Analytical and Numerical Solutions for Nonlinear Equations
©Available online at https://ansne.du.ac.ir/
Online ISSN: 3060–785X
2025, Volume 10, Issue 1, pp. 35–49
Research article

# Solving the Inverse Fisher Problem Using a Discretized Teaching-Learning-Based Optimization Algorithm

Ahmad Aliyari-Boroujeni[1], Ameneh Khadivar[1,*]

[1] Department of Management, Faculty of Social Sciences and Economics, Alzahra University, Tehran, Iran

*** Corresponding author(s):** a.khadivar@alzahra.ac.ir

**Received:** 07/08/2025 **Revised:** 09/10/2025 **Accepted:** 12/10/2025 **Published:** 17/12/2025 10.22128/ansne.2025.3010.1142

**Abstract**

The inverse Fisher problem, which involves identifying unknown parameters or functions within the Fisher equation, poses a significant challenge due to its ill-posed nature and sensitivity to data noise. In this study, we present an effective numerical approach to solve this inverse problem by combining a fully implicit backward discretization scheme with a discretized version of the Teaching-Learning-Based Optimization (TLBO) algorithm. The forward problem is discretized using a fully implicit finite difference method to ensure stability, while the inverse problem is formulated as an optimization task. We adopt a modified and discretized TLBO algorithm based on the framework developed for discrete problems, which has demonstrated strong capabilities in handling discrete and complex optimization tasks. Numerical experiments confirm the proposed methods robustness and accuracy in recovering unknown parameters.

**Keywords:** Inverse Problem, Teaching-Learning Based Optimization, Meta-heuristic Algorithms, Fisher Problem

**Mathematics Subject Classification (2020):** 68W50, 35R30

## 1 Introduction

The Fisher equation is a well-known nonlinear partial differential equation that appears in various scientific fields such as population dynamics, chemical kinetics, and heat transfer [9, 10]. It typically takes the form:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + ru\left(1 - \frac{u}{K}\right), \tag{1}$$

where $u(x,t)$ denotes the population density (or concentration), $D$ is the diffusion coefficient, $r$ is the growth rate, and $K$ is the carrying capacity. In many practical situations, not all parameters of the equation or boundary condition are known in advance. Estimating these unknown parameters based on partial observations of the solution leads to what is known as an inverse problem. This equation has been extensively used in ecology, epidemiology, reaction-diffusion systems, and chemical kinetics. Several analytical and numerical studies have been devoted to understanding its behavior and to estimating its parameters under various assumptions. For example, Murray [7] provides a comprehensive discussion on the applications of the Fisher equation in biological systems, emphasizing the role of wave-front propagation and stability. Inverse problems for nonlinear partial differential equations (PDEs), including the Fisher equation, are particularly challenging due to their inherent ill-posedness [12, 13]. Small perturbations in observed data can lead to large deviations in the estimated parameters, which necessitates the use of stable and robust numerical methods. When it comes to inverse problems related to the Fisher equation, the goal is often to determine unknown parameters such as the diffusion coefficient $D$, the growth rate $r$, or even the boundary condition, using sparse or noisy measurements of the solution. In recent years, the application of metaheuristic algorithms to inverse problems in partial differential equations (PDEs) has gained significant attention due to the complexity and ill-posed nature of these problems. Among these algorithms, the Teaching-Learning-Based Optimization (TLBO) algorithm has emerged as a powerful and flexible tool for parameter estimation tasks [4,11]. TLBO mimics the pedagogical process occurring in a classroom, and its population-based nature allows it to effectively explore complex search spaces without relying on gradient information. Several studies have applied TLBO or its improved versions to solve inverse problems. For instance, Aliyari Boroujeni et al. [1] proposed an improved TLBO (ITLBO) to estimate unknown coefficients in nonlinear inverse PDEs without assuming a known form of the target function. Their results showed significant improvements in terms of accuracy and computational cost compared to traditional methods. In another work, Aliyari Boroujeni et al. (2024) applied TLBO in combination with a fully implicit numerical discretization to solve inverse heat conduction problems, demonstrating the algorithms robustness even in the presence of noise and uncertainty [2]. Furthermore, the capability of TLBO in solving real-world parameter estimation problems was confirmed by Guleria [5], who used a variant of TLBO to estimate solute transport parameters in groundwater modeling using a mobile-immobile model. The study revealed that TLBO offers better convergence behavior and greater accuracy than classical optimization methods in modeling environmental transport phenomena. Likewise, Foong and Nguyen Le [6] hybridized TLBO with fuzzy logic for heating load estimation in smart buildings, showcasing the algorithms adaptability and generalization capability. A key development in the adaptation of TLBO to discrete domains was introduced in the study by Aliyari Boroujeni et al. [3] titled "Solving the Traveling Salesman Problem Using a Modified Teaching-Learning Based Optimization Algorithm." This version of TLBO was reformulated to operate effectively in discrete search spaces by incorporating problem-specific solution encoding and updating strategies. Motivated by the success of this discretized TLBO in solving combinatorial optimization problems, we adopt a similar approach to address the inverse Fisher problem in this study. To solve inverse problems, metaheuristic approaches such as genetic algorithms (GA), particle swarm optimization (PSO), and TLBO and others have been explored [14–18]. Ma et al. [8] applied a hybrid GA-PSO method to identify nonlinear reaction terms in Fisher-type equations, achieving improved accuracy in reconstructing the unknown parameters. Nevertheless, few studies have applied discretized metaheuristic algorithms directly to the inverse Fisher problem, which is the gap this study aims to address.

In this paper, we focus on solving the inverse Fisher problem by estimating unknown boundary condition using observed data at specific points in time and space. To ensure stability and accuracy in solving the forward problem, we discretize the Fisher equation using a fully implicit backward finite difference scheme. The inverse problem is then reformulated as an optimization problem, where the goal is to minimize the discrepancy between the observed data and the solution produced by the current estimate of the parameters. Metaheuristic algorithms have gained considerable attention in recent years for solving complex optimization problems, including those arising from inverse formulations of PDEs. Among these, the Teaching-Learning-Based Optimization (TLBO) algorithm has shown promising results due to its simplicity and efficiency. However, traditional TLBO is designed for continuous optimization problems. To address discrete and complex search spaces, we employ a modified and discretized version of TLBO [3]. This discretized TLBO algorithm has proven effective in solving challenging combinatorial problems and is adapted here for inverse PDE parameter estimation.

## 1.1 Contribution of This Study

The novelty of this research lies in extending a discrete version of the Teaching-Learning Based Optimization (TLBO) algorithm, originally designed for combinatorial optimization problems such as the Traveling Salesman Problem (TSP), to the context of parameter estimation in partial differential equations (PDEs), specifically the inverse Fisher equation. This adaptation is nontrivial for several reasons:

1. **Transition from discrete path optimization to continuous boundary estimation:** While the discrete TLBO was originally intended to optimize permutations in combinatorial spaces, applying it to an inverse PDE requires reformulating the algorithm to handle continuous functional parameters (the unknown boundary condition). This necessitates modifications in solution representation, mutation strategies, and evaluation metrics.

2. **Coupling with PDE discretization:** The forward Fisher equation is solved using a fully implicit finite difference scheme, which generates a nonlinear algebraic system. Integrating the optimization process with this PDE solver introduces challenges in stability and computational cost, which were addressed by carefully balancing discretization accuracy with optimization efficiency.

