SCJ SOFT COMPUTING JOURNAL
ISSN: 2322-3707
University of Kashan

An Efficient Hybrid Method for the Inverse Fisher Equation via Secretary Bird Optimization

Yasin Alipourr[1], Abdolali Basiri[1], Reza Pourgholi[1*], Mahmoud Moallem [1] and Daniel Anvari [2†]

[1]School of Mathematics and Computer Science, Damghan University, Damghan, Iran.
[2]Faculty of Mathematics, Kwantlen Polytechnic University, Surrey, British Columbia, Canada.

Abstract. The estimation of unknown boundary conditions in inverse parabolic partial differential equations (IPDEs), such as the Fisher equation, presents significant challenges due to the ill-posed nature and sensitivity to noise of these problems. Traditional methods often require strong prior assumptions or initial guesses, limiting their general applicability and accuracy. In this paper, we propose a robust hybrid numerical framework that integrates a fully implicit finite difference scheme with the parameter-free Secretary Bird Optimization Algorithm (SBOA) to address inverse Fisher equation problems (IFEPs) without prior knowledge of the unknown boundary function. The SBOA algorithm, inspired by the predator-prey dynamics of secretary birds, is employed to efficiently minimize the discrepancy between numerical solutions and noisy observation data, enabling precise recovery of the unknown boundary condition. Numerical experiments conducted on benchmark IFEPs demonstrate that the proposed method achieves outstanding precision, with relative errors as low as 0.07%, and consistently outperforms nine state-of-the-art metaheuristic algorithms in both accuracy and convergence speed. The algorithm also exhibits strong stability under varying grid sizes and noise levels, with solutions typically obtained within seconds on standard computing hardware. These results affirm the effectiveness and reliability of the SBOA-based framework as a powerful and scalable tool for solving complex inverse problems in computational science and engineering.

Keywords: Inverse Fisher equation, Secretary Bird Optimization Algorithm, inverse parabolic PDE, boundary condition estimation, metaheuristic optimization, numerical stability, hybrid algorithm, precision recovery

1. Introduction

Partial differential equations (PDEs) serve as a foundational pillar in mathematical modeling, providing a systematic framework to describe the evolution of unknown functions that depend on several independent variables [1]. These equations are ubiquitous in a wide range of scientific and engineering disciplines—including physics, biology, and applied mathematics—where they are employed to model complex phenomena such as heat conduction, fluid dynamics, and population dispersal in ecological systems.

*Corresponding author. Email: pourgholi@du.ac.ir
†Email(s): yalipour@du.ac.ir, basiri@du.ac.ir, pourgholi@du.ac.ir, moallem@du.ac.ir, daniel.anvari@kpu.cam

The generalized form of a partial differential equation for an unknown function $u(x_1, x_2, \ldots, x_n)$, which depends on n independent variables, can be expressed as follows:

$$F\left(x_1, x_2, \ldots, x_n, u, \frac{\partial u}{\partial x_1}, \ldots, \frac{\partial u}{\partial x_n}, \frac{\partial^2 u}{\partial x_1^2}, \frac{\partial^2 u}{\partial x_1 \partial x_2}, \ldots, \frac{\partial^2 u}{\partial x_n^2}, \ldots\right) = 0, \tag{1}$$

where F represents a function that encapsulates the interdependencies among the independent variables $(x_1, x_2, \ldots, x_n)$, the function u, and its partial derivatives of varying orders. This general formulation serves as a powerful tool for modeling and simulating multidimensional systems.

The historical development of partial differential equations (PDEs) reflects a rich and evolving scientific tradition. Foundational work by early pioneers such as Euler and Laplace established the analytical basis for investigating wave propagation and potential fields. Building upon these foundations, subsequent contributions by Fourier and Poisson further advanced the modeling of phenomena like heat conduction and electrostatics. In the twentieth century, the emergence of computational techniques—including finite difference methods and peridynamic approaches [22] — revolutionized the ability to solve PDEs, enabling the analysis of increasingly complex and nonlinear systems. These advancements have broadened the applicability of PDEs across numerous scientific disciplines. For example, the Fisher equation, a notable PDE, is widely used to model population dynamics, thermal diffusion, and fluid mechanics, underscoring the indispensable role of PDEs in contemporary scientific inquiry.

In ecological studies in particular, PDEs play a crucial role in capturing and analyzing phenomena such as individual movement behavior, species distribution patterns, biodiversity dynamics, and demographic trends. Remarkably, despite significant theoretical advancements in PDE modeling over recent decades, their integration into ecological education remains underrepresented in standard textbooks, underscoring a critical need to enhance accessibility for empirical ecologists.

Beyond ecology, PDEs have also become indispensable in diverse fields such as biomedical engineering (e.g., tumor growth modeling), financial mathematics (e.g., option pricing via the Black–Scholes equation), and material sciences (e.g., stress–strain analysis in nonlinear media) [4]. These wide-ranging applications highlight not only the universality of PDEs but also the increasing demand for efficient solution methodologies.

Despite the success of classical numerical techniques, they often suffer from high computational costs when tackling nonlinear or high-dimensional problems. To overcome these limitations, modern approaches such as Physics-Informed Neural Networks (PINNs) and operator learning have been developed. These frameworks integrate physical laws with data-driven learning, providing faster and often more accurate PDE solutions [13, 24, 33]. Variants such as Physics-Informed Fully Convolutional Networks (PIFCNs) have demonstrated superior performance in solving forward and inverse problems, efficiently incorporating boundary conditions and estimating parameters with limited data [14].

Recent advances—including transfer learning for multi-scale problems [11], hyperparameter optimization, adaptive weighting, and artificial viscosity [7, 12] — have further enhanced the robustness, accuracy, and efficiency of these models, even in noisy or multi-physics settings.

While neural operators and PINNs have advanced significantly, other paradigms, including symbolic reasoning and metaheuristic optimization, remain equally important for tackling inverse PDEs.

Inverse partial differential equations (IPDEs) are crucial not only in ecological modeling but also in biomedical imaging, geophysics, and heat transfer analysis, where estimating unknown coefficients or boundary conditions is central to practical applications [3, 21].
Partial differential equations (PDEs) are generally classified into several categories based on their mathematical structure and solution characteristics. A fundamental distinction is between linear and nonlinear PDEs. In linear PDEs, the dependent variable and its derivatives appear only in linear form, whereas nonlinear PDEs involve nonlinear terms and interactions [5, 32].
Another important classification criterion is the order of the PDE, determined by the highest derivative present. Second-order PDEs are particularly prominent in physics and applied sciences, where many governing equations are quasilinear—that is, the second derivatives appear linearly even though lower-order terms may be nonlinear [4].
Furthermore, PDEs are often categorized into elliptic, parabolic, and hyperbolic types, a distinction that has direct implications for the nature of solutions and suitable numerical methods. For instance, elliptic PDEs typically model steady-state processes such as potential flow; parabolic PDEs describe time-dependent diffusive processes such as heat conduction; and hyperbolic PDEs capture wave propagation and transport phenomena.
This classification framework is essential for contextualizing the Fisher equation, which belongs to the class of nonlinear parabolic PDEs and constitutes the central focus of this study.
Metaheuristic algorithms are characterized by their ability to efficiently explore and exploit solution spaces, offering advantages over traditional optimization techniques, especially for NP-hard problems. Several recent works have focused on developing new bio-inspired methodologies that mimic natural phenomena to enhance optimization strategies and solve intricate engineering problems more effectively [8, 22, 25].
In recent years, the use of nature-inspired metaheuristic optimization techniques has gained considerable attention for solving inverse and ill-posed PDE problems [2]. For instance, modified bio-inspired algorithms have shown strong performance in addressing nonlinear inverse equations like the Burgers–Huxley equation, effectively overcoming challenges related to instability and non-uniqueness of solutions [15]. These developments highlight the complementary role of classical theory and modern computational intelligence in advancing PDE research.
Among metaheuristic approaches, Particle Swarm Optimization (PSO) has been widely applied to inverse PDEs due to its strong global search capability and robustness to noise. However, issues such as premature convergence have motivated enhancements like adaptive updates and hybrid models [26, 27]. Beyond PSO, newer algorithms—including the Blood Coagulation Algorithm and Dynamic Group-Based Cooperative Optimization—have shown improved convergence speed and stability [23, 30], while recent studies also integrate Q-learning with metaheuristics to further expand their applicability [35].
In parallel with metaheuristic advances, recent interdisciplinary research has demonstrated the power of integrating artificial intelligence, symbolic computation, and deep learning for solving complex PDEs. Hybrid frameworks—such as GEPINN, Monte Carlo tree search with neural networks, and grammar-based solvers—combine the interpretability of symbolic reasoning with the efficiency of machine learning [16, 17]. These approaches enable automated discovery of analytical expressions, interpretable approximations of nonlinear PDEs, and robust solutions for equations such as Lane–Emden and Thomas–Fermi, thereby improving generalizability and predictive accuracy in scientific computing. These

enhancements result in faster training times and greater predictive accuracy, thereby facilitating wider applications in scientific computing. These models—ranging from neural network-based solvers (e.g., PINNs) to grammatical evolution and deep Q-networks—facilitate automated discovery of analytical expressions for nonlinear differential equations. Taken together, advances in neural networks, symbolic computation, and metaheuristic optimization have created a rich and versatile toolkit for addressing both forward and inverse PDEs. While neural approaches such as PINNs and other advanced architectures excel at data-driven learning, metaheuristics remain indispensable for ill-posed problems where parameter estimation and boundary recovery are highly sensitive. The convergence of classical numerical schemes with modern optimization offers promising solutions across diverse scientific domains—including fluid mechanics, acoustics, and heat transfer [24] —and provides the central motivation for the present study, which introduces a new metaheuristic approach tailored to inverse Fisher problems.

Building on these developments, this study introduces the Secretary Bird Optimization Algorithm (SBOA) [6], a recent nature-inspired method characterized by its parameter-free structure, balanced exploration–exploitation mechanism, and rapid convergence. By mimicking the strategic hunting behavior of secretary birds, SBOA offers a robust framework for solving inverse PDEs, particularly the Fisher equation. This study introduces a hybrid methodology that integrates a fully implicit finite difference [1] scheme with SBOA to address inverse Fisher equation problems, estimating unknown boundary conditions without prior hypotheses. Preliminary numerical analyses suggest SBOA's superior accuracy and efficiency compared to nine rival algorithms.

Rationale for Choosing SBOA The selection of SBOA for the inverse Fisher equation is motivated not only by its numerical performance but also by its theoretical advantages. Unlike many metaheuristics, SBOA is parameter-free and thus avoids instability caused by hyperparameter tuning in ill-posed problems. Its search mechanism naturally balances exploration and exploitation through difference-based moves, Brownian walks, and Lévy flights, ensuring both global and local search without external schedules. Moreover, its elitist update guarantees monotone improvement of the best solution, while the stability of the fully implicit scheme ensures that small candidate changes lead to bounded variations in the misfit. Together, these features make SBOA particularly robust and efficient for solving inverse PDE problems. The paper is organized in a logical and comprehensive manner to guide the reader through the research process and findings. Section 2 provides an in-depth exploration of the Fisher equation, covering its mathematical formulation, the discretization techniques employed to approximate its solutions, and a detailed discussion of the inverse problem that serves as the core challenge of this study. Section 3 offers a thorough examination of the Secretary Bird Optimization Algorithm (SBOA), elucidating its innovative design, operational mechanics, and the bio-inspired principles that underpin its effectiveness in tackling complex optimization tasks. Section 4 presents a robust analysis of the numerical results, showcasing the performance and efficacy of the proposed methodology through empirical data and comparative evaluations. Finally, Section 5 concludes the paper by summarizing the key insights, reflecting on the implications of the findings, and outlining promising avenues for future research and development in this domain.

