitness to f_{best}.

```
for t = 1tot
for i = 1 to Np
  Determine the velocity U_i
  Determine the new position (X_i)
  Bound X_i
  Find objective function value
  Update the population by including X_i & f_i
  Update P_best and f_pbest
  Update G_best and f_gbest
End
End
```

PSO is a computational technique that iteratively optimizes a problem to reduce error. It is a statistical method used to determine parameter values. To find the optimal answer to an optimization problem, the particles communicate, share their knowledge, and follow a simple rule. It is an innovative method for evaluating the best shape parameter value of RBF using the nonlinear partial differential equation. It is a global search optimization strategy and offers numerous characteristics in the parameter space.

4. Numerical Applications

In this section, the numerical solution of Fisher's equation using the RBF pseudospectral method is obtained for the applications of the above novel approach. Two problems of Fisher's equation are solved numerically using the present approach, and their results are calculated using the different error norms: L_∞, L_2, L_{rms}, absolute errors along with the shape parameter values. A comparison of the obtained results is presented for the effectiveness and applicability of the proposed method. First, derivatives are approximated using RBF, and then solutions are determined by MATLAB software with version R2022a using both the algorithms with the intel (R) Pentium processor and the window 7 operating system. The cubic radial pattern basis is taken as a basis function for the numerical simulation of the equation. The initial parameters for both the algorithms are as follows: number of decision variables = 1 and values of lower and upper bounds of decision variables are 0 and 1, respectively. Same sets of values are used for obtaining the shape parameter that minimizes the errors in both the problems of Fisher's equation.

The process of the proposed approach is shown graphically in Figure 1.

Following are the formulae used for the calculation of the errors:

$$\begin{aligned} \text{Absolute error} &= \left|v_{\text{exact}}(x_i,t) - v(x_i,t)\right|; \quad 1 \le i \le N, \\ L_\infty &= \max\left(\left|v_{\text{exact}}(x_i,t) - v(x_i,t)\right|\right); \quad 1 \le i \le N, \\ L_2 &= \sqrt{h \sum_{i=1}^{N} \left|v_{\text{exact}}(x_i,t) - v(x_i,t)\right|^2}, \\ L_{\text{rms}} &= \sqrt{\frac{1}{\text{N}} \sum_{i=1}^{\text{N}} \left|v_{\text{exact}}(x_i,t) - v(x_i,t)\right|^2}. \end{aligned} \tag{10}$$

4.1. Test Problem 1. Taking one dimensional Fisher's equation (1) with $F(v) = v^2(1-v)$; as $v_t = v_{xx} + v^2(1-v)$ on the domain [0, 1] with initial condition and exact solution are, respectively,

$$\begin{aligned} v(x,0) &= \frac{1}{2} - \frac{1}{2}\tanh\left(a + \frac{\sqrt{2}\,x}{4}\right); \\ v(x,t) &= \frac{1}{2} - \frac{1}{2}\tanh\left(a + \frac{\sqrt{2}}{4}\left(x - \frac{1}{\sqrt{2}}\,t\right)\right). \end{aligned} \tag{11}$$

Numerical simulation of problem 1 is carried out by taking $\Delta t = 0.0001$, $N = 21$ at time interval 0.2, 0.5, 1. Table 1 represents the computed error norms: L_∞, L_2, L_{rms} by ABC