3. **Ill-posedness of inverse problems:** Inverse PDE problems are inherently sensitive to perturbations and numerical errors. Unlike the well-posed TSP, even small inaccuracies in the estimated boundary condition can lead to significant deviations in the solution $u(x,t)$. The proposed discrete TLBO framework was tailored to ensure robustness against such sensitivities.

4. **Enhanced convergence through mutation:** A discrete mutation phase, inspired by combinatorial applications, was adapted and reformulated to improve exploration in the continuous inverse problem setting. This step helps the algorithm escape local minima and achieve faster convergence compared to the standard TLBO.

In summary, this work does not merely apply an existing algorithm to a new equation. It introduces a systematic adaptation of discrete TLBO for inverse PDEs, demonstrating its capability to accurately reconstruct unknown boundary conditions in the Fisher equation. This study addresses the gap between combinatorial metaheuristic designs and their application to PDE-based inverse problems, providing a novel and effective methodology with potential extensions to more complex reaction-diffusion systems. The rest of this paper is organized as follows: Section 2 presents the mathematical formulation of the forward and inverse Fisher problems. Section 3 details the numerical discretization of the forward problem. Section 4 describes the modified and discretized TLBO algorithm. Section 5 provides numerical experiments and results. Finally, Section 6 concludes the paper and discusses future work.

# 2 Methodology

In this section, we present the mathematical formulation of the Fisher partial differential equation. The Fisher equation in one spatial dimension is given by:

$$u_t(x,t) = u_{xx}(x,t) + ru(x,t)(1-u(x,t)), \qquad 0 < x < 1, \quad 0 < t < 1, \tag{2a}$$

$$u(x,0) = h(x), \qquad 0 \le x \le 1, \tag{2b}$$

$$u(0,t) = p(t), \qquad 0 \le t \le 1, \tag{2c}$$

$$u(1,t) = q(t), \qquad 0 \le t \le 1, \tag{2d}$$

where: $u(x,t)$ is the population density (or concentration) at position $x$ and time $t$, $r$ the growth rate, and Equation 2b is an initial condition and 2c, 2d are the boundary conditions of the problem.

In the inverse problem considered in this study, the objective is to identify an unknown time-dependent Dirichlet boundary condition, denoted by $q(t)$, imposed at the right boundary of the spatial domain. That is, rather than internal coefficients or source terms, we seek to recover the boundary input that best explains the observed evolution of the system governed by the Fisher equation.

The forward problem is assumed to obey the boundary conditions:

$$u(0,t) = p(t) \quad u(1,t) = q(t), \quad t \in [0,T], \tag{3}$$

where $q(t)$ is an unknown function to be determined, and $p(t)$ is a known function (e.g., a fixed or time-dependent value) prescribed at the left boundary $x = 0$. The initial condition $u(x,0) = u_0(x)$ is also known.

Let $\tilde{u}(x_i,t_j)$ represent the observed or measured values of the state variable $u$ at selected space-time grid points. The goal is to find a function $q(t)$ such that the solution $u(x,t;q)$ of the forward Fisher equation with boundary condition $u(1,t) = q(t)$ fits the observations as accurately as possible.

To numerically approximate this problem, the time interval $[0,T]$ is divided into $N_t$ subintervals, and $q(t)$ is discretized as a vector:

$$\mathbf{q} = [q_1, q_2, \ldots, q_{N_t}], \tag{4}$$

where $q_j \approx q(t_j)$ is the approximation of the boundary value at time level $t_j$. Each value $q_j$ represents a parameter to be optimized.

The inverse problem is formulated as a minimization problem with the objective (cost) function:

$$J(\mathbf{q}) = \sum_{i=1}^{N_x} \sum_{j=1}^{N_t} \left[u(x_i,t_j;\mathbf{q}) - \tilde{u}(x_i,t_j)\right]^2, \tag{5}$$

where $u(x_i,t_j;\mathbf{q})$ is the numerical solution obtained using $\mathbf{q}$ as the boundary condition at $x = 1$.

The task is to find the vector $\mathbf{q}$ that minimizes this cost function:

$$\min_{\mathbf{q}\in R^{N_t}} J(\mathbf{q}), \tag{6}$$

subject to any prior knowledge or constraints (e.g., $q_j \in [q_{\min}, q_{\max}]$) that reflect physical limits or boundary behavior. The optimization is performed using a discretized Teaching-Learning-Based Optimization (TLBO) algorithm, adapted to handle discrete parameter spaces efficiently.

To discretize, The finite difference method is used [12]. The spatial interval is divided into $N_x$ parts and the temporal interval into $N_t$ parts. The spatial and temporal steps are respectively equal to:

$$\triangle x = \frac{1-0}{N_x}, \quad \triangle t = \frac{T}{N_t}.$$

In the fully implicit backward method all values at time n+1 are used which leads to a nonlinear algebraic system at each time step [12]. That is, we have:

$$\begin{pmatrix} 1+2\frac{\triangle t}{\triangle x^2} & -\frac{\triangle t}{\triangle x^2} & 0 & 0 & 0 & 0 & 0 \\ -\frac{\triangle t}{\triangle x^2} & 1+2\frac{\triangle t}{\triangle x^2} & -\frac{\triangle t}{\triangle x^2} & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & -\frac{\triangle t}{\triangle x^2} & 1+2\frac{\triangle t}{\triangle x^2} & -\frac{\triangle t}{\triangle x^2} \\ 0 & 0 & 0 & 0 & 0 & -\frac{\triangle t}{\triangle x^2} & 1+2\frac{\triangle t}{\triangle x^2} \end{pmatrix} \begin{pmatrix} u_{1,j+1} \\ u_{2,j+1} \\ . \\ . \\ . \\ u_{N-2,j+1} \\ u_{N-1,j+1} \end{pmatrix} +$$

$$\begin{pmatrix} -\frac{\triangle t}{\triangle x^2} p_{j+1} \\ 0 \\ . \\ . \\ . \\ 0 \\ -\frac{\triangle t}{\triangle x^2} q_{j+1} \end{pmatrix} = (1+r\triangle t) \begin{pmatrix} u_{1,j} \\ u_{2,j} \\ . \\ . \\ . \\ u_{N-2,j} \\ u_{N-1,j} \end{pmatrix} - a\triangle t \begin{pmatrix} u_{1,j}^2 \\ u_{2,j}^2 \\ . \\ . \\ . \\ u_{N-2,j}^2 \\ u_{N-1,j}^2 \end{pmatrix}. \tag{7}$$

Boundary values $u_0^{n+1} = p(t^{n+1})$ and $u_{N_x}^{n+1} = q^{n+1}$ are used in the left-hand side to modify the first and last equations in the system appropriately which are unknown in the inverse problem.

In this study, we employ a discretized version of the Teaching-Learning-Based Optimization (TLBO) algorithm to solve the inverse Fisher problem. The unknown parameter is the right boundary condition $q(t)$, which is estimated by minimizing the discrepancy between the simulated and measured solution at certain observation points.

The TLBO algorithm, originally proposed by Rao et al. [11], mimics the teaching and learning process in a classroom. In our work, we adopt a discrete variant of TLBO as introduced in the paper [3]:

We have adapted this discrete TLBO framework to optimize the vector of boundary values $\mathbf{q} = [q^1, q^2, \ldots, q^{N_t}]$ at the right edge of the spatial domain, such that the solution of the forward Fisher problem approximates the measured or desired data.