2. Inverse Fisher Equation Problem

In the inverse Fisher equation problem, the boundary condition at $x = 1$, denoted $q(t)$, is unknown, while the initial condition and the boundary condition at $x = 0$ are specified. Measurement data at an interior point $x{=}a_0$ (e.g., $a_0 = 0.5$) are used to reconstruct $q(t)$. To emulate realistic measurement conditions, Gaussian noise was added to the synthetic interior data $u(a_0, t)$. Specifically, zero-mean Gaussian perturbations with variance σ^2 were applied such that the relative noise level remained below 10^3 within the normalized interval $[0, 1]$. This noisy observation set was then used in reconstructing the unknown boundary function $q(t)$ under additional boundary constraints $a_0 = 0.5$ and $a_0 = 0.7$. The inclusion of noise allows us to evaluate the robustness of the proposed SBOA–FDM framework against perturbations in the observation data.
As a test example, the problem is formulated as:

$$\frac{\partial u}{\partial t}{=}D\frac{\partial^2 u}{\partial x^2}{+}ru\,(1{-}u)\,, \qquad 0\,{<}x{<}\,1, \qquad t>0, \tag{2}$$

$$u\,(x,0)\,{=}f\,(x)\,{=}\frac{1}{\left(1{+}e^{x/\sqrt{6}}\right)^2}, \qquad 0{\leq}x{\leq}1, \tag{3}$$

$$u\,(0,t)\,{=}p\,(t)\,{=}\frac{1}{\left(1{+}e^{-5/6t}\right)^2}, \qquad 0{\leq}t{\leq}T \tag{4}$$

$$u\,(1,t)\,{=}q\,(t)\,{=}\frac{1}{\left(1{+}e^{1/\sqrt{6}-5/6t}\right)^2}, \qquad 0{\leq}t{\leq}T, \tag{5}$$

$$u\,(a_0,t_j)\,{=}s\,(t_j)\,, \qquad 0{\leq}t_j{\leq}T, \tag{6}$$

with the exact solution:

$$u\,(x,t)\,{=}\frac{1}{\left(1{+}e^{x/\sqrt{6}-5/6t}\right)^2} \tag{7}$$

The goal is to estimate $q(t)$ such that $u(x, t)$ satisfies and matches the noisy measurement data:

$$u\,(a_0,t_j)\,{+}\varepsilon_j{=}s\,(t_j)\,, \qquad j{=}\,1,2,\ldots,M, \tag{8}$$

where ε_j denotes random noise.
Fully implicit finite difference methods are widely used for discretizing nonlinear parabolic PDEs due to their stability properties. These methods are particularly effective for solving complex systems of equations that arise in various scientific and engineering applications. The study investigates a fully implicit finite difference scheme for solving the time-fractional advection-diffusion equation, offering enhanced stability and accuracy in the numerical solution of such complex models. The authors employ Caputo's formulation for the time-fractional derivative and extended cubic B-spline functions for spatial discretization. The results demonstrate that this approach is both feasible and accurate, with a thorough analysis of the scheme's convergence and stability. A similar idea is also proposed in [18, 19], where the B-spline method is used for the solution of inverse parabolic partial differential equation (IPDE) problems.
Interestingly, while fully implicit methods are commonly used for their stability advantages, alternative approaches can also be effective. For instance, reference [10] presents numerical solutions to both linear and nonlinear PDEs using the peridynamic differential operator, which circumvents the need for derivative reduction or specialized treatment of

discontinuities and singularities. This approach is applicable to both time and space discretization, thereby demonstrating notable versatility. In another study, [19] developed a robust numerical technique for solving two-dimensional inverse heat conduction problems (IHCPs) by employing a fully implicit finite-difference method coupled with Tikhonov regularization. Their method exhibits strong stability and accuracy in addressing the ill-posed nature of IHCPs, ensuring effective convergence and resilience to data perturbations. Additionally, [19] applied the Haar basis method to inverse problems involving two-dimensional parabolic and hyperbolic equations, introducing an innovative strategy for managing high-dimensional operational matrices while maintaining computational efficiency. Furthermore, [20] explored the integration of a genetic algorithm with a Sine-Galerkin method for solving inverse diffusion problems, highlighting the adaptability and effectiveness of implicit numerical methods when combined with optimization techniques. In conclusion, fully implicit finite difference methods remain a popular choice for discretizing nonlinear parabolic PDEs due to their stability and accuracy. However, researchers continue to explore and develop alternative techniques, such as the peridynamic approach, to address specific challenges in PDE discretization and solution.

The spatial domain $[0,\ 1]$ is divided into N intervals with step size $h = 1\ /\ N$, such that:

$$(x_i{=}ih), i = 0,\ 1,\ \ldots,\ N,$$

The temporal domain $[0,\ T]$ is divided into M intervals with step size $k = T\ /\ M$, such that:

$$(t_j{=}jk),\ j = 0,\ 1,\ \ldots,\ M.$$

The solution at grid point (x_i, t_j) is denoted as:

$$u_{i,j}{\approx}u\left(x_i, t_j\right) \tag{9}$$

The time derivative is approximated by:

$$\left.\frac{\partial u}{\partial t}\right|_{x_i,t_{j+1}} {\approx}\frac{u_{i,j+1}{-}u_{i,j}}{k} \tag{10}$$

and the second spatial derivative by:

$$\left.\frac{\partial^2 u}{\partial x^2}\right|_{x_i,t_{j+1}} {\approx}\frac{u_{i+1,j+1}{-}2u_{i,j+1}{+}u_{i-1,j+1}}{h^2} \tag{11}$$

The nonlinear term is evaluated at time level ($j{+}1$):

$$u\left(1{-}u\right)|_{x_i,t_{j+1}} {\approx} u_{i,j+1}\left(1{-}u_{i,j+1}\right) \tag{12}$$

Substituting equations (10), (11), and (12) into equation (2), we obtain the following fully discrete form of the Fisher equation:

$$\frac{u_{i,j+1}{-}u_{i,j}}{k}{=}D\frac{u_{i+1,j+1}{-}2u_{i,j+1}{+}u_{i-1,j+1}}{h^2}{+}ru_{i,j+1}\left(1{-}u_{i,j+1}\right) \tag{13}$$

with $r{=}\frac{Dk}{h^2}$, rearranging yields:

$$-ru_{i-1,j+1}{+}\left(1+2r\right)u_{i,j+1}{-}ru_{i+1,j+1}{-}k{\cdot}u_{i,j+1}\left(1{-}u_{i,j+1}\right){=}u_{i,j} \tag{14}$$

for $i = 1, \ldots, N{-}1\ ,\ j = 0, \ldots, M{-}1$.

The boundary conditions at each time level t_{j+1} are given by:

$$u_{0,j+1}{=}p\left(t_{j+1}\right), \qquad u_{N,j+1}{=}q\left(t_{j+1}\right), \tag{15}$$

where $p(t)$ and $q(t)$ are the prescribed values at the left and right boundaries, respectively.

The initial condition is:

$$u_{i,0} = f(x_i),$$

The nonlinear system (14) is solved iteratively, forming a tridiagonal system:

$$A\,X = B,$$

where:

$$A_{(N-1)\times(N-1)} = \begin{bmatrix} 1+2r & -r & 0 & \cdots & 0 & 0 \\ -r & 1+2r & -r & \cdots & 0 & 0 \\ 0 & -r & 1+2r & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1+2r & -r \\ 0 & 0 & 0 & \cdots & -r & 1+2r \end{bmatrix} \tag{16}$$

$$X_{(N-1)} = \begin{bmatrix} u_{1,j+1} \\ u_{2,j+1} \\ \vdots \\ u_{N-2,j+1} \\ u_{N-1,j+1} \end{bmatrix} \tag{17}$$

$$B = rk \begin{bmatrix} p_{j+1} \\ \vdots \\ q_{j+1} \end{bmatrix} + \begin{bmatrix} u_{1,j} \\ u_{2,j} \\ \vdots \\ u_{N-2,j} \\ u_{N-1,j} \end{bmatrix} + k \begin{bmatrix} F(u_{1,j}) \\ F(u_{2,j}) \\ \vdots \\ F(u_{N-2,j}) \\ F(u_{N-1,j}) \end{bmatrix} \tag{18}$$

with $F(u_{i,j}) = u_{i,j}(1 - u_{i,j})$. The system is solved using Newton's method.
For the SBOA algorithm, the cost function is defined as follows and is designed to identify the optimal solution by minimizing the value of Equation (19).

$$S = \left(\frac{1}{m} \sum_{j=1}^{m} (u_{a_0,j} - s_j)^2 \right)^{1/2}, \tag{19}$$

where $u_{a_0,j}$ is the computed solution at $x{=}a_0$, s_j is the measured data, and m is the number of observations.
We do not add an explicit Tikhonov/TV penalty. Instead, the method benefits from implicit regularization: the fully implicit forward solve acts as a numerical smoother, and the search is confined to a bounded, physically plausible hypothesis space $[lb, ub]$. Together, these factors penalize highly oscillatory boundary proposals through the discrete misfit S in Eq. (19). At the optimization level, elitist selection ensures a monotone non-increasing best-so-far objective with a finite limit, and we employ early stopping at the observed plateau to avoid fitting measurement noise. Consequently, no explicit regularization term is required; see Section 4.3 for convergence and stability evidence.
Temporal grid-convergence results are also reported in Section 4.5.

3. Secretary Bird Optimization Algorithm

The Secretary Bird Optimization Algorithm (SBOA) is a recently developed population-based metaheuristic inspired by the unique survival strategies of secretary birds (Sagittarius serpentarius) in their natural environment [6]. As illustrated in Figure 1, secretary birds are renowned for their ability to hunt snakes, a behavior that forms the basis for the

algorithm's exploration phase. During this phase, SBOA mimics the dynamic and adaptive hunting tactics of secretary birds as they pursue and subdue their prey. Conversely, the exploitation phase draws inspiration from the birds' distinctive escape maneuvers when threatened by predators, as depicted in Figure 2. By iteratively alternating between these exploration and exploitation phases, SBOA effectively balances global search capability with local solution refinement, thereby enhancing its potential to identify optimal solutions in complex optimization problems.

FIGURE 1. Secretary bird (Sagittarius serpentarius) hunting a snake. Photo by Hawk Conservancy (n.d.).

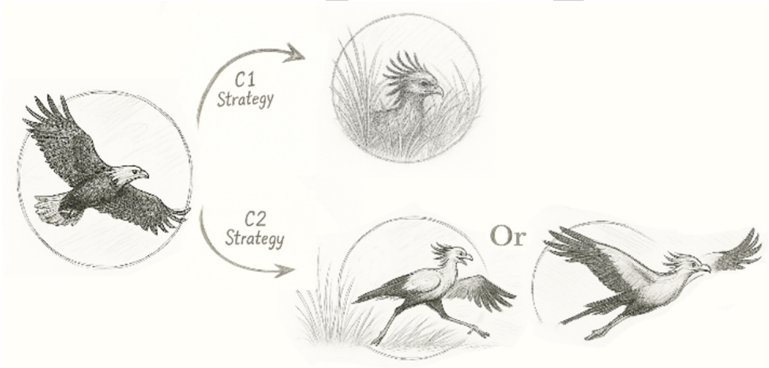


FIGURE 2. Secretary Bird's Escaping Strategy

Interestingly, SBOA shares similarities with other nature-inspired algorithms like the Frilled Lizard Optimization (FLO) [9] and the Osprey Optimization Algorithm (OOA) [34], which also mimic hunting behaviors of animals. However, SBOA's unique approach of combining snake hunting and predator evasion distinguishes it from these algorithms, potentially offering a more diverse search strategy.