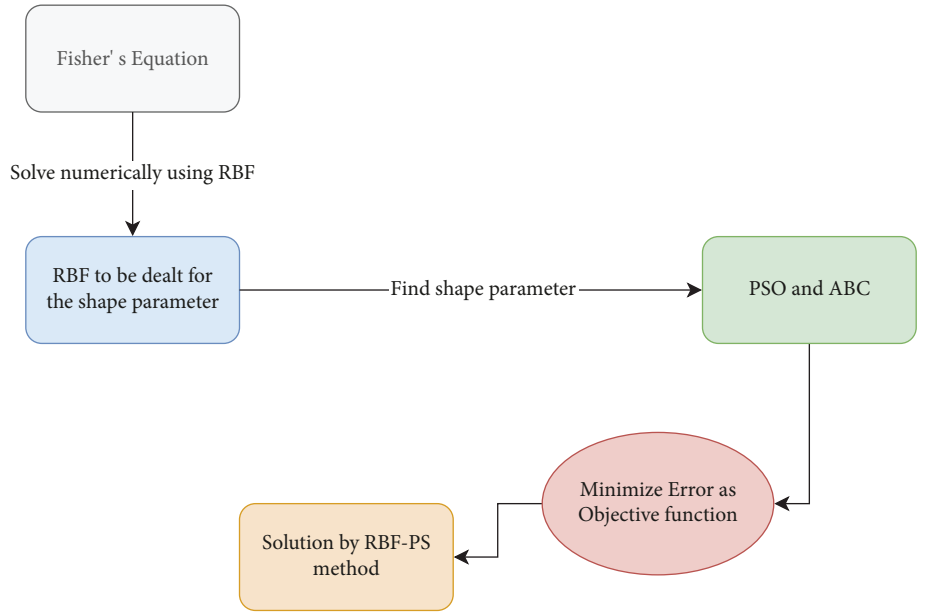

Figure 1: Graphical representation of the proposed approach.

Table 1: Comparison of error norms of problem 1 with $\Delta t = 0.0001$ and $N = 21$ on varied T and iteration = 71.

Time (T)	Methods	L_∞	L_2	L_{rms}
0.2	ABC	$8.4869e-05$	$1.1884e-05$	$1.6641e-06$
	PSO	$1.2639e-05$	$2.7580e-06$	$6.0185e-07$
	[24]	$2.1050e-05$	$4.4750e-03$	$3.6330e-03$
	[25]	$1.0500e-06$	$7.0220e-07$	$1.6040e-07$
0.5	ABC	$7.4960e-03$	$1.6358e-03$	$3.5695e-04$
	PSO	$1.2542e-04$	$2.7369e-05$	$5.9725e-06$
	[24]	1.7360e–05	4.4440e–03	3.8210e–03
	[25]	$1.1240e-06$	$8.4700e-07$	$1.8480e-07$
1.0	ABC	$3.0199e-03$	$6.5900e-04$	$1.4381e-04$
	PSO	$1.3587e-04$	$2.9649e-05$	$6.4700e-06$
	[24]	$1.1030e-05$	$2.6900e-03$	$2.4640e-03$
	[25]	$1.2030e-06$	$9.4990e-07$	$2.1540e-07$

and PSO algorithm and comparison is carried out with results in the literature [24, 25]. It can be seen from the results that the errors are less by the PSO algorithm comparative to the ABC's approach. The obtained results in Table 1 are compared to the results calculated by other numerical methods available in the literature. By optimizing the errors, the shape parameter resulted in a value 0.018062 using PSO and 0.065821 using ABC at which the best errors occur. The values of the shape parameter at different T for $\Delta t = 0.0001$ and $N = 21$ are given in Table 2. Comparative analysis of absolute errors using both the algorithms at $N = 21$ and $\Delta t = 0.0001$ with time 0.01, 0.02, and 0.03 as shown by Table 3 which concluded the errors are less by the PSO algorithm in comparison to ABC. Figure 2 demonstrates the graphical solution for 21 node points with $\Delta t = 0.0001$ on domain [0, 1].

4.2. Test Problem 2. We consider the general Fisher's (1) with $F(v) = v(1 - v^a)$; as $v_t = v_{xx} + v(1 - v^a)$ with domain [0, 1], whose exact solution and initial condition are given as follows, respectively:

$$v(x,t) = \left(\frac{1}{2} + \frac{1}{2}\tanh\left(-\frac{a}{2\sqrt{2a+4}}\left(x - \frac{a+4}{\sqrt{2a+4}}\,t\right)\right)\right)^{(2/a)};$$
$$v(x,t) = \left(\frac{1}{2} + \frac{1}{2}\tanh\left(-\frac{a}{2\sqrt{2a+4}}\,x\right)\right)^{(2/a)}. \tag{12}$$

Table 2: Shape parameter (ε) values of problem 1 with different T at $\Delta t = 0.0001$ and $N = 21$.