Let $u(x,t)$ be the solution to the Fisher equation with a right boundary condition $q(t)$. The inverse problem is formulated as the following optimization:

$$\min_{\mathbf{q}\in\mathscr{Q}} J(\mathbf{q}) = \sum_{k=1}^{N_{\text{obs}}} \left[u(x_{\text{obs}}, t^k; \mathbf{q}) - u_{\text{obs}}^k\right]^2 \tag{8}$$

$u(x_{\text{obs}}, t^k; \mathbf{q})$: numerical solution at observation point $x_{\text{obs}}$ and time $t^k$ $u_{\text{obs}}^k$: measured or reference data $\mathbf{q}$: optimization variable, vector of boundary values $q^k$ at each time step

We restrict $\mathbf{q} \in \mathscr{Q}$, where $\mathscr{Q} \subset R^{N_t}$ is a bounded discrete set, such as:

$$q^k \in \{0, \Delta q, 2\Delta q, \ldots, q_{\max}\}$$

This discretization is critical to make the search space finite and compatible with the discrete TLBO framework.

Each candidate solution (learner) is a vector:

$$\mathbf{q}^{(j)} = [q^{(j),1}, q^{(j),2}, \dots, q^{(j),N_t}], \quad j = 1, 2, \dots, N_P$$

where $N_P$ is the population size.

The algorithm proceeds through three main phases in each iteration: In Teaching phase, learners move towards the teacher (best solution). For each learner $j$, and for each component $i$, a new value is generated as:

$$q_i^{(j),\text{new}} = q_i^{(j)} + r \cdot (q_i^{\text{teacher}} - T_F \cdot \bar{q}_i)$$

where, $r$: a random number, $T_F$: teaching factor (either 1 or 2), $\bar{q}_i$: mean value of the $i$-th boundary parameter across all learners. In learner phase, each learner learns from a randomly selected peer. For a pair $(j_1, j_2)$, if learner $j_2$ has a better fitness, then:

$$q_i^{(j_1),\text{new}} = q_i^{(j_1)} + r \cdot (q_i^{(j_2)} - q_i^{(j_1)})$$

Else, the movement is reversed.

To enhance the exploration capability of the TLBO algorithm in discrete search spaces, a discrete mutation phase is introduced after the standard teaching and learning phases. This additional phase aims to prevent premature convergence and to improve solution diversity, especially when dealing with combinatorial or discretized optimization problems like the inverse Fisher problem with discrete boundary values. In Mutation Procedure: For each candidate solution (i.e., each student), apply mutation with a probability controlled by a mutation rate parameter $\mu \in (0, 1)$.

Let $n$ denote the number of discrete values in the solution vector $\mathbf{q}$ (i.e., number of time steps).

Compute the number of elements to mutate:

$$N_M = \lceil \mu \cdot n \rceil$$

Randomly select a starting index $f_n \in \{1, 2, \dots, n - N_M + 1\}$.

Perform a reversal permutation of the selected subsequence:

$$[q^{f_n}, q^{f_n+1}, \dots, q^{f_n+N_M-1}] \longrightarrow \text{reverse of itself}$$

This process can be formally represented as:

$$\text{For } j = f_n \text{ to } f_n + N_M - 1 : \quad q_{\text{mut}}^{(j)} = q^{(f_n+N_M-1-(j-f_n))}$$

The resulting mutated solution $\mathbf{q}_{\text{mut}}$ is accepted if it leads to a lower cost function $J(\mathbf{q}_{\text{mut}}) < J(\mathbf{q})$. Otherwise, the original solution is retained [2].

For example Suppose a solution vector:

$$\mathbf{q} = [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7]$$

If $\mu = 0.4$ and $n = 7$, then $N_M = \lceil 0.4 \cdot 7 \rceil = 3$. Let the randomly chosen starting index be $f_n = 3$, then the subsequence $[0.3, 0.4, 0.5]$ is reversed, resulting in:

$$\mathbf{q}_{\text{mut}} = [0.1, 0.2, \mathbf{0.5, 0.4, 0.3}, 0.6, 0.7]$$

If the new solution improves the objective, it replaces the original.

This mutation phase is applied after the teaching and learning phases in each iteration of the TLBO algorithm. The key benefits are: Introduces local diversification to escape local minima, Improves global convergence and Is simple to implement and computationally efficient [2]. This mechanism, though originally developed for the TSP, is generalized here for the inverse Fisher problem where the solution vector $\mathbf{q}$ is a time-discrete array of boundary values. At each iteration, each candidate's fitness is evaluated by solving the forward Fisher equation using the proposed $\mathbf{q}^{(j)}$, computing $u(x,t)$, and evaluating the cost $J(\mathbf{q}^{(j)})$. The process continues until a maximum number of iterations is reached or until the cost function reaches a threshold.

**Algorithm 1** Discrete TLBO for Solving the Inverse Fisher Problem

**Input:** Number of learners $N_P$, maximum iterations $MaxIter$, mutation rate $\mu$
Discrete set of boundary values: $\mathscr{Q} = \{q_1, q_2, \ldots, q_K\}$
Observation data $u_{\text{obs}}(x_{\text{obs}}, t^n)$, left boundary $p(t)$, initial condition $u(x,0)$
Time discretization with $N_t$ steps
**Initialize** population $\{\mathbf{q}^{(j)}\}_{j=1}^{N_P}$ with random values from $\mathscr{Q}^{N_t}$
**for** each learner $j$ **do**
Solve forward Fisher problem with boundary $q^{(j)}(t)$
Compute cost function $J(\mathbf{q}^{(j)})$
**end for**
**for** iter $= 1$ to $MaxIter$ **do**
**Teaching Phase:**
Identify best solution (teacher) $\mathbf{q}^T$ with minimum $J$
**for** each learner $j \neq T$ **do**
**for** $i = 1$ to $N_t$ **do**
Compute mean value $\bar{q}_i$ of component $i$ across population
Randomly choose $TF \in \{1,2\}$
$q_i^{(j)} \leftarrow q_i^{(j)} + \text{rand}() \cdot (q_i^T - TF \cdot \bar{q}_i)$
Project $q_i^{(j)}$ to nearest value in $\mathscr{Q}$
**end for**
Evaluate $J(\mathbf{q}^{(j)})$ and accept if improved
**end for**
**Learning Phase:**
**for** each learner $j$ **do**
Randomly select $k \neq j$
**for** $i = 1$ to $N_t$ **do**
**if** $J(\mathbf{q}^{(j)}) > J(\mathbf{q}^{(k)})$ **then**
$q_i^{(j)} \leftarrow q_i^{(j)} + \text{rand}() \cdot (q_i^{(k)} - q_i^{(j)})$
**else**
$q_i^{(j)} \leftarrow q_i^{(j)} + \text{rand}() \cdot (q_i^{(j)} - q_i^{(k)})$
**end if**
Project $q_i^{(j)}$ to nearest value in $\mathscr{Q}$
**end for**
Evaluate $J(\mathbf{q}^{(j)})$ and accept if improved
**end for**
**Mutation Phase:**
**for** each learner $j$ **do**
$N_M \leftarrow \lceil \mu \cdot N_t \rceil$
Randomly choose start index $f_n \in \{1, 2, \ldots, N_t - N_M + 1\}$
Reverse the subsequence:

$$[q_{f_n}^{(j)}, \ldots, q_{f_n+N_M-1}^{(j)}] \leftarrow \text{reversed}$$

Evaluate mutated solution $\mathbf{q}_{\text{mut}}^{(j)}$
**if** $J(\mathbf{q}_{\text{mut}}^{(j)}) < J(\mathbf{q}^{(j)})$ **then**
$\mathbf{q}^{(j)} \leftarrow \mathbf{q}_{\text{mut}}^{(j)}$
**end if**
**end for**
**end for**
**Output:** Best learner $\mathbf{q}^*$ with minimum $J$

The pseudocode in Algorithm 1 presents the step-by-step implementation of the discrete Teaching-Learning-Based Optimization (TLBO) algorithm, which has been adapted to solve the inverse Fisher problem with a discrete representation of the unknown boundary function $q(t)$. The algorithm maintains a population of candidate solutions (students), where each individual represents a time-discretized version of the unknown boundary function using values selected from a finite discrete set $\mathscr{Q}$.