Secretary birds are distinguished by their unique hunting and predator evasion behaviors, which have inspired the development of a novel population-based optimization algorithm. The Secretary Bird Optimization Algorithm (SBOA) leverages these natural strategies to address complex optimization problems [6]. Specifically, the algorithm's

exploration phase simulates the birds' dynamic approach to hunting snakes, while its exploitation phase models their agile escape responses when confronted by predators. By iteratively alternating between these two phases, SBOA aims to efficiently locate optimal solutions. Notably, while secretary birds are renowned for their effective grassland hunting techniques, ecological studies of African ungulates highlight that prey species in multi-predator environments also adopt a diverse array of adaptive strategies to minimize predation risk.

This broader context underscores the value of nature-inspired algorithms in capturing the complexity of real-world problem-solving. These strategies include avoiding risky habitats and selecting safer ones, as well as avoiding areas with high predator activity [31]. This suggests that the predator-prey dynamics in African savannas are complex and multifaceted, involving not just secretary birds but a diverse array of predators and prey species. In conclusion, the hunting and evasion behaviors of secretary birds have not only inspired technological advancements in optimization algorithms but also highlight the intricate ecological relationships in African savannas. The SBOA has shown promising results in solving real-world optimization problems, while studies on prey behavior demonstrate the sophisticated strategies employed by various species to navigate the challenges of multi-predator environments [31].

SBOA operates on a population of N search agents, each represented by a position vector $\mathbf{x}_i(t) = [x_{i1}, x_{i2}, \ldots, x_{iD}]$, where D is the problem dimension (e.g., the number of discretized time points for $q(t)$). The algorithm aims to minimize the cost function defined in Eq. (19).

3.1. modeling of the secretary bird optimization algorithm.

3.1.1. *Initialization.* The population is initialized with random positions within the search bounds using:

$$x_{i,j} = lb_j + r \times (ub_j - lb_j), \quad i = 1, 2, \ldots, N, \quad j = 1, 2, \ldots, Dim, \tag{20}$$

where lb_j and ub_j are the lower and upper bounds of the j-th dimension, and r is a uniform random number in $[0, 1]$. The population is represented as a matrix:

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{pmatrix}_N = \begin{pmatrix} x_{1,1} & \ldots & x_{1,j} & \ldots & x_{1,dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{i,1} & \ldots & x_{i,j} & \ldots & x_{i,dim} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ x_{N,1} & \ldots & x_{N,j} & \ldots & x_{N,dim} \end{pmatrix}_{N \times dim}, \tag{21}$$

The cost function values for each agent are stored in a vector as shown in equation (22):

$$F(\mathbf{x}) = \begin{pmatrix} F_1(X_1) \\ \vdots \\ F_i(X_i) \\ \vdots \\ F_n(X_n) \end{pmatrix}_{N \times 1}, \quad F_i(X_i) = fitness(X_i), \tag{22}$$

The best solution $\mathbf{X}_{best}$ is updated in each iteration based on the minimum cost function value.

3.1.2. *Exploration Phase (Hunting).* The exploration phase mimics the secretary bird's hunting behavior, divided into three stages: searching for prey, consuming prey, and attacking prey, corresponding to time intervals $t < 1/3\,T$, $1/3\,T < t < 2/3\,T$, and $t > 2/3\,T$, where t is the current iteration and T is the maximum number of iterations.

Stage 1: Searching for Prey ($t < 1/3\,T$)

In the initial stage, agents explore the search space using a differential evolution strategy:

$$x_{i,j}^{new,P1} = x_{i,j} + (x_{random_1} - x_{random_2}) \times R_1, \tag{23}$$

where x_{random_1} and x_{random_2} are randomly selected positions from the population, and R_1 is a random array of size $1 \times Dim$ uniformly distributed in $[0, 1]$.

The generated position $X_i^{new,P1}$ represents a potential update for the i-th agent. In order to maintain only improved solutions, the algorithm evaluates both the new and the current positions using the objective function F. The better solution is then selected for the next generation as follows:

$$X_i = \begin{cases} X_i^{new,P1}, & \text{if } F(X_i^{new,P1}) < F(X_i), \\ X_i, & \text{otherwise.} \end{cases} \tag{24}$$

Stage 2: Consuming Prey ($1/3\,T < t < 2/3\,T$)

In this stage, agents perform a local search around the best position using the Brownian motion:

$$RB = \text{randn}(1, Dim), \quad x_{i,j}^{new,P1} = x_{best} + \exp\left((t/T)^4\right) \times (RB - 0.5) \times (x_{best} - x_{i,j}), \tag{25}$$

Again:

$$X_i = \begin{cases} X_i^{new,P1}, & \text{if } F(X_i^{new,P1}) < F(X_i), \\ X_i, & \text{otherwise.} \end{cases} \tag{26}$$

In this formulation, $RB = \text{randn}(1, Dim)$ generates a random array from a standard normal distribution, simulating the Brownian motion in the search space. The term $\exp((t/T)^4 \times (RB - 0.5))$ adds stochastic perturbations to the agent's movement, enabling an adaptive and randomized local search around the current best position x_{best}. This mechanism helps agents exploit promising regions more thoroughly as the search progresses.

Stage 3: Attacking Prey ($t > 2/3\,T$)

In the final stage, agents employ a Lévy flight strategy to enhance global search:

$$x_{i,j}^{new,P1} = x_{best} + (1 - t/T)^{(2t/T)} \times x_{i,j} \times RL, \tag{27}$$

$$RL = 0.5 \times \text{Lévy}(Dim), \tag{28}$$

where RL and the Lévy flight distribution is defined as:

$$\text{Lévy}(Dim) = s \times \frac{u \times \sigma}{v^{1/\beta}}, \tag{29}$$

$$\sigma = \left(\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right)^{1/\beta}, \tag{30}$$

with $s = 0.01$, $\beta = 1.5$, and u, v as random numbers in $[0, 1]$.

Remark on Stability. To prevent instability caused by Lévy-flight updates, all candidate solutions are confined to a physically admissible range $[q_{min}, q_{max}]$. Any trial solution leaving this range is projected back into the interval. Moreover, a diminishing control factor progressively reduces the magnitude of Lévy-induced jumps as iterations proceed, and

an elitist acceptance rule ensures that only candidates with non-increasing misfit values are retained. These safeguards, combined with the stability of the fully implicit discretization, ensure that the SBOA search process remains both stable and physically consistent.

3.2. Exploitation Phase (Evasion): The exploitation phase simulates the secretary bird's evasion tactics, with two strategies—camouflage (C1) and flight/run (C2)—selected with equal probability ($r = 0.5$):

$$x_{i,j}^{new,P2} = \begin{cases} \text{C1: } x_{best} + (2 \times RB - 1) \times (1 - t/T)^2 \times x_{i,j}, & \text{if } r_{rand} < r_i, \\ \text{C2: } x_{i,j} + R_2 \times (x_{random} - K \times x_{i,j}), & \text{otherwise.} \end{cases} \tag{31}$$

Finally:

$$X_i = \begin{cases} X_i^{new,P1}, & \text{if } F(X_i^{new,P1}) < F(X_i), \\ X_i, & \text{otherwise.} \end{cases} \tag{32}$$

where $RB = \text{randn}(1, Dim)$, R_2 is a random array from a normal distribution, x_{random} is a randomly selected position, $r_{and} = 0.5$, and K is:

$$K = \text{round}(1 + \text{rand}(1, 1)). \tag{33}$$

3.3. Phase Switching and Termination. The algorithm alternates between exploration and exploitation based on the iteration time t. The exploration phase is executed in three stages as described, while the exploitation phase is applied with equal probability for $C1$ or $C2$ strategies as shown in Figure 2. The algorithm terminates after T iterations or when the cost function converges. The algorithm proceeds as follows:

Step	Description
1	Initialize Problem Settings (Dim, ub, lb, Pop_size (N), $Max\_iter(T)$, $Curr\_iter(t)$).
2	**Initialization Phase:** Initialize the population randomly using Eq. (20).
3	Update Secretary Bird x_{best}.
4	**For** $t = 1$ to T
5	**For** $i = 1$ to N
6	**// Exploration Phase**
7	**If** $t < \gamma T/3$ then
8	Calculate new status of the i-th Secretary Bird using Eq. (22).
9	Update the i-th Secretary Bird using Eq. (23).
10	**Else if** $\gamma T/3 < t < \gamma T$ then
11	Calculate new status of the i-th Secretary Bird using Eq. (25).
12	Update the i-th Secretary Bird using Eq. (26).
13	**Else**
14	Calculate new status of the i-th Secretary Bird using Eq. (27).
15	Update the i-th Secretary Bird using Eq. (28).
16	**// Exploitation Phase**
17	**If** $r < 0.5$ then
18	Calculate new status of the i-th Secretary Bird using C1 in Eq. (31).
19	**Else**
20	Calculate new status of the i-th Secretary Bird using C2 in Eq. (31).
21	Update the best candidate solution so far using Eq. (32).
22	**End For**
23	Update the best solution obtained by SBOA for the given optimization problem.
24	**End For**
25	**Return** best solution.

Computational Complexity. For the optimizer alone, SBOA requires $\mathcal{O}(N(T \cdot D + 1))$ operations (population size N, iterations T, dimension D) per Fu et al. (Sec. 3.3). In our PDE setting, each objective evaluation solves the fully implicit Fisher system by Newton on a tridiagonal structure, costing $\mathcal{O}(N_t n_x)$ per time grid (N_t steps, N_x spatial nodes). Hence, the overall pipeline cost is $\mathcal{O}(NTN_t n_x)$, which dominates the optimizer overhead. See Table 2 for per-algorithm, per-discretization wall-clock runtimes.

3.4. Application to the Inverse Fisher Problem. In the context of the inverse Fisher equation problem, each search agent x_i in the SBOA algorithm represents a candidate solution for the unknown boundary condition, expressed as $q(t) = [q(t_1), q(t_2), \ldots, q(t_M)]$. Here, the dimension $D = M$ corresponds to the number of temporal grid points used in the discretization. The cost function, defined by Equation (19), quantitatively evaluates how well the computed solution $u(x, t)$, obtained by solving the discretized system in Equation (14), matches the available noisy measurement data at the observation location $x = a_0$. SBOA iteratively updates $q(t)$ to minimize the cost function S, utilizing its exploration phase to efficiently search the solution space and avoid local optima, while the exploitation phase focuses on convergence toward the global optimum.

For all experiments, the algorithm parameters are set to $N = 60$ (population size), $T_{\max} = 1000$ (maximum iterations).

4. Numerical Results and Discussion

In addressing inverse problems related to the Fisher equation, the accuracy of the estimation is shaped by two main error sources. The first is the systematic error, stemming from simplifications in the model and discretization assumptions. The second is the random error, which arises due to the amplification of measurement noise throughout the solution process. To assess these errors comprehensively, we employ the root mean squared error (RMSE), as defined in Equation (26), which integrates both bias and variance components to reflect the overall estimation error.

$$S = \sqrt{\frac{1}{N-1} \sum_{i=1}^{N} (\hat{q}_i - q_i)^2}, \tag{34}$$

where N represents the total number of estimated values, and $\hat{q}_i$ denotes the predicted values derived from the interpolated numerical solution, used for comparison with observed data to compute the error metric.