Time (T)	ε	
	ABC	PSO
0.2	0.065821	0.065599
0.5	0.077052	0.291291
1.0	0.094752	0.018062

Table 3: Comparison of absolute errors of problem 1 at $N = 11$, $\Delta t = 0.0001$ with varied time T.

X	T					
	0.01	0.02	0.03	0.01	0.02	0.03
	ABC			PSO		
0.1	$7.9581e-07$	$7.1076e-06$	$1.2974e-05$	$1.0359e-08$	$7.9851e-09$	$3.3186e-10$
0.2	$7.2611e-07$	$1.8189e-06$	$2.9827e-06$	$3.2868e-09$	$1.4958e-08$	$7.5032e-08$
0.3	$1.8184e-07$	$1.5313e-06$	$2.2719e-06$	$6.3746e-09$	$4.0134e-09$	$4.8178e-09$
0.4	$5.9365e-08$	$3.5776e-06$	$6.3239e-06$	$3.0546e-09$	$1.3058e-08$	$5.8104e-08$
0.5	$8.8167e-09$	$5.1369e-06$	$8.6968e-06$	$7.1879e-09$	$1.8792e-09$	$6.1799e-08$
0.6	$2.5911e-08$	$5.4938e-06$	$8.5366e-06$	$3.4203e-09$	$1.3243e-08$	$5.1268e-08$
0.7	$4.1140e-08$	$4.9160e-06$	$8.4974e-06$	$1.1938e-10$	$1.1625e-08$	$2.1215e-08$
0.8	$4.3687e-07$	$2.9209e-06$	$6.4768e-06$	$6.5021e-09$	$3.6290e-08$	$8.7129e-08$
0.9	$1.4408e-06$	$6.2425e-07$	$1.8456e-06$	$1.6344e-08$	$5.7144e-08$	$8.0607e-08$

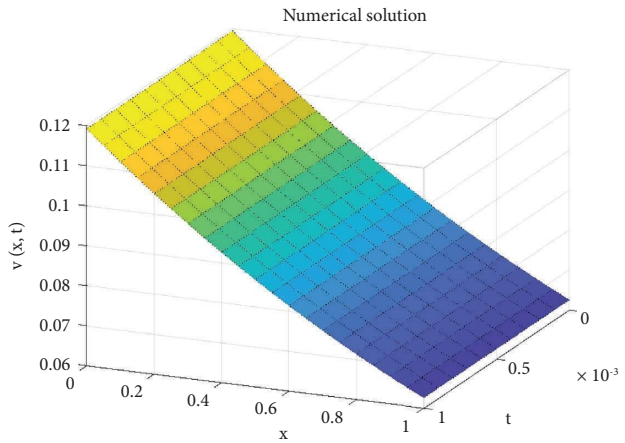

Figure 2: Numerical simulation of problem 1 with $N = 21$, $\Delta t = 0.0001$, and $\Delta t \leq 0.001$.

Table 4: Comparison of numerical and exact solutions of problem 2 at $N = 21$ and $\Delta t = 0.0001$ with time T.