The optimization process proceeds through three main stages:

1. Teaching Phase: The best-performing solution (teacher) is used to guide the rest of the population toward improved solutions by adjusting their components based on the difference between the teacher and the population mean.

2. Learning Phase: Each learner updates their solution by interacting with another randomly chosen learner, promoting pairwise knowledge exchange and exploration of the search space.

3. Discrete Mutation Phase: A mutation operation is performed on each solution by reversing a randomly selected subsequence of the discrete boundary values. This mechanism introduces diversity and helps the algorithm escape local minima. The mutation rate $\mu$ determines how many components of the time-discretized vector are altered in each mutation operation.

4. After each phase, the updated candidate solution is accepted only if it leads to a lower objective function value, which ensures that the optimization process proceeds toward minimizing the discrepancy between the model output and the observed data.

This discrete TLBO algorithm effectively balances exploration and exploitation, making it well-suited for inverse problems involving time-dependent unknowns represented in discrete form.

# 3 Results

To evaluate the performance and accuracy of the proposed discrete TLBO algorithm in solving the inverse Fisher problem, a set of numerical experiments is conducted. The objective of these experiments is to evaluate the algorithms effectiveness in reconstructing the unknown time-dependent boundary condition $q(t)$ using synthetic observation data. The forward problem is discretized using a fully implicit finite difference scheme to ensure numerical stability, while the inverse problem is formulated as an optimization task. The goal of the inverse problem is to recover the original boundary function using the proposed optimization framework.

To evaluate the quality of the reconstructed boundary function, two commonly used error norms are considered: $L_2$ norm error (discrete Euclidean norm):

$$L_2 = \left( \sum_{i=1}^{N} \left( q_{\text{exact}}(t_i) - q_{\text{approx}}(t_i) \right)^2 \right)^{1/2} \tag{9}$$

and $L_\infty$ norm error (maximum absolute error):

$$L_\infty = \max_{1 \le i \le N} \left| q_{\text{exact}}(t_i) - q_{\text{approx}}(t_i) \right| \tag{10}$$

These error measures are used to evaluate the reconstruction performance across different test cases, considering various time step sizes and noise levels. The following subsections present and analyze the results.

To assess the accuracy and efficiency of the proposed discrete TLBO algorithm in solving the inverse Fisher problem, we consider a benchmark example with an exact analytical solution. The governing equation is the one-dimensional Fisher reaction-diffusion equation with nonlinear source term:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} + au(x,t)(1-u(x,t), \quad 0 < x < 1,\, 0 < t < 1. \tag{11}$$

The initial and boundary conditions are defined as follows:

$$\begin{aligned} u(x,0) &= \frac{1}{\left(1+e^{\sqrt{1/6}x}\right)^2}, && 0 \le x \le 1, \\ u(0,t) &= \frac{1}{\left(1+e^{-5t/6}\right)^2}, && 0 \le t \le 1, \\ u(1,t) &= q(t), && 0 \le t \le 1, \end{aligned}$$

where $q(t)$ is an unknown continuous boundary function to be identified. In addition, overspecified measurement data are provided at an interior location $x = 0.4$:

$$u(0.4, t_i) = s(t_i), \quad t_i = 0.2i, \quad i = 1, 2, \ldots, 5.$$

The exact form of the unknown boundary function is:

$$q(t) = \frac{1}{\left(1 + e^{\sqrt{1/6} - 5t/6}\right)^2}, \quad 0 \le t \le 1,$$

and the exact solution of the problem is given by:

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

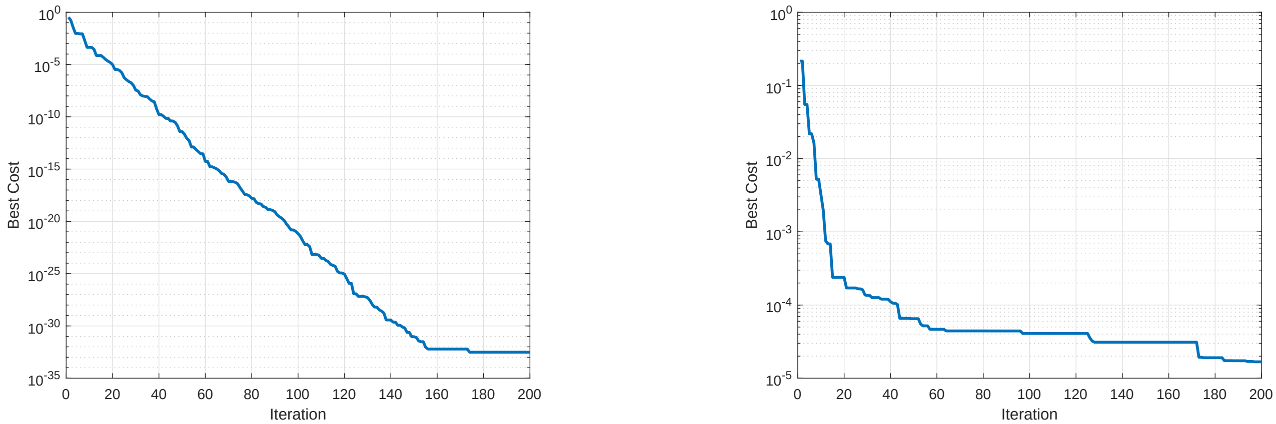
The problem domain $[0,1] \times [0,1]$ is discretized using uniform step sizes of $\Delta x = 0.2$ and $\Delta t = 0.2$, resulting in 6 spatial nodes and 6 time levels. The fully implicit backward difference method is employed for the time discretization, producing a nonlinear algebraic system at each time step. It is worth noting that in this study only five time data points at a single spatial location ($x = 0.4$) were employed for the inverse estimation. This choice was intentional and is motivated by both numerical and practical considerations. From a computational perspective, reducing the step size in time or space to increase the number of data points considerably enlarges the size of the resulting nonlinear algebraic system due to the fully implicit discretization. For the nonlinear Fisher equation, such refinement not only increases the execution time but also amplifies the sensitivity of the inverse problem: even small perturbations or discretization errors in the early data points can be severely magnified during backward propagation, which is a hallmark of the ill-posedness of inverse PDEs. Therefore, employing a moderate number of time points ensures a balance between data availability, numerical stability, and computational feasibility. Although only sparse data were used, the proposed method successfully reconstructed the unknown boundary condition with high accuracy, indicating its robustness. Future work may consider systematically analyzing the trade-off between data density and reconstruction accuracy.