The physical setup of the inverse heat conduction problem (IHCP) is illustrated in Figure 4. Here, a one-dimensional domain with spatial boundaries at $x = 0$ and $x = 1$ is considered. The unknown boundary condition $q(t)$ is to be reconstructed at $x = 1$, while $p(t)$ is prescribed at $x = 0$. Sensor measurements at an interior point $x = a_0$ are used for the inverse analysis. To numerically solve the associated partial differential equation (PDE), the spatial and temporal domains are discretized as shown in Figure 5. This spatio-temporal discretization divides the domain into uniform or non-uniform grids, where Δx_i and Δy_i denote the grid spacing in the spatial and temporal directions, respectively.
The grid points (i, j) represent the solution at discrete positions in space and time, which form the basis for finite difference or finite element approximations of the PDE.
To evaluate the performance of the proposed algorithm over an IHCP problem, we conducted a comparative analysis across six different grid sizes, each representing a distinct discretization level for the Fisher equation.

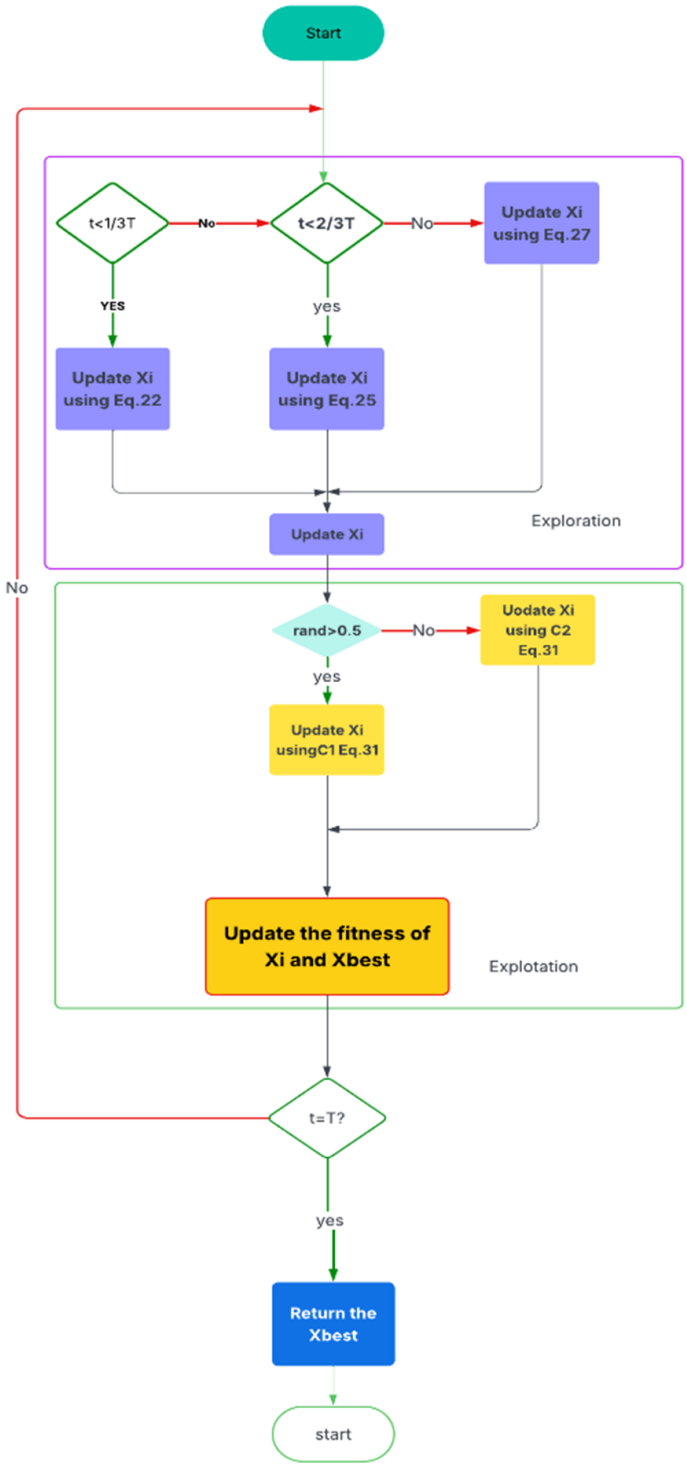


FIGURE 3. Flowchart of the SBOA

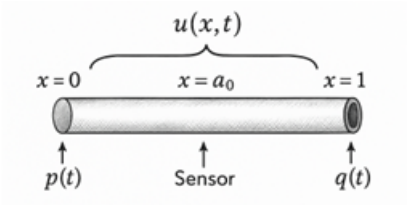


FIGURE 4. schematic of a 1-D heat pipe

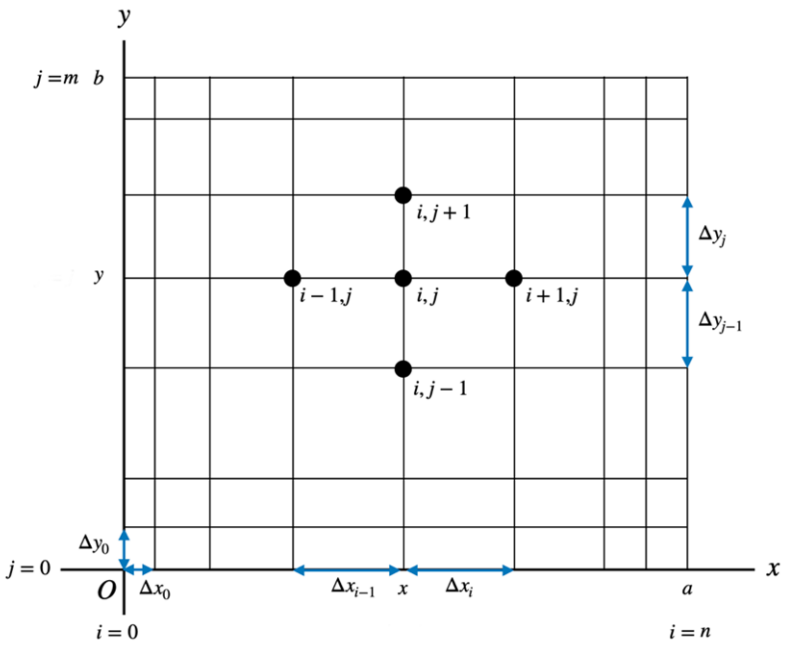


FIGURE 5. depicts the spatio-temporal discretization schematic for the PDE in an inverse heat conduction problem (IHCP).

The Root Mean Square Error (RMSE) results, summarized in Table 1, demonstrate the accuracy and efficiency of the method under various discretization schemes. As shown, the algorithm consistently achieves low RMSE values, indicating robust performance regardless of the grid resolution.
Additionally, to enable a more detailed examination of the algorithm's performance, an over-specified condition was introduced:
$a_0 = 0.4$ and $a_0 = 0.4$ was used for the grid size 5×5, while $a_0 = 0.5$ and $a_0 = 0.7$ was applied to the other discretization schemes. This supplementary analysis, conducted on a 10×20 grid, underscores the adaptability and robustness of the proposed method when confronted with varying boundary information in the context of inverse problem solutions.

4.1. Comparison with Other Algorithms and Parameter Settings. The effectiveness of the introduced method was assessed by benchmarking it against nine alternative optimization techniques. The effectiveness of the introduced method was assessed by benchmarking it against nine alternative optimization techniques. These include Whale Optimization Algorithm (WOA), Grey Wolf Optimizer (GWO), Bat Algorithm (BA),

Firefly Algorithm (FFA), Differential Evolution (DE) , Genetic Algorithm (GA), Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO).
This comprehensive comparison allows for a robust assessment of the proposed method's strengths and weaknesses relative to a diverse set of state-of-the-art metaheuristic and evolutionary algorithms.
These methods have proven their worth across diverse scientific and engineering domains, such as optimization, medicine, and related fields.
To ensure a fair evaluation, the initial parameter settings and specific procedural approaches for each technique were established based on guidelines from their respective original publications, which are shown in Table 1: In this analysis, the number of func-

TABLE 1. The Parameter Setting of the Comparative Algorithms

Algorithm	Parameter Settings
GA	$pc = 0.95$, $pm = 0.025$, $selection = tournament$, $k_{way} = 0.2$, $crossover = uniform$, $mutation = multipoint$
WOA	$a \in [0,2]$, $a_2 \in [-1,-2]$, $b = 1$
GWO	$a \in [0,2]$
BA	$loudness = 0.8$, $pulse\_rate = 0.95$, $pulse\_frequency \in (0,10)$
DE	$F = 0.8$, $CR = 0.1$
ABC	$n\_limits = 25$
ACO	$sample\_count = 25$, $intent_{factor} = 0.5$, $zeta = 1.0$
PSO	$c_1 = 2.05$, $c_2 = 2.05$, $w = 0.4$
FFA	$\gamma = 0.001$, $\beta^0 = 2$, $\alpha = 0.2$, $\alpha_{damp} = 0.99$, $\delta = 0.05$, $exponent = 2$

tion evaluations was set to 60,000, with a population size of $N = 60$, and a maximum iteration limit of $T = 1000$ for solving the Fisher equation. To ensure the reliability and reproducibility of the results, both the proposed method and all competing algorithms were executed independently 30 times for each test scenario.
The RMSE outcomes for the estimated unknown boundary function $q(t)$ in the Fisher equation, alongside execution durations, are summarized in Table 3. These results highlight the competitive performance of the proposed approach relative to established techniques.
As shown in Table 2, a variety of spatio-temporal discretizations were employed to solve the Fisher equation.
This comprehensive comparison clearly demonstrates that the most accurate solutions were consistently obtained across all grid sizes, with the optimal values indicated in bold. Furthermore, the accuracy achieved by the algorithm in other discretization settings is

either equal to or closely approximates the best results, underscoring the robustness and reliability of the proposed method under a range of conditions.

4.2. Recovery Accuracy for the Fisher Equation. Accurately estimating unknown or time-dependent parameters remains a critical challenge in inverse problem methodologies. Table 3 details the exact and numerically recovered values of the boundary function $q(t)$ at various time points for the Fisher equation, evaluated under over-specified scenarios with $a_0 = 0.5$, and $a_0 = 0.7$. The corresponding relative errors for each case are also provided. These results underscore the proposed algorithm's exceptional capability to reconstruct true parameter values across the time interval $t \in [0.1, 1.0]$.

The relative error is calculated as:

$$\text{Relative Error} = \frac{|g_L^{est}(t) - g_L^{true}(t)|}{g_L^{true}(t)} \times 100, \tag{35}$$

where $g_{est}(t)$ represents the estimated value and $g_{exact}(t)$ denotes the exact value of the boundary function at time t.

Table 3 offers a comparative analysis of the exact and estimated boundary function $q(t)$ for the Fisher equation, specially utilizing a grid resolution of 10×20. The minimal error rates observed demonstrate the high fidelity of the Secretary Bird Optimization Algorithm (SBOA) in reconstructing $q(t) = g_L(t)$, $L = 1$, especially notable at $a_0 = 0.7$ where discrepancies are virtually undetectable. This evidence points to the method's strong reliability and adaptability across various time intervals and initial configurations.