X	T	Numerical solution			Exact solution
		PSO	ABC	[26]	
0.25	0.5	0.3341	0.3301	0.3341	0.3340
	1	0.4559	0.4554	0.4557	0.4557
	2	0.6842	0.6786	0.6839	0.6839
0.5	0.5	0.3057	0.3034	0.3057	0.3057
	1	0.4255	0.4248	0.4255	0.4255
	2	0.6582	0.6536	0.6592	0.6592
0.75	0.5	0.2782	0.2769	0.2783	0.2783
	1	0.3949	0.3944	0.3954	0.3954
	2	0.6308	0.6271	0.6333	0.6333

Table 4 represents the comparison of the results obtained by ABC and PSO algorithm with results in the literature [26] for $a = 1$ with $\Delta t = 0.0001$ at $N = 21$ along with exact solutions. As shown in Table 5, a comparison of different error norms calculated by both the algorithms at $\Delta t = 0.00001$, $N = 21$ with various time intervals 0.001, 0.002,

Table 5: Comparative study of error norms at $\Delta t = 0.00001$ and $N = 21$ with time interval at iteration = 71.

Methods	T	$\mathbf{L}_\infty$	$\mathbf{L}_2$	$\mathbf{L}_{\text{rms}}$
ABC	0.001	$3.54e-10$	$1.06e-10$	$3.22e-11$
PSO	0.001	$1.44e-11$	$3.15e-12$	$6.87e-13$
[23]	0.001	$4.13e-11$	$1.45e-08$	$3.16e-09$
ABC	0.002	$1.23e-09$	$3.72e-10$	$1.12e-10$
PSO	0.002	$4.18e-11$	$9.13e-12$	$1.99e-12$
[23]	0.002	$2.97e-11$	$3.98e-08$	$8.68e-09$
ABC	0.003	$2.26e-09$	$6.81e-10$	$2.05e-10$
PSO	0.003	$8.01e-11$	$1.74e-11$	$3.81e-12$
[23]	0.003	$1.00e-11$	$7.53e-08$	$1.64e-08$
ABC	0.004	$2.80e-09$	$8.44e-10$	$2.54e-10$
PSO	0.004	$1.01e-10$	$2.20e-11$	$4.81e-12$
[23]	0.004	$9.43e-12$	$1.21e-07$	$2.63e-08$

Table 6: Values of shape parameter values (ε) with time intervals at $\Delta t = 0.00001$ and $N = 21$.

T	Parameter (ε) values	
	ABC	PSO
0.001	0.253516	0.253874
0.002	0.267221	0.251490
0.003	0.263530	0.253324
0.004	0.272190	0.253231

Table 7: Comparison of absolute errors of problem 2 with $N = 11$, $\Delta t = 0.0001$, and $a = 1$ at different T and iteration = 20.

X	ABC			[23]		
	T 0.001	0.002	0.003	0.001	0.002	0.003
0.1	$6.1583e-11$	$1.7297e-09$	$2.3772e-09$	$2.684e-10$	$4.563e-09$	$1.798e-08$
0.2	$1.4237e-11$	$1.0123e-09$	$1.4069e-10$	$2.930e-11$	$3.683e-11$	$3.384e-10$
0.3	$7.1848e-12$	$4.8205e-10$	$5.2623e-11$	$3.477e-11$	$2.240e-11$	$1.830e-11$
0.4	$6.4626e-12$	$2.3021e-10$	$1.9678e-11$	$6.720e-12$	$2.199e-12$	$2.511e-12$
0.5	$7.7925e-12$	$1.1580e-10$	$2.5276e-12$	$4.654e-12$	$3.591e-12$	$2.555e-12$
0.6	$1.1981e-11$	$7.5733e-11$	$2.4497e-12$	$1.134e-11$	$1.051e-11$	$1.001e-11$
0.7	$2.2177e-11$	$8.3356e-11$	$1.3942e-12$	$1.974e-12$	$2.707e-12$	$3.189e-12$
0.8	$5.0164e-11$	$1.4345e-10$	$6.0104e-12$	$2.685e-11$	$1.646e-11$	$1.819e-11$
0.9	$1.8208e-10$	$1.6611e-10$	$2.3064e-10$	$2.758e-11$	$4.511e-10$	$1.749e-09$