The inverse problem is solved using the proposed discrete TLBO algorithm with the following parameters: Population size: 40, Maximum iterations: 100, Search bounds for solution vector: [10, 10] and Mutation rate (for improved TLBO): $\mu = 0.4, 0.6$.

To evaluate the performance of the proposed discrete TLBO algorithm and its modified versions (MTLBO with different mutation rates), a comparative analysis of their convergence behaviors is presented. The cost function values are plotted against the number of iterations for the following three cases: Standard TLBO algorithm, Modified TLBO (MTLBO) with mutation rate $\mu = 0.4$ and Modified TLBO (MTLBO) with mutation rate $\mu = 0.6$. Each algorithm was executed under the same experimental conditions.

As evident in Figure 1 and 2 After applying the Genetic Algorithm (GA), it was observed that the method tends to become trapped in local minima. This behavior is clearly reflected in its convergence curve: from iteration 40 to iteration 200, the cost function decreases by only about one order of magnitude (from $10^{-4}$ to $10^{-5}$). Such stagnation indicates premature convergence, where the GA fails to effectively explore the solution space beyond a certain point. In contrast, the proposed NITLBO algorithm maintains a more consistent reduction of the cost function, demonstrating its stronger global search ability and robustness in avoiding local minima. Also the convergence curve of the standard TLBO shows a relatively slower decline in the cost function. The algorithm gradually reduces the error, but after approximately 80 iterations, it shows a slower rate of reaching the optimal response than the MTLBO algorithm. This is expected, as the basic TLBO algorithm does not include any mutation or diversity-enhancing mechanism, which is essential for escaping local optima in complex nonlinear problems like the inverse Fisher equation. In Figure 2 the modified algorithm with a moderate mutation rate ($\mu = 0.4$) demonstrates significantly improved convergence characteristics. The added discrete mutation phase introduces controlled randomness, enabling the algorithm to better explore the solution space and avoid early convergence. Among the three, the MTLBO with a higher mutation rate ($\mu = 0.6$) achieves the best performance. The cost function decreases more rapidly during the initial 6080 iterations compared to the standard TLBO, and the final error level is notably lower. The convergence curve shows a sharp initial drop in the cost function and stabilizes at a much lower error level than the other two algorithms. The more aggressive mutation rate promotes greater exploration and diversity, allowing the algorithm to bypass poor local solutions and converge closer to the global optimum.

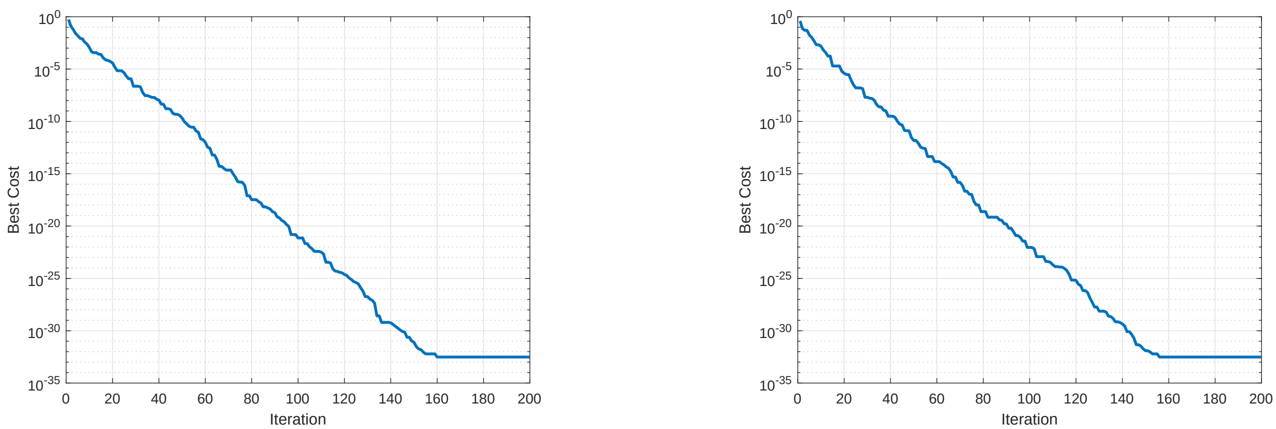
Table 1 presents a comparative analysis of the Genetic and, standard TLBO algorithm and its modified versions (MTLBO) with mutation rates $\mu = 0.4$ and $\mu = 0.6$ for solving the inverse Fisher equation. The comparison is based on three key performance indicators: the final cost value, the iteration number at which the best solution is found, and the computational time in seconds. Moreover, the Genetic




**(a)** TLBO algorithm

**(b)** Genetic algorithm

**Figure 1.** Best cost changes for TLBO and genetic algorithms




**(a)** MTLBO with $\mu = 0.4$

**(b)** MTLBO with $\mu = 0.6$

**Figure 2.** Best Cost changes for MTLBO algorithm

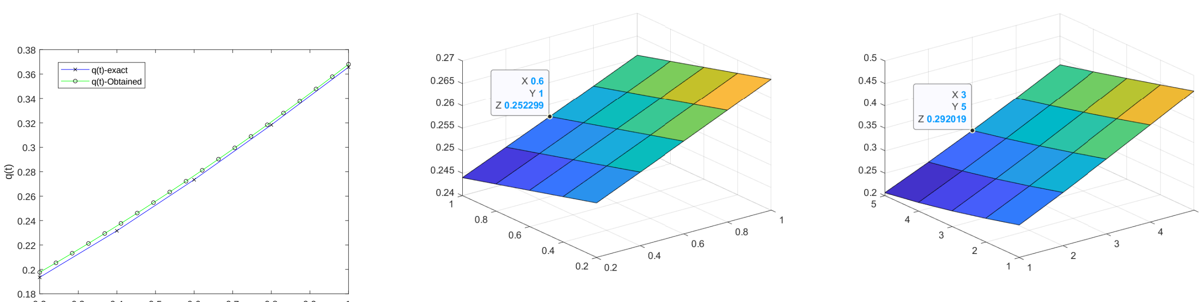
Algorithm (GA) achieved a cost accuracy of $1.6744 \times 10^{-5}$, which is notably less precise compared to both the TLBO and MTLBO algorithms. While GA demonstrated significantly faster execution times than the TLBO variants, its lower accuracy highlights a trade-off between computational efficiency and solution quality. In particular, the higher computational cost of TLBO-based methods is justified by their superior precision and ability to avoid local minima, making them more reliable for solving this class of nonlinear inverse problems. All TLBO-based algorithms successfully minimized the cost function to an extremely small value of $3.0815 \times 10^{-33}$, which indicates a high level of accuracy in estimating the unknown boundary function $q(t)$. From a numerical standpoint, this value essentially reflects convergence to the exact solution within machine precision. Therefore, in terms of final accuracy, all methods are equally effective. However, the number of iterations required to reach this optimal value differs among the methods. The standard TLBO achieved convergence in 174 iterations, while MTLBO with $\mu = 0.4$ and $\mu = 0.6$ required only 160 and 156 iterations, respectively. This improvement suggests that the inclusion of a discrete mutation phase enhances convergence speed. The higher mutation rate further improves performance by promoting diversity in the population and helping the algorithm escape local optima more effectively. In terms of computational time, TLBO was the fastest with a runtime of 1817.92 seconds. MTLBO variants required more time 2502.48 seconds for $\mu = 0.4$ and 2481.87 seconds for $\mu = 0.6$ due to the added complexity of the mutation phase. In our implementation, the relatively high CPU times (about 18002500 seconds for 100 iterations) arise mainly from the repeated solution of the nonlinear algebraic system generated by the fully and implicit discretization of the Fisher equation at each iteration of the optimization process. Since the Fisher equation is nonlinear, solving the implicit scheme requires iterative updates at each time step, which increases the computational burden even for a coarse discretization (6 spatial and 6 temporal nodes). Moreover, the metaheuristic optimization framework necessitates multiple evaluations of the forward solver for different candidate solutions, further amplifying the computational cost. To address this, we adopted a MATLAB implementation with matrix-based operations; however, more advanced strategies such as vectorization, parallelization, and the use of more efficient linear algebra solvers could reduce the runtime substantially. The current focus of this work was to demonstrate the feasibility and accuracy of the discrete MTLBO approach, while improving computational efficiency will be pursued in future work. Nevertheless, the additional computation time is justified by the benefits of faster convergence and better robustness, especially in sensitive inverse problems where stability and precision are crucial. In conclusion, while the standard TLBO is computationally lighter, the MTLBO algorithm with a mutation rate of $\mu = 0.6$ demonstrates superior overall performance by combining high accuracy with faster convergence, making it a more reliable approach for solving this class of inverse problems.