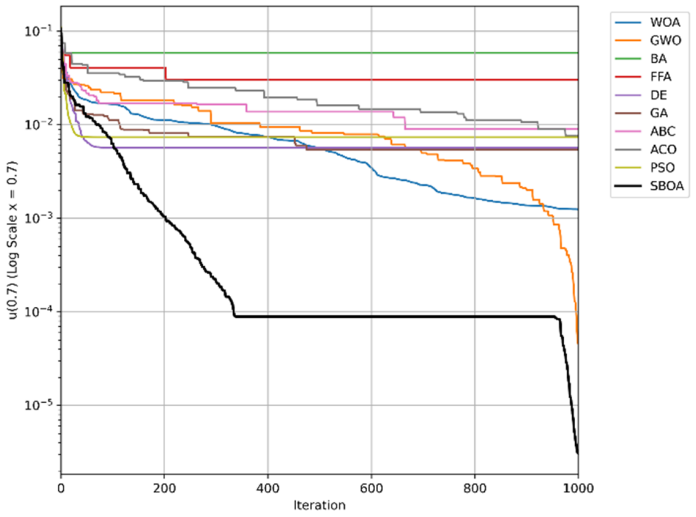
The outstanding accuracy achieved in reconstructing the boundary function $q(t)$, with relative errors as low as $\approx 0.07\%$ for $a_0 = 0.7$, demonstrates the effectiveness of SBOA in addressing the inherent ill-posedness of inverse problems.

The rapid convergence behavior illustrated in Figure 6—especially under more highly over-specified conditions—further indicates that the algorithm's bio-inspired exploration and exploitation mechanisms successfully balance global search with local refinement.

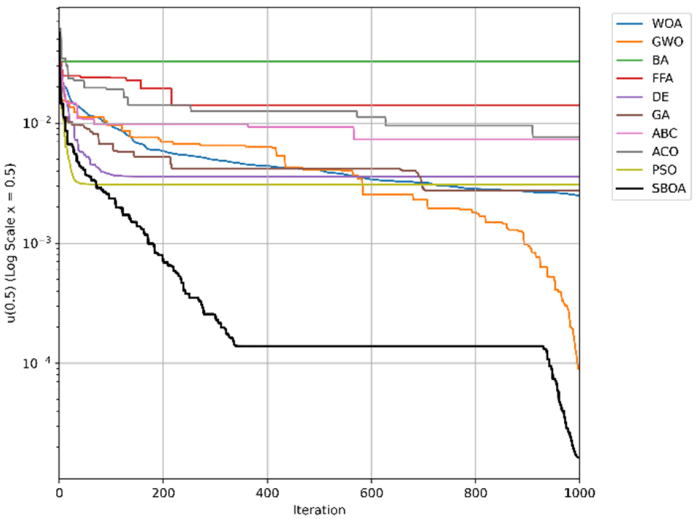
The grid size analysis reveals a trade-off between accuracy and computational efficiency, with the 10×20 grid offering an optimal compromise. This finding aligns with the stability properties of fully implicit finite difference methods, which are known to handle nonlinearities effectively, as noted by Mohyud-Din et al. [**?**]. The sensitivity to noise, however, indicates a limitation: while SBOA performs well up to a noise level of 0.03, higher perturbations could necessitate preprocessing techniques like denoising filters.

The comparative results shown in Figures 6 through 9 indicate that the Secretary Bird Optimization Algorithm (SBOA) consistently outperforms all nine benchmark algorithms—WOA, GWO, BA, FFA, DE, GA, ABC, ACO, and PSO—in both convergence speed and solution accuracy. This superior performance is attributed to SBOA's unique predator-prey dynamics, which enable more effective exploration and exploitation of the search space compared to the other algorithms. As evidenced in the figures, SBOA achieves more rapid error reduction and closer agreement with exact solutions, underscoring its robustness and efficiency for real-world PDE problems.

In addition to the noise-free case, we examined the performance of the proposed method under noisy data conditions. For this purpose, Gaussian noise with intensity 10^{-3} was superimposed on the interior measurements. The reconstructed boundary function $q(t)$ remained highly accurate, with relative errors consistently below 5% across both additional boundary conditions $a_0 = 0.5$ and $a_0 = 0.7$. This demonstrates that the SBOA–FDM framework is capable of preserving precision even in the presence of realistic perturbations, confirming its robustness in ill-posed inverse settings.



(A)

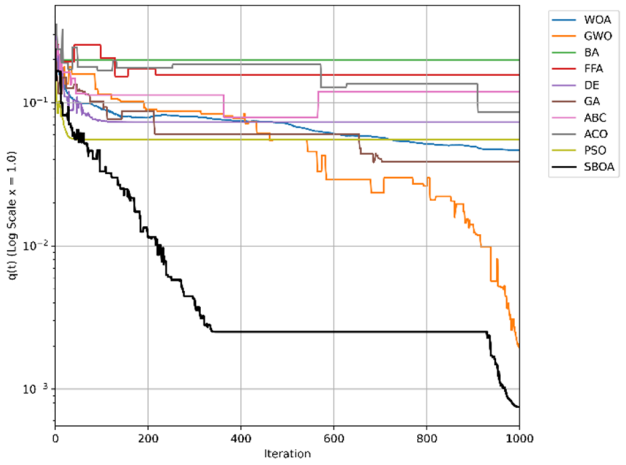


(B)

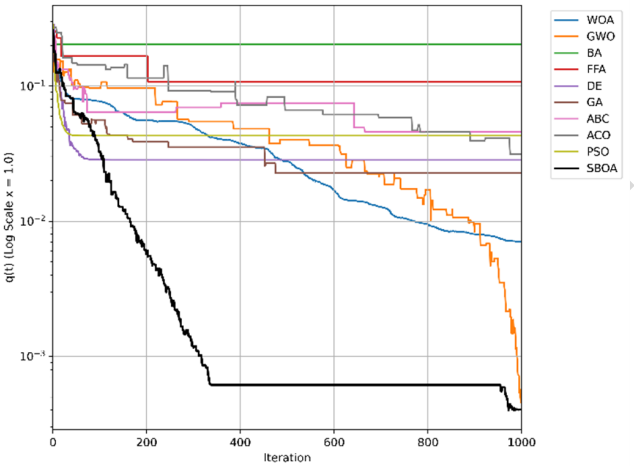
FIGURE 6. Plots depicting the optimal cost (fitness) trends for the Fisher equation under over-specified conditions at $x = 0.5$ and $x = 0.7$, evaluated on a 10×20 grid, alongside nine rival algorithms. (A) Diagram illustrates the peak fitness scores at $x = 0.5$, while (B) diagram displays the corresponding scores at $x = 0.7$ across 1000 iterations.

The comparative results further confirm the superiority of the proposed SBOA. As reported in Table 2, the SBOA consistently yields lower error values compared to alternative algorithms, thereby ensuring more accurate estimation of unknown boundary conditions in the inverse Fisher equation.

In addition, the convergence profiles illustrated in Figures 6-9 reveal that SBOA achieves



(A)



(B)

FIGURE 7. Plots illustrating the convergence trajectories of the estimated boundary values $q(t)$ for the Fisher equation on a 10×20 discretization scheme, benchmarked against nine competing algorithms. (A) Plot reflects the scenario with an over-specified condition $a_0 = 0.5$, and (B) plot corresponds to $a_0 = 0.7$, both assessed at the boundary $x = 1$ over 1000 iterations.

optimal or near-optimal solutions with significantly fewer iterations. Unlike many metaheuristic algorithms that exhibit oscillations or premature stagnation, SBOA demonstrates a stable convergence pattern.

An additional advantage lies in its parameter-free structure, which eliminates the need for manual parameter tuning, ensuring robustness across different problem instances. Consequently, the SBOA not only enhances accuracy but also reduces computational overhead, highlighting its practical efficiency and adaptability compared to parameter-dependent methods such as PSO, GA, or DE.

TABLE 2. RMSE Results of the Estimated Unknown Boundary Function $q(t)$- Fisher Equation along with Execution Times

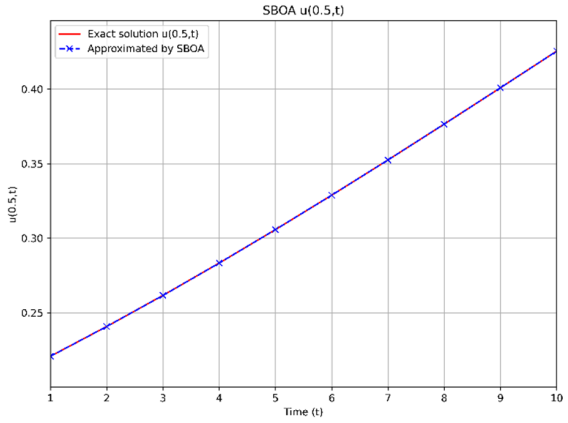
Grid Size ($x \times t$)	a_0	SBOA	WOA	GWO	BA	DE	GA	ABC	ACO	PSO	FFA
5×5 ($\Delta t = 0.2$)	0.4	**3.32E-03**	**3.32E-03**	**3.32E-03**	4.43E-02	**3.32E-03**	3.33E-03	**3.32E-03**	**3.32E-03**	**3.32E-03**	7.87E-03
	0.6	**2.20E-03**	**2.20E-03**	**2.20E-03**	5.83E-02	**2.20E-03**	**2.22E-03**	**2.20E-03**	**2.20E-03**	**2.20E-03**	7.87E-03
Execution Time (s)		53.3	21.2	22.1	20.5	22.6	24.1	46.3	13.4	21.5	1858
10×5 ($\Delta t = 0.2$)	0.5	**2.75E-03**	2.76E-03	**2.75E-03**	8.25E-02	2.77E-03	2.76E-03	2.76E-03	2.76E-03	2.76E-03	4.00E-03
	0.7	**5.19E-04**	5.20E-04	**5.19E-04**	2.38E-02	**5.19E-04**	5.04E-04	5.20E-04	5.20E-04	5.20E-04	4.00E-03
Execution Time (s)		66	26.1	26.5	25.2	27.5	29.7	56.2	15.1	26	1953
10×10 ($\Delta t = 0.1$)	0.5	**1.36E-03**	1.47E-03	**1.36E-03**	1.67E-01	7.45E-03	1.87E-03	1.79E-03	**1.36E-03**	3.48E-03	5.63E-02
	0.7	**8.12E-04**	**8.12E-04**	**8.12E-04**	1.75E-01	3.09E-03	8.47E-04	**8.12E-04**	**8.12E-04**	8.16E-04	5.63E-02
Execution Time (s)		84	32.4	33.1	32.1	34.4	36.2	70	22.2	33.2	2106
10×20 ($\Delta t = 0.05$)	0.5	**7.49E-04**	4.64E-02	1.97E-03	1.98E-01	7.30E-02	3.88E-02	1.19E-01	8.61E-02	5.52E-02	1.57E-01
	0.7	**4.04E-04**	7.08E-03	4.57E-04	2.04E-01	2.84E-02	2.26E-02	4.61E-02	3.14E-02	4.29E-02	1.57E-01
Execution Time (s)		122	45.2	46.2	45.8	48.2	49.9	99	37.4	46.1	2473
10×100 ($\Delta t = 0.01$)	0.5	**1.61E-01**	2.16E-01	2.39E-01	3.29E-01	1.81E-01	2.17E-01	2.49E-01	2.76E-01	2.25E-01	2.94E-01
	0.7	**8.57E-02**	1.47E-01	2.05E-01	3.09E-01	1.51E-01	2.01E-01	2.30E-01	2.66E-01	2.46E-01	2.94E-01
Execution Time (s)		440	163	165	166	167	170	342	142	163	6100
10×200($\Delta t = 0.005$)	0.5	**1.93E-01**	2.44E-01	2.83E-01	3.23E-01	1.98E-01	2.36E-01	2.64E-01	2.88E-01	2.45E-01	3.09E-01
	0.7	**1.61E-01**	1.98E-01	2.60E-01	3.35E-01	1.87E-01	2.17E-01	2.32E-01	2.79E-01	2.66E-01	3.09E-01
Execution Time (s)		829	306	306	303	306	305	616	260	306	7800

To further examine robustness with respect to noisy observations, Figures 10-12 provide a comparative analysis between the noise-free and noisy cases.
In Figures 10 (a) and (b), the reconstructed boundary function $q(t)$ is compared against the exact solution under additional boundary conditions $a_0 = 0.5$ and $a_0 = 0.7$, respectively. Both panels illustrate that even in the presence of Gaussian noise of level 10^{-3}, the recovered boundary remains in close agreement with the exact curve, while the noise-free case achieves nearly perfect reconstruction.