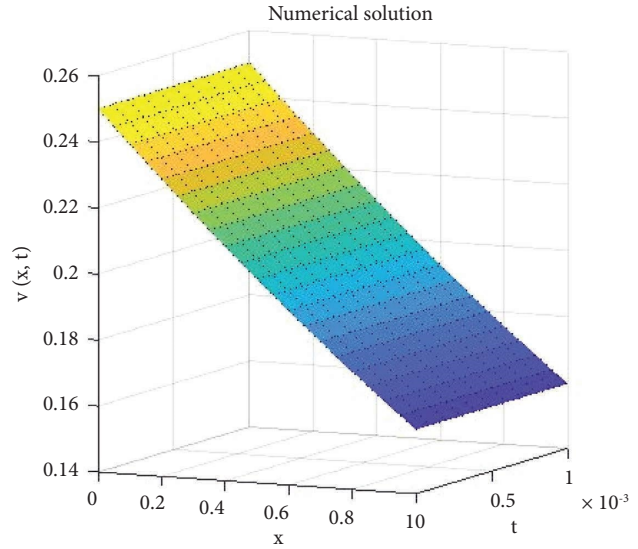

Figure 3: Numerical simulation of problem 1 with $N = 21$, $\Delta t = 0.0001$, and $\Delta t \leq 0.001$.

0.003, and 0.004 and iteration = 71 is similar to results in the literature [23]. The analysis of the shape parameter values is carried out in Table 6 which shows the PSO algorithm as the best optimizer for error giving the value of shape parameter as 0.251490. Table 7 presents the comparative analysis of absolute errors of the problem by ABC with [23] at different time levels which seems better than the available results. Here, the optimised shape parameter value is 0.246477. The numerical solution is presented in Figure 3 for $N = 21$, $\Delta t = 0.0001$ at various time intervals.

Using the current hybrid approach, two Fisher's equations are solved numerically and the results are derived in terms of various error norms, including absolute errors and shape parameter values. The comparison of the obtained results is performed and presented to test the efficacy and application of this novel approach.

5. Conclusion

In this paper, a novel hybrid technique is proposed for computing the numerical solution of Fisher's equation using PSO and ABC optimization algorithms with RBF-PS. Using PSO and ABC optimization algorithms, the concept of ideal shape parameters is proposed because there is a discrepancy between numerical stability and accuracy when using different radial basis functions. To show the accuracy and efficiency of the specific method, two problems are solved numerically. Based on their error norms and shape parameter values, the obtained results are compared with the results available in the literature. The obtained results are more accurate in comparison to the results available in the literature. Form the results, it can be concluded that PSO gives more accurate results compared to the ABC algorithm in terms of less errors. Furthermore, the present work can be explored with various other optimization algorithms, such as the genetic algorithm, the ant colony optimization algorithm, the bacteria foraging optimization algorithm, and the firefly algorithm. Thus, the work has scope to solve the partial differential equation existing in various other fields with minimum errors.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors would like to thank their affiliations for facilitating the publication of this paper through their support.

References

[1] B. Yagmahan and M. M. Yenisey, "Ant colony optimization for multi objective flow shop scheduling problem," *Computers & Industrial Engineering*, vol. 54, no. 3, pp. 411–420, 2008.

[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Micro Machine and Human Science,MHS'95, Proceedings of the 6th International Symposium on IEEE*, pp. 39–43, New York, NY, USA, January 1995.

[3] K. V. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, Springer Verlag, Berlin, Berlin, Germany, 2005.

[4] J. Vesterstrom and R. Thomsen, "A comparative study of differential evolution, particle swarm optimization and evolutionary algorithms on numerical benchmark problems," *Evolutionary Computation Congress*, vol. 2, pp. 1980–1987, 2004.

[5] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*, Morgan Kaufmann Publishers, Los Altos, CA, USA, 2001.

[6] D. B. Fogel, *Introduction to Evolutionary Computation 1:Basic Algorithms and Operators*, Taylor & Francis USA, Philadelphia, PA, USA, 2000.