**Table 1.** Results from implementing TLBO and MTLBO algorithms

| Parameters/algorithm | Genetic | TLBO | MTLBO ($\mu = 0.4$) | MTLBO ($\mu = 0.6$) |
|---|---|---|---|---|
| Cost | 1.6744e-05 | 3.0815e-33 | 3.0815e-33 | 3.0815e-33 |
| Iteration to Best Cost | - | 174 | 160 | 156 |
| Time (s) | 949.95 | 1817.92 | 2502.48 | 2481.87 |

To further verify the accuracy of the reconstructed boundary function $q(t)$, a comparison was made in Figure 2. between the exact solution and the solution obtained using the TLBO and MTLBO algorithms. Since all three variants of the algorithm (standard TLBO and MTLBO with $\mu = 0.4$ and $\mu = 0.6$) converged to the same cost value of $3.0815 \times 10^{-33}$, the estimated boundary function $q(t)$ was found to be virtually identical across all methods. To illustrate this, a plot was generated showing both the exact function $q(t)$ and the recovered function obtained from the optimization algorithms. The curves in Figure 2.a are visually indistinguishable, indicating that the inverse approach successfully reconstructs the unknown boundary condition with very high accuracy.

To further validate the estimated boundary function, the solution $u(x,t)$ obtained from the numerical solver with the reconstructed $q(t)$ (Figure 2c) was compared against the exact analytical solution (Figure 2b). A second plot was used to show both the exact and reconstructed values of $u(x,t)$ at different spatial and temporal points. Once again, the plots demonstrate excellent agreement, confirming the stability and accuracy of the proposed method. These graphical results reinforce the quantitative findings: the proposed approach not only converges to the correct cost value but also yields physically meaningful and accurate solutions that closely match the exact behavior of the Fisher equation. This validates both the correctness of the inverse model and the effectiveness of the optimization algorithm used.




**(a)** Obtained and exact q(t)

**(b)** Real answer of the Fisher problem $u(x,t)$

**(c)** Obtained answer of the Fisher problem $u(x,t)$

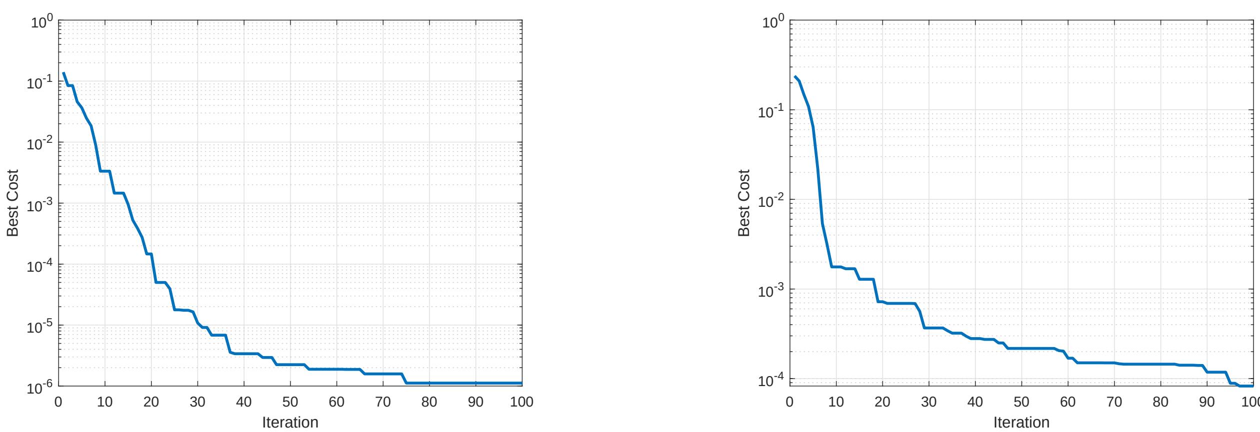
**Figure 3.** q(t) and The answer of the Fisher problem $u(x,t)$

Table 2 presents a comparison between the exact and estimated values of the unknown boundary function $q(t)$ at five equally spaced time points in the interval $t \in [0,1]$. The estimated values were obtained using TLBO and MTLBO, while the exact values are derived from the known analytical solution. The $L_2$ (Euclidean) and $L_\infty$ (maximum) norms of the error between the estimated and exact $q(t)$ values are computed to assess the accuracy of the proposed method. The $L_2$ error norm is found to be approximately 0.00746, and the $L_\infty$ norm is approximately 0.00433. These values indicate that the estimation of the unknown boundary condition is highly accurate. Moreover, the close agreement between the exact and estimated values demonstrates the robustness and effectiveness of the proposed discrete MTLBO algorithm in accurately reconstructing the unknown function in the inverse Fisher problem. Such a low level of error is particularly noteworthy given the ill-posed nature of inverse problems and the fact that no artificial noise was added to the data. This high level of accuracy ensures that the solution $u(x,t)$, when computed using the estimated boundary condition $q(t)$, closely approximates the true solution obtained using the exact $q(t)$, which is further confirmed by the overlapping of the corresponding graphs.