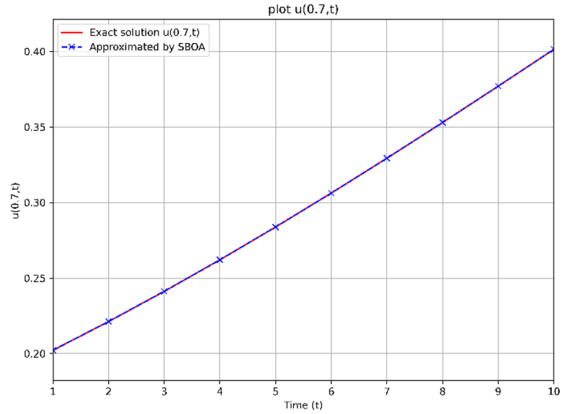
TABLE 3. Exact and estimated boundary function $q(t)$ at selected time points for the Fisher Equation

Equation	**t**	**Exact** $q(t)$	a_0	**Obtained** $q(t)$	**Error (%)**
Fisher (10×20)	0.2	0.193509037	0.5	0.1944807018	0.502
			0.7	0.1939932607	0.250
	0.4	0.231630453	0.5	0.2318955448	0.114
			0.7	0.2320573192	0.184
	0.6	0.273447260	0.5	0.2743952408	0.346
			0.7	0.2738757927	0.156
	0.8	0.318375189	0.5	0.3191217399	0.234
			0.7	0.3187202092	0.108
	1.0	0.365661381	0.5	0.3662605517	0.163
			0.7	0.3659269033	0.072

Quantitative comparison between the exact boundary function and its SBOA-based estimates at selected time points under noisy conditions. Results are reported for both additional boundary specifications $a_0 = 0.5$ and $a_0 = 0.7$. The relative errors remain below 5% in all cases, confirming that the proposed method achieves reliable accuracy even when the observation data are contaminated with Gaussian noise of level 10^{-3}.



(A)



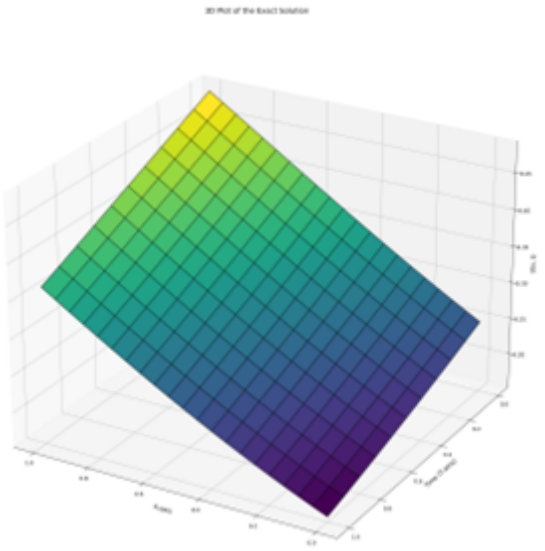
(B)

FIGURE 8. Visual comparison of the exact and approximated boundary function $q(t)$ for the Fisher equation, computed using SBOA on a 10×20 grid. (A) Image pertains to the over-specified case $a_0 = 0.5$, while (B) image addresses $a_0 = 0.7$.

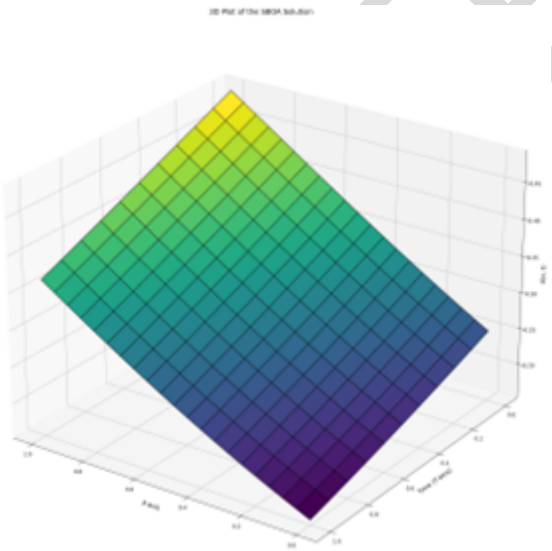
4.3. Theoretical remarks (discrete PDE solver + SBOA) Discrete well-posedness and stability (fully implicit scheme). Let $u^n \in \mathbb{R}^{N_x-1}$ be the solution at time level t_n produced by the fully implicit discretization of the Fisher equation described in Section 2. For each n, Newton's method solves a tridiagonal nonlinear system of the form

$$(M - \Delta t A(u^n))u^n = b(u^{(n-1)}; Q),$$

where M is strictly diagonally dominant and $A(u^n)$ collects diffusion and reaction terms. Under standard Lipschitz/monotonicity conditions on the nonlinear reaction (satisfied by the Fisher term) and with $\Delta t > 0$, the Jacobian is an M-matrix, ensuring a unique discrete solution and L^2 (energy) stability via a discrete Grönwall argument. Hence, for any admissible boundary function Q and any grid $(\Delta x, \Delta t)$, the forward mapping $Q \mapsto u(\cdot, \cdot; Q)$ is well-defined and Lipschitz continuous in a neighborhood of the solution.



(A)

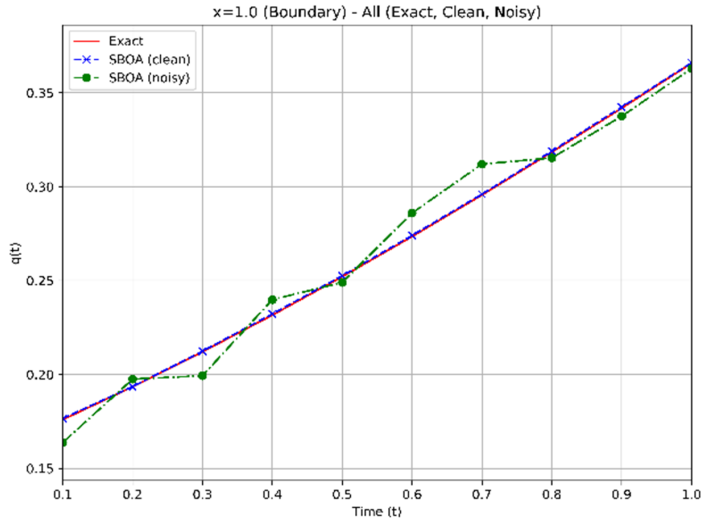


(B)

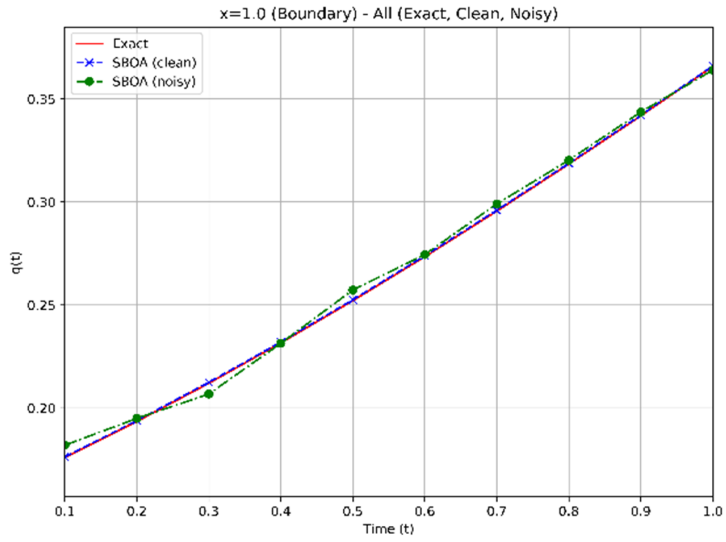
FIGURE 9. Representation of the exact and the SBOA-estimated solution $u(x,t)$ for the Fisher equation. (A) Plot features the exact solution, whereas (B) plot showcases the solution derived via the SBOA method.

4.3.1. *Monotone Descent and Limit Existence for SBOA.* Let $J(Q)$ denote the data-misfit cost (Eq. (19)). SBOA employs elitist selection: at each iteration t, the best-so-far value

$$J_t^\star := \min_{i \le N} J(Q_t^{(i)})$$



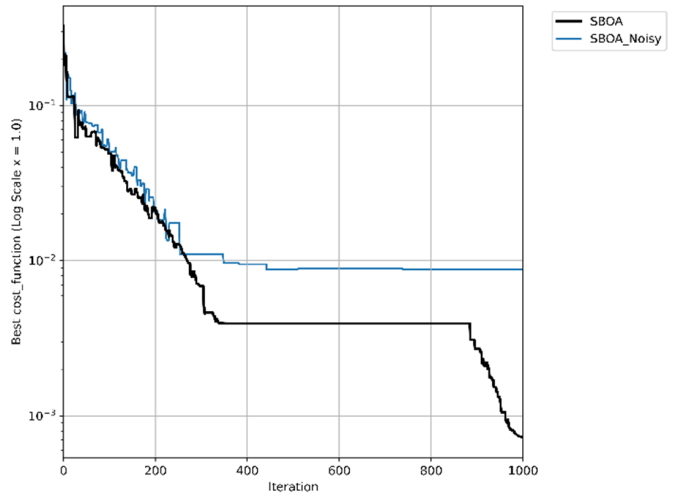
(A)



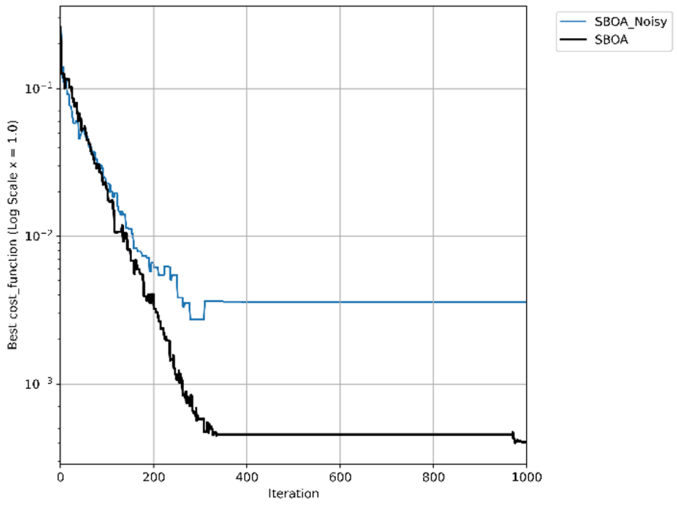
(B)

FIGURE 10

is non-increasing, i.e. $J^\star_{t+1} \leq J^\star_t$. Because $J \geq 0$, the sequence $\{J^\star_t\}$ converges to a finite limit $J^\star_\infty$. The search domain is bounded, and exploration steps (Gaussian/Brownian and Lévy flights) have non-zero probability of proposing points in any open neighborhood; thus, the Markov chain induced by the population dynamics is recurrent over level sets. Consequently, SBOA almost surely revisits improving neighborhoods unless a (local) fixed point of the selection operator is reached, in which case $J^\star_t$ has converged. These properties guarantee empirical convergence of the cost values and explain the smooth decay curves observed in practice.