[7] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.

[8] K. M. Passino, "Biomimicry of bacterial foraging for distributed optimization and control," *Control Systems Magazine IEEE*, vol. 22, pp. 52–67, 2002.

[9] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*, Oxford University Press, Oxford, UK, 1999.

[10] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Technical Report TR06, Erciyes University Press, Kayseri, Turkey, 2005.

[11] D. Özdemïr and S. A. F. A. Dörterler, "An adaptive search equation-based artificial bee colony algorithm for transportation energy demand forecasting," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 4, pp. 1251–1268, 2022.

[12] Z. Bilici, D. Özdemir, and H. Temurtaş, "Comparative analysis of metaheuristic algorithms for natural gas demand forecasting based on meteorological indicators," *Journal of Engineering Research*, vol. 18, Article ID 100127, 2023.

[13] D. Özdemir, S. Dörterler, and D. Aydın, "A new modified artificial bee colony algorithm for energy demand forecasting problem," *Neural Computing & Applications*, vol. 34, no. 20, pp. 17455–17471, 2022.

[14] R. A. Fisher, "The wave of advance of advantageous genes," *Annals of Eugenics*, vol. 7, no. 4, pp. 355–369, 1937.

[15] J. Gazdag and J. Canosa, "Numerical solution of Fisher's equation," *Journal of Applied Probability*, vol. 11, no. 3, pp. 445–457, 1974.

[16] S. Tang and R. O. Weber, "Numerical study of Fisher's equation by a Petrov-Galerkin finite element method," *The Journal of the Australian Mathematical Society, Series B. Applied Mathematics*, vol. 33, no. 1, pp. 27–38, 1991.

[17] A. M. Wazwaz, "The tanh method for traveling wave solutions of nonlinear equations," *Applied Mathematics and Computation*, vol. 154, no. 3, pp. 713–723, 2004.

[18] Y. Tan, H. Xu, and S. J. Liao, "Explicit series solution of travelling waves with a front of Fisher equation," *Chaos, Solitons & Fractals*, vol. 31, no. 2, pp. 462–472, 2007.

[19] M. S. Sahimi and D. J. Evans, "The alternating group explicit iterative method to solve parabolic and hyperbolic partial differential equations," *Annual Review of Heat Transfer*, vol. 2, pp. 283–389, 1989.

[20] T. Hagstrom and H. B. Keller, "The numerical calculation of traveling wave solutions of nonlinear parabolic equations,"

SIAM Journal on Scientific and Statistical Computing, vol. 7, no. 3, pp. 978–988, 1986.

[21] N. Parekh and S. Puri, "A new numerical scheme for the Fisher equation," *Journal of Physics A: Mathematical and General*, vol. 23, no. 21, pp. 1085–1091, 1990.

[22] R. C. Mittal and G. Arora, "Efficient numerical solution of Fisher's equation by using B-spline method," *International Journal of Computer Mathematics*, vol. 87, no. 13, pp. 3039–3051, 2010.

[23] G. Bhatia and G. Arora, "A meshfree numerical technique based on radial basis function pseudospectral method for Fisher's equation," *International Journal of Nonlinear Sciences and Numerical Simulation*, vol. 91, 2019.

[24] A. Verma, R. Jiwari, and M. E. Koksal, "Analytic and numerical solutions of nonlinear diffusion equations via symmetry reductions," *Advances in Difference Equations*, vol. 2014, no. 1, p. 229, 2014.

[25] M. Tamsir, N. Dhiman, and V. K. Srivastava, "Cubic trigonometric B-spline differential quadrature method for numerical treatment of Fisher's reaction-diffusion equations," *Alexandria Engineering Journal*, vol. 57, no. 3, pp. 2019–2026, 2018.

[26] R. C. Mittal and R. Jiwari, "Numerical study of Fisher's equation by using differential quadrature method," *International Journal of Information Technologies and Systems*, vol. 5, pp. 143–160, 2009.