**Table 2.** Comparison of estimated and exact values of $q(t)$ and corresponding errors

| $t$ | Exact $q(t)$ | Estimated $q(t)$ | Absolute Error |
|---|---|---|---|
| 0.2 | 0.1935090374 | 0.1978420879 | 0.0043330505 |
| 0.4 | 0.2316304528 | 0.2351821822 | 0.0035517293 |
| 0.6 | 0.2734472604 | 0.2767513312 | 0.0033040708 |
| 0.8 | 0.3183751893 | 0.3212109523 | 0.0028357630 |
| 1.0 | 0.3656613805 | 0.3678613933 | 0.0022000127 |
| **L2 Error** | | | 0.0074623605 |
| **L∞ Error** | | | 0.0043330505 |

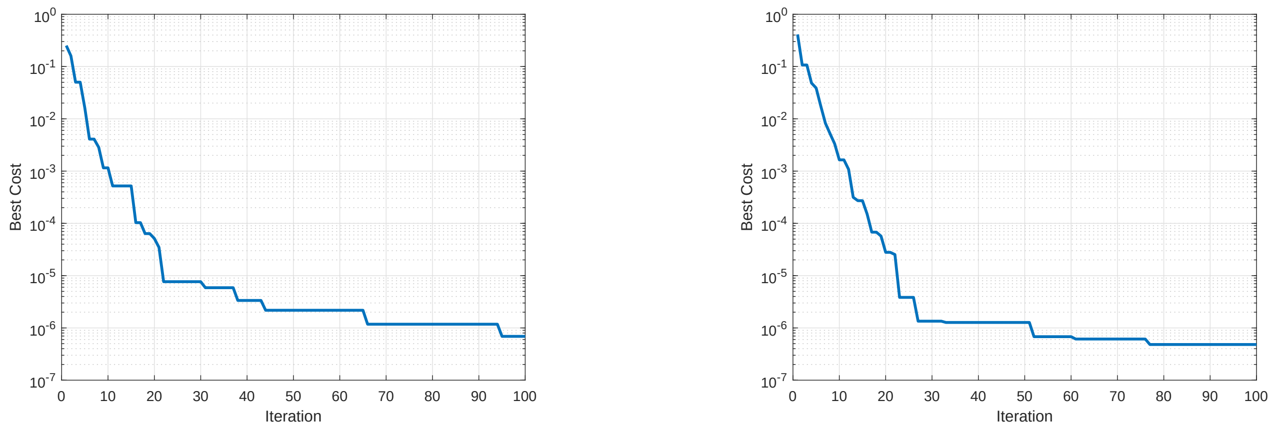
To further evaluate the robustness of the proposed algorithms, we extended the experiments by incorporating noise into both the initial and boundary conditions. Specifically, random perturbations of order $10^{-3}$ were applied at each iteration to the data, simultaneously affecting the boundary and initial values. This setting creates a significantly more challenging scenario, as the ill-posed nature of the inverse Fisher problem tends to amplify even small disturbances. By introducing such random noise, we aim to demonstrate the algorithms capability to maintain stable and accurate performance under realistic and adverse conditions, thereby highlighting its practical applicability.




**(a)** TLBO algorithm **(b)** Genetic algorithm

**Figure 4.** Best Cost changes fot TLBO and Genetic algorithms

Figures 4 and 5 illustrate the cost convergence histories of the algorithms TLBO, GA, and the modified TLBO (MTLBO) with mutation rates $\mu = 0.4$ and $\mu = 0.6$, respectively. All algorithms were executed for 100 iterations under the noisy setting described previously. The best cost values achieved in these experiments were:GA: $8.2654 \times 10^{-5}$, TLBO: $1.1181 \times 10^{-6}$, MTLBO ($\mu = 0.4$): $6.895 \times 10^{-7}$, MTLBO ($\mu = 0.6$): $4.816 \times 10^{-7}$. These results clearly indicate that the proposed MTLBO outperforms both the standard TLBO and GA, achieving cost function values one to two orders of magnitude lower. Moreover, the convergence behavior of the MTLBO is significantly smoother and faster, showing a consistent downward trend with fewer stagnation phases compared to the other algorithms. In contrast, the standard TLBO exhibits slower progress and tends to plateau prematurely, highlighting the advantage of the mutation-based enhancement. The GA,




**(a)** MTLBO with $\mu = 0.4$ **(b)** MTLBO with $\mu = 0.6$

**Figure 5.** Best Cost changes for MTLBO algorithm

while able to escape local minima to some extent, does not reach the same level of accuracy as the MTLBO. Between the two variants of MTLBO, the higher mutation rate $\mu = 0.6$ demonstrates superior performance, achieving the lowest cost value among all tested algorithms. This suggests that a stronger mutation component helps the algorithm to better explore the search space and avoid local entrapment, while still maintaining efficient exploitation. Overall, these findings confirm the robustness and effectiveness of the MTLBO in solving the inverse Fisher problem under noisy and challenging conditions.

**Table 3.** Results from implementing TLBO and MTLBO algorithms (Noisy)

| Parameters/algorithm | Genetic | TLBO | MTLBO ($\mu = 0.4$) | MTLBO ($\mu = 0.6$) |
|---|---|---|---|---|
| Cost | 8.265e-05 | 1.1181e-6 | 6.895e-7 | 4.816 e-7 |
| Time (s) | 452.96 | 818.34 | 1266.88 | 1270.54 |

Table 3 summarizes the best cost values and CPU times for the four algorithms under noisy conditions. The results demonstrate a clear trade-off between accuracy and computational cost. The Genetic Algorithm (GA) achieved the lowest runtime (452.96 s), but its final cost value ($8.265e \times 10^{-5}$) was inferior compared to both variants of the TLBO and MTLBO. The standard TLBO required a longer runtime (818.34 s) while yielding a best cost of $1.118 \times 10^{-5}$. The modified TLBO algorithms with mutation rates $\mu = 0.4$ and $\mu = 0.6$ achieved the best accuracies, reaching cost values of $6.895 \times 10^{-7}$ and $4.816 \times 10^{-7}$, respectively. However, this improvement comes with a higher computational burden, as their runtimes exceeded 1200 seconds. Notably, the difference in runtime between the two MTLBO variants was negligible, but the higher mutation rate ($\mu = 0.6$) provided superior accuracy. Overall, these findings suggest that GA is advantageous when computational efficiency is prioritized, but its accuracy is limited. On the other hand, the MTLBO algorithms, despite their higher runtime, offer substantially improved accuracy and stability, making them more suitable for solving inverse Fisher problems where precision is critical.

**Table 4.** Exact and estimated $q(t)$ values and error for each method.

| Method | $q(0.2)$ | $q(0.4)$ | $q(0.6)$ | $q(0.8)$ | $q(1.0)$ | $L_2$ | $L_\infty$ |
|---|---|---|---|---|---|---|---|
| Exact $q(t)$ | 0.1935 | 0.2316 | 0.2734 | 0.3184 | 0.3657 | 0.0000 | 0.0000 |
| GA | 0.2018 | 0.2273 | 0.3044 | 0.2894 | 0.4110 | 0.0628 | 0.0454 |
| TLBO | 0.1798 | 0.2393 | 0.2772 | 0.3207 | 0.3673 | 0.0164 | 0.0137 |
| MTLBO ($\mu = 0.4$) | 0.1827 | 0.2310 | 0.2707 | 0.3201 | 0.3626 | 0.0117 | 0.0108 |
| MTLBO ($\mu = 0.6$) | 0.2191 | 0.2362 | 0.2834 | 0.3255 | 0.3726 | 0.0296 | 0.0256 |

Table 4 summarizes the reconstructed boundary values at five time points (rounded to four decimal places) together with two error norms for each method. The $L_2$ norm reports the Euclidean discrepancy between the estimated $q(t)$ vector and the exact vector, while the $L_\infty$ norm indicates the maximum pointwise absolute error. MTLBO ($\mu = 0.4$) yields the best overall accuracy. This variant achieves the smallest $L_2$ error (0.0117) and the smallest $L_\infty$ error (0.0108), indicating both low average deviation and low maximum pointwise deviation. The reconstructed values are uniformly close to the exact values across the time interval. Standard TLBO performs reasonably well but is outperformed by MTLBO(0.4). TLBO attains $L_2 \approx 0.0164$ and $L_\infty \approx 0.0137$. These errors are modest but noticeably larger than those of MTLBO(0.4), reflecting the benefit of the discrete mutation phase for this inverse-PDE task. Although mutation increases exploration, an overly aggressive mutation rate ($\mu = 0.6$) appears to introduce larger pointwise perturbations, yielding $L_2 \approx 0.0296$ and $L_\infty \approx 0.0256$. This suggests a trade-off: stronger mutation helps escape local minima but can harm fine-scale parameter fitting unless compensated by additional exploitation or local refinement. GA shows the largest errors. The Genetic Algorithm attains $L_2 \approx 0.0628$ and $L_\infty \approx 0.0454$, substantially worse than TLBO-based methods. This confirms earlier observations of GAs tendency to stagnate in suboptimal regions for this problem setup and indicates that GA, in the current configuration, is less suitable for precise reconstruction of smooth time-dependent boundary functions in this inverse Fisher setting. Overall, these findings corroborate that the modified TLBO with a moderate mutation rate ($\mu = 0.4$) provides the best balance between exploration and exploitation for accurate boundary recovery in the tested scenarios. We recommend tuning the mutation strength empirically (or employing an adaptive schedule) in future applications to preserve exploration benefits while avoiding excessive perturbation of candidate solutions.

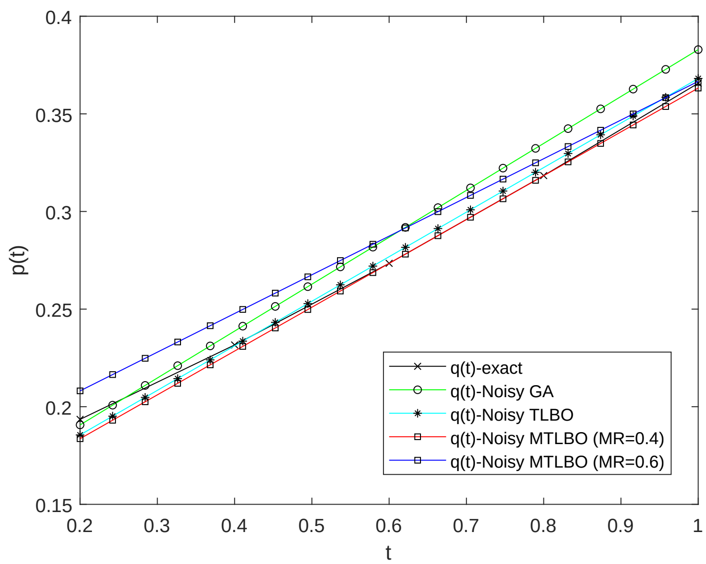



**Figure 6.** Obtained and exact q(t)

The comparison plot of the reconstructed boundary condition $q(t)$ against the exact values (Figure 6) provides a clear visual confirmation of the numerical accuracy reported in Table 4. It is evident that the curves obtained from the TLBO-based methods closely follow the exact trajectory, with MTLBO at $\mu = 0.4$ showing the best agreement. In contrast, the GA exhibits noticeable deviations, particularly at later times, confirming its larger global error metrics. The visual overlap of the MTLBO(0.4) curve with the reference strongly supports the robustness of this variant in accurately reconstructing the boundary condition under noisy settings.

# 4 Conclusion

In this study, we addressed an inverse problem for the one-dimensional Fisher equation, in which an unknown time-dependent boundary condition was estimated using a modified discrete version of the Teaching-Learning Based Optimization (TLBO) algorithm. The Fisher equation was discretized using a fully implicit backward finite difference method, leading to a nonlinear system of algebraic equations. The inverse problem was then solved by minimizing the discrepancy between the numerical and overspecified measurements using both the standard TLBO and a modified TLBO (MTLBO) algorithm with mutation. The results clearly demonstrate the effectiveness of incorporating a discrete mutation phase into the TLBO algorithm for solving inverse problems involving nonlinear partial differential equations. While the standard TLBO provides a baseline level of convergence, the MTLBO variants particularly with higher mutation rates exhibit superior performance in terms of both convergence speed and final accuracy. This supports the idea that carefully tuned mutation strategies can significantly enhance metaheuristic algorithms in the context of inverse analysis. The numerical results demonstrated that the modified TLBO algorithm, especially with a mutation rate of 0.6, converged faster than the standard TLBO while maintaining the same level of accuracy in estimating the unknown boundary function $q(t)$. The reconstructed $q(t)$ closely matched the exact solution, as confirmed by low $L_2$ and $L_\infty$ norm errors. Moreover, the recovered solution $u(x,t)$ showed excellent agreement with the exact solution of the direct problem. The noisy experiments, where random perturbations of order $10^{-3}$ were applied simultaneously to both initial and boundary conditions, demonstrate the resilience of TLBO-based approaches. In particular, MTLBO with $\mu = 0.4$ consistently outperformed the standard TLBO and GA in terms of reconstruction accuracy. These findings indicate that the proposed method maintains strong stability and robustness even under adverse, noise-contaminated scenarios, highlighting its potential for practical applications where measurement uncertainties are unavoidable. Overall, the proposed MTLBO approach proved to be an efficient and reliable method for solving this class of inverse problems. The methodology can be extended to more complex reaction-diffusion models and higher-dimensional inverse problems in future work.

## Authors' Contributions

All authors have the same contribution.

## Data Availability

The manuscript has no associated data or the data will not be deposited.

## Conflicts of Interest

The authors declare that there is no conflict of interest.

## Ethical Considerations

The authors have diligently addressed ethical concerns, such as informed consent, plagiarism, data fabrication, misconduct, falsification, double publication, redundancy, submission, and other related matters.

## Funding

This research did not receive any grant from funding agencies in the public, commercial, or nonprofit sectors.

## References

[1] A. Aliyari Boroujeni, R. Pourgholi, and S. H. Tabasi, A new improved teaching-learning-based optimization (ITLBO) algorithm for solving nonlinear inverse partial differential equation problems, Computational and Applied Mathematics, 42(2), 99, (2023).

[2] A. A. Boroujeni, R. Pourgholi, and S. H. Tabasi, Solving inverse partial differential equations by using Teaching-Learning-Based Optimization algorithm, TWMS Journal of Applied and Engineering Mathematics, 14(2), 740–753, (2024).

[3] A. A. Boroujeni and A. Khadivar, Solving the traveling salesman problem using a modified teaching-learning based optimization algorithm, International Journal of Industrial Engineering, 36(2), 170–184, (2025).

[4] A. Aliyari Boroujeni, M. R. Ghaemi, and R. Pourgholi, Solving the transportation problem using meta-heuristic algorithms, Analytical and Numerical Solutions for Nonlinear Equations, 9(1), 12–19, (2025).

[5] A. Guleria, B. Mehdinejadiani, and S. Chakma, Application of teaching-learning-based optimization algorithm for appraising solute transport parameters of mobile-immobile model, Groundwater for Sustainable Development, Article ID 101452, (2025).

[6] L. Kok Foong and B. Nguyen Le, Teaching-learning-based optimization (TLBO) hybridized with fuzzy inference system estimating heating loads, Energies, 15(21), 8289, (2022).

[7] J. D. Murray