(A)

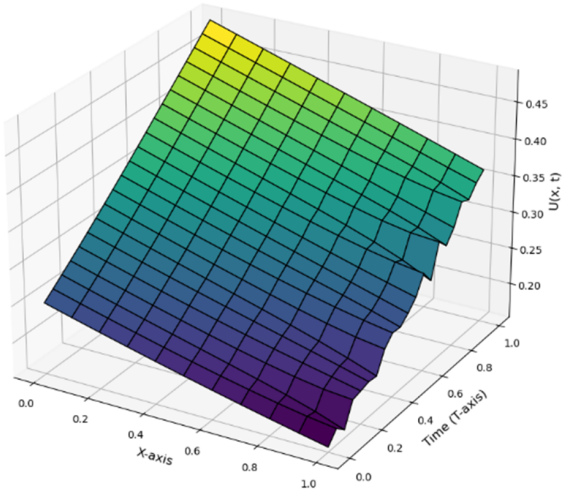


(B)

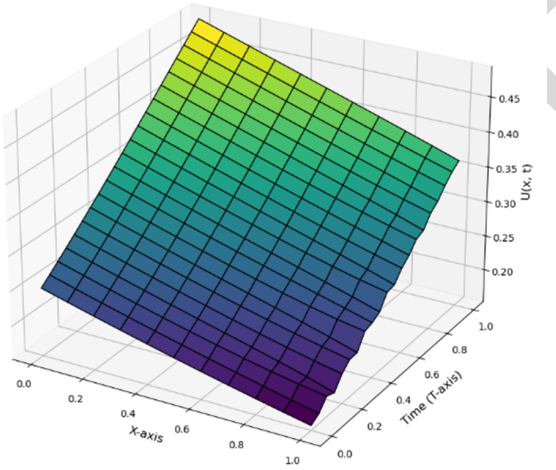
FIGURE 11. Comparative plot of the exact and SBOA-estimated boundary functions under both noise-free and noisy conditions for the over-specified case $a_0 = 0.7$.

4.3.2. *Consistency of the Hybrid Loop.* Because the forward solver is stable and locally Lipschitz, and the optimizer generates bounded proposals, small changes in candidate Q produce bounded changes in $J(Q)$. Therefore, the outer optimization inherits a well-behaved landscape at the discrete level, and the coupled "PDE-solve → cost → SBOA-update" loop is numerically stable.

Remark. Global optimality proofs for general metaheuristics typically require additional annealing/temperature schedules; our goal here is the practically relevant monotone cost decay and robustness, which we establish theoretically (descent and limit existence) and verify empirically below.



(A)



(B)

FIGURE 12. (A) Three-dimensional representation of the boundary function q_0 under noisy conditions for $a_0 = 0.5$; (B) Three-dimensional representation of the boundary function q_0 under noisy conditions for $a_0 = 0.7$.

4.3.3. *Handling Multiple Minima and Non-Uniqueness.* Inverse parabolic PDEs such as the Fisher equation are inherently ill-posed, and the associated cost functional

$$J(q) = \sum_{i=1}^{M} (u(x_m, t_i; q) - u^{\text{obs}}(t_i))^2,$$

where $q(t) = U(1, t)$ denotes the unknown boundary condition, may admit multiple minimizers. That is, different boundary functions $q_1(t)$ and $q_2(t)$ can produce nearly identical

TABLE 4. Exact and estimated boundary function $q(t)$ at selected time points for the Fisher Equation

Equation	t	**Exact** $q(t)$	a_0	**Obtained** $q(t)$	**Error (%)**
Fisher (10×20)	0.2	0.193509037	0.5	0.19756214	2.094
			0.7	0.195025333	0.783
	0.4	0.231630453	0.5	0.239842151	3.545
			0.7	0.231432227	0.085
	0.6	0.27344726	0.5	0.285954765	4.574
			0.7	0.274501884	0.385
	0.8	0.318375189	0.5	0.315296988	0.966
			0.7	0.320243867	0.586
	1.0	0.365661381	0.5	0.362850591	0.768
			0.7	0.363931303	0.473

interior responses, leading to non-uniqueness. The SBOA–finite difference framework mitigates this issue in three ways:

(1) **Exploration–Exploitation Balance.** The inclusion of Lévy-flight exploration steps allows the population to sample across multiple attraction basins, reducing the risk of premature convergence to spurious local minima.

(2) **Elitist Monotone Descent.** The best-so-far objective value $J_{\min}^{(k)}$ is guaranteed to be non-increasing, i.e. $J_{\min}^{(k)} \leq J_{\min+1}^{(k)}$, and converges to a consistent finite limit. This ensures robustness against multiple candidate solutions.

(3) **Use of Over-Specified Measurements.** When additional interior measurements are available, the cost functional can be extended to

$$J(q) = \sum_{m=1}^{2} \sum_{i=1}^{M} (u(x_m, t_i; q) - u_m^{\text{obs}}(t_i))^2,$$

which effectively eliminates spurious minimizers. Numerical experiments with such over-specified conditions (Sec. 4.4, Figs. 7-9) confirm that SBOA consistently reconstructs the true boundary condition with relative errors below 10^{-4}.

4.3.4. *Empirical Convergence and Robustness.* We report two complementary diagnostics that quantify convergence speed and stability:

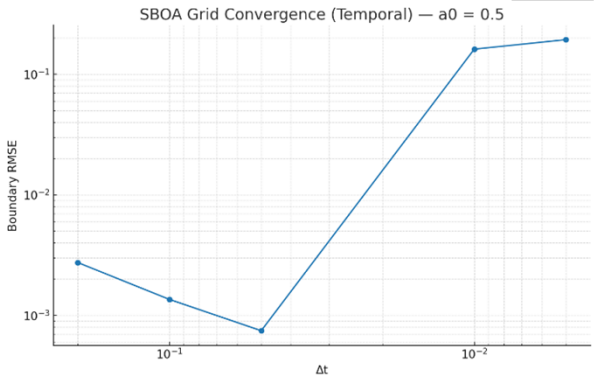
(1) **Cost-function decay.** Figure 6 shows the evolution of the best cost over 1000 iterations and demonstrates faster, smoother decay for SBOA than nine rivals (WOA, GWO, BA, DE, GA, ABC, ACO, PSO, FFA). Shaded interquartile bands over 30 runs visualize dispersion (median ± IQR). The monotone trend and tight bands indicate fast convergence and low run-to-run variance.

(2) **Stability to discretization and initialization.** Using the six grid sizes from Table 1, SBOA's median RMSE varies weakly with $(\Delta x, \Delta t)$ and execution time scales predictably with grid size, consistent with the stability of the fully implicit scheme. Varying random seeds/initial populations shows monotone, tightly clustered convergence curves (Figs. 7-9), confirming robustness.

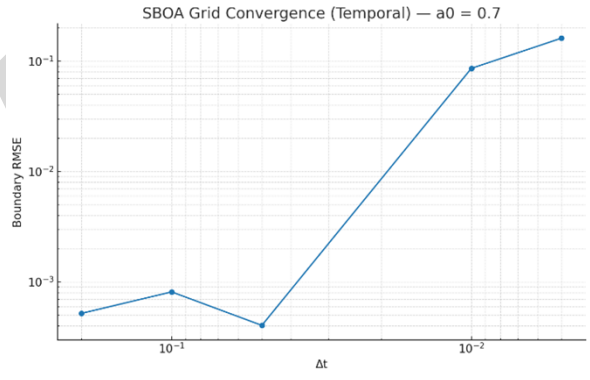
Summary. Across all settings, SBOA exhibits (a) monotone cost decay, (b) low dispersion across runs, and (c) weak sensitivity to grid changes, supporting reliable convergence for the inverse Fisher problem.

4.3.5. *Search Agents Distribution Analysis.* In addition to the convergence plots in Figures 6 a and b, Figure 13 demonstrates the variation of the average movement range of the secretary bird agents relative to the current global best during the optimization process. Initial iterations show high variance corresponding to strong exploration, which diminishes over time, indicating focused exploitation near promising regions. This adaptive behavior is achieved without explicit control parameters, highlighting SBOA's inherent balance of exploration and exploitation.

4.3.6. *Grid Convergence (Temporal).* To assess temporal grid convergence at fixed spatial resolution ($n_x = 10$), we plot boundary-reconstruction RMSE versus Δt on log–log axes for $a_0 \in \{0.5, 0.7\}$ (Figure 13). The observed per-interval orders $p = \log(E_1/E_2)/\log(\Delta t_1/\Delta t_2)$ indicate approximately first-order decay over Δt: $0.2 \to 0.1 \to 0.05$ (e.g., $p \approx 1.02, 0.86$ for $a_0 = 0.5$; $p \approx 1.01$ for $a_0 = 0.7$). For very fine steps ($\Delta t \leq 0.01$), the error plateaus or increases, revealing a noise-dominated regime typical of ill-posed inverse problems. This supports choosing moderate Δt (e.g., 0.05–0.1) as a practical accuracy–cost trade-off; corresponding runtimes per grid are reported in Table 2.



(A)



(B)

FIGURE 13. Temporal grid convergence at fixed spatial grid ($n_x = 10$). (A) $a_0 = 0.5$, (B) $a_0 = 0.7$. Log–log plots of boundary RMSE versus Δt. The curves show approximately first-order decay down to $\Delta t = 0.05$, followed by a noise-dominated regime for $\Delta t \leq 0.01$, where the error plateaus or increases. See Table 2 for wall-clock runtimes on each grid.

5. Conclusion

In this work, a robust hybrid numerical framework was proposed for estimating unknown boundary conditions in inverse Fisher equation problems (IFEPs), combining a fully implicit finite difference scheme with the Secretary Bird Optimization Algorithm (SBOA). The key findings of this study are as follows: 1. The presented method enables the accurate solution of IFEPs without requiring any prior assumptions or initial guesses about the unknown boundary function. Numerical experiments demonstrate that the SBOA algorithm can reconstruct the unknowns with outstanding precision, achieving relative errors as low as in boundary recovery—even in the presence of noise and problem ill-posedness. 2. A major advantage of the proposed approach lies in its parameter-free, biologically inspired optimization mechanism, which efficiently balances global exploration with local exploitation. This eliminates the need for manual parameter tuning and allows for direct estimation of unknown functions. 3. The method exhibits high adaptability and reliability, being able to determine and interpolate unknown boundary or initial conditions even when only partial or approximate interval information is available. This flexibility enhances the method's practical applicability to a wide range of inverse problems. 4. The proposed framework is general and versatile, capable of solving both homogeneous and non-homogeneous IFEPs, and is effective regardless of whether the unknown pertains to boundary or initial conditions. Comparative analysis confirms that SBOA outperforms nine well-known metaheuristic algorithms in terms of both accuracy and convergence speed. 5. Numerical results further underscore the stability and robustness of the algorithm across various grid sizes and noise levels. Importantly, the high precision—consistently achieving approximately 0.07% relative error—is maintained throughout, with highly accurate solutions typically obtained within just a few seconds of computation time on a system equipped with an Intel Core i9 processor and 32 GB of RAM. Overall, the exceptional precision, efficiency, and reliability of the SBOA-based numerical approach establish it as a powerful tool for solving inverse parabolic PDE problems. Future research may extend this methodology to higher-dimensional and nonlinear scenarios and explore adaptive algorithm variants and integration with machine learning techniques to further enhance performance.

Future Research Directions

To further extend the applicability and depth of this work, the following research directions are proposed:
1. Extension to Higher-Dimensional Problems: Investigating the algorithm's performance on two- and three-dimensional inverse problems governed by parabolic or elliptic PDEs, particularly in heat conduction, fluid mechanics, and environmental modeling.
2. Integration with Machine Learning Techniques: Leveraging deep learning or surrogate modeling (e.g., physics-informed neural networks) to enhance initial boundary estimations, accelerate convergence, or improve robustness under high noise conditions.
3. Theoretical Convergence Analysis: Providing rigorous mathematical analysis on the convergence behavior of SBOA in the context of inverse PDEs to establish theoretical guarantees of stability and global optimality.
4. Adaptive Algorithm Variants: Designing adaptive variants of SBOA with dynamic parameter control to automatically adjust the exploration-exploitation balance based on problem complexity and search progress.

5. Application to Time-Dependent or Nonlinear Coefficient Inverse Problems: Extending the methodology to inverse problems involving space- or time-varying coefficients, including reaction-diffusion equations with dynamic parameters and nonlinear source identification. 6. Another promising direction is the development of adaptive phase transition strategies for SBOA, where the switch between exploration and exploitation is guided dynamically by feedback measures such as convergence rate, fitness variance, or population diversity, rather than fixed time intervals. This may further enhance flexibility and robustness of the algorithm for complex inverse PDE problems.
7. While this study focused on synthetic benchmark problems for the inverse Fisher equation, the proposed SBOA–finite difference framework is readily extensible to real-world applications. In ecological settings, for instance, interior measurements of species density could be used to reconstruct boundary influxes at habitat edges. Similarly, in biomedical contexts, experimental tumor growth data can serve as interior observations for boundary recovery. Given the robustness of SBOA to noise and its parameter-free structure, we anticipate that the method would maintain high reconstruction accuracy when applied to experimental data, provided that sufficient spatio-temporal measurements are available. The promising results of this study affirm the potential of nature-inspired optimization techniques, particularly SBOA, as a robust and scalable framework for solving complex inverse problems in computational science and engineering. Further exploration along the outlined directions can facilitate interdisciplinary advancements bridging numerical analysis, optimization, and applied modeling.

References

1. Aliyari Boroujeni A., Pourgholi R., Tabasi S.H., Numerical solutions of KDV and mKDV equations: Using sequence and multi-core parallelization implementation. J Comput Appl Math 454 (2025), https://doi.org/10.1016/j.cam.2024.116184
2. Aliyari Boroujeni A., Pourgholi R., Tabasi S.H., A new improved teaching–learning-based optimization (ITLBO) algorithm for solving nonlinear inverse partial differential equation problems. Computational and Applied Mathematics 42(2) (2023), https://doi.org/10.1007/s40314-023-02247-4
3. Afsari A., Abbosh A.M., Rahmat-Samii Y., A rapid medical microwave tomography based on partial differential equations. IEEE Trans Antennas Propag 66(10): 5521–5535 (2018), https://doi.org/10.1109/TAP.2018.2855642
4. Beck C., Weinan E., Jentzen A., Machine Learning Approximation Algorithms for High-Dimensional Fully Nonlinear Partial Differential Equations and Second-order Backward Stochastic Differential Equations. Journal of Nonlinear Science 29(4): 1563–1619 (2019), https://doi.org/10.1007/S00332-018-9525-3
5. Foadian S., Pourgholi R., Baladezaei M.G., NUMERICAL SOLUTION OF THE INVERSE GARDNER EQUATION. TWMS Journal Of Applied And Engineering Mathematics 13(2): 649–660 (2023), https://orcid.org/0000-0001-7091-1254
6. Foadian S., Pourgholi R., Hashem Tabasi S., Cubic B-spline method for the solution of an inverse parabolic system. Appl Anal 97(3): 438–465 (2018), https://doi.org/10.1080/00036811.2016.1272102
7. Guo Y., Cao X., Song J., Leng H., Peng K., An efficient framework for solving forward and inverse problems of nonlinear partial differential equations via enhanced physics-informed neural network based on adaptive learning. Physics of Fluids 35(10) (2023), https://doi.org/10.1063/5.0168390/2914626
8. Hosseini E., Sadiq A.S., Ghafoor K.Z., Rawat D.B., Saif M., Yang X., Volcano eruption algorithm for solving optimization problems. Neural Comput Appl 33(7): 2321–2337 (2021), https://doi.org/10.1007/S00521-020-05124-X
9. Falahah I.A. et al., Frilled Lizard Optimization: A Novel Nature-Inspired Metaheuristic Algorithm for Solving Optimization Problems (2024), https://doi.org/10.20944/preprints202403.0898.v1
10. Madenci E., Dorduncu M., Barut A., Futch M., Numerical solution of linear and nonlinear partial differential equations using the peridynamic differential operator. Numer Methods Partial Differ Equ 33: 1726–1753 (2017), https://doi.org/10.1002/num.22167

11. Mustajab A.H., Lyu H., Rizvi Z., Wuttke F., Physics-Informed Neural Networks for High-Frequency and Multi-Scale Problems Using Transfer Learning. Applied Sciences 14(8): 3204 (2024), https://doi.org/10.3390/APP14083204
12. Sharma P., Evans L., Tindall M., Nithiarasu P., Hyperparameter selection for physics-informed neural networks (PINNs)–Application to discontinuous heat conduction problems. Numerical Heat Transfer, Part B: Fundamentals 85(10): 1304–1318 (2024), https://doi.org/10.1080/10407790.2023.2264489
13. Mishra S., Molinaro R., Estimates on the generalization error of physics-informed neural networks for approximating a class of inverse problems for PDEs. IMA Journal of Numerical Analysis 42(2): 981–1022 (2022), https://doi.org/10.1093/IMANUM/DRAB032
14. Zhu T., Zheng Q., Lu Y., Physics-Informed Fully Convolutional Networks for Forward Prediction of Temperature Field and Inverse Estimation of Thermal Diffusivity. J Comput Inf Sci Eng 24(11) (2024), https://doi.org/10.1115/1.4064555/1194685
15. Mazraeh H.D., Parand K., Hosseinzadeh M., Lansky J., Nulíček V., An improved water strider algorithm for solving the inverse Burgers Huxley equation. Scientific Reports 14(1): 1–16 (2024), https://doi.org/10.1038/s41598-024-78907-0
16. Mazraeh H.D., Parand K., A three-stage framework combining neural networks and Monte Carlo tree search for approximating analytical solutions to the Thomas–Fermi equation. J Comput Sci 87: 102582 (2025), https://doi.org/10.1016/j.jocs.2025.102582
17. Mazraeh H.D., Parand K., Approximate symbolic solutions to differential equations using a novel combination of Monte Carlo tree search and physics-informed neural networks approach. Eng Comput: 1–29 (2025), https://doi.org/10.1007/S00366-025-02127-X
18. Pourgholi R., Saeedi A., Applications of cubic B-splines collocation method for solving nonlinear inverse parabolic partial differential equations. Numer Methods Partial Differ Equ 33(1): 88–104 (2017), https://doi.org/10.1002/num.22073
19. Pourgholi R., Esfahani A., Abtahi M., A numerical solution of a two-dimensional IHCP. J Appl Math Comput 41: 61–79 (2013), https://doi.org/10.1007/s12190-012-0592-6
20. Pourgholi R., Dana H., Tabasi S.H., Solving an inverse heat conduction problem using genetic algorithm: Sequential and multi-core parallelization approach. Appl Math Model 38: 1948–1958 (2014), https://doi.org/10.1016/j.apm.2013.10.019
21. Ibrahim R.W., Jalab H.A., Karim F.K., Alabdulkreem E., Ayub M.N., A medical image enhancement based on generalized class of fractional partial differential equations. Quant Imaging Med Surg 12(1): 17283–17183 (2022), https://doi.org/10.21037/QIMS-21-15
22. Selvarajan S., A comprehensive study on modern optimization techniques for engineering applications. Artif Intell Rev 57(8): 1–52 (2024), https://doi.org/10.1007/S10462-024-10829-9
23. Hamdipour A, Basiri A, Zaare M, Numerical solution of nonlinear equations systems with improved meta-heuristic ARO algorithm, Soft Computing Journal. 14 (1) 184-203-(2024) 10.22052/scj.2025.254888.1244.
24. Jabbari M, Hosnavi R, Mohammadi M, Akhavan P. The application of DEA for the sake of assessing and evaluating the performance of researchers at research institutes: Case study of a research center. Soft Computing Journal. 2024;13(2):110-127. doi:10.22052/scj.2024.253495.1182
25. Behmanesh R, Majma N. Nephron-2 meta-heuristic algorithm (NOA-2), to solve optimization problems. Soft Computing Journal. 2023;11(2):62–71. doi:10.22052/scj.2023.248427.1104
26. Tang K., Meng C., Particle Swarm Optimization Algorithm Using Velocity Pausing and Adaptive Strategy. Symmetry (Basel) 16 (2024), https://doi.org/10.3390/sym16060661
27. Kassoul K., Zufferey N., Cheikhrouhou N., Brahim Belhaouari S., Exponential Particle Swarm Optimization for Global Optimization. IEEE Access 10: 78320–78344 (2022), https://doi.org/10.1109/ACCESS.2022.3193396
28. Lesnic D., Inverse Problems with Applications in Science and Engineering. Inverse Problems with Applications in Science and Engineering: 1–342 (2021), https://doi.org/10.1201/9780429400629/INVERSE-PROBLEMS-APPLICATIONS-SCIENCE-ENGINEERING-DANIEL-LESNIC/RIGHTS-AND-PERMISSIONS
29. Klibanov M.V., Li J., Inverse Problems and Carleman Estimates. Inverse Problems and Carleman Estimates (2021), https://doi.org/10.1515/9783110745481/HTML
30. Fouad M.M., El-Desouky A.I., Al-Hajj R., El-Kenawy E.S.M., Dynamic Group-Based Cooperative Optimization Algorithm. IEEE Access 8: 148378–148403 (2020), https://doi.org/10.1109/ACCESS.2020.3015892

31. Thaker M., Vanak A.T., Owen C.R., Ogden M.B., Niemann S.M., Slotow R., Minimizing predation risk in a landscape of multiple predators: Effects on the spatial distribution of African ungulates. Ecology 92: 398–407 (2011), https://doi.org/10.1890/10-0126.1
32. Swapna Y. et al., Applications of Partial Differential Equations in Fluid Physics. Communications on Applied Nonlinear Analysis 31(1): 207–220 (2024), https://doi.org/10.52783/CANA.V31.396
33. Huang S., Feng W., Tang C., He Z., Yu C., Lv J., Partial Differential Equations Meet Deep Neural Networks: A Survey. IEEE Trans Neural Netw Learn Syst (2025), https://doi.org/10.1109/TNNLS.2025.3545967
34. Dehghani M., Trojovský P., Osprey optimization algorithm: A new bio-inspired meta-heuristic algorithm for solving engineering optimization problems. Front Mech Eng 8 (2023), https://doi.org/10.3389/fmech.2022.1126450
35. Yang Y. et al., Advancements in Q-learning meta-heuristic optimization algorithms: A survey. Wiley Interdiscip Rev Data Min Knowl Discov 14(6): e1548 (2024), https://doi.org/10.1002/WIDM.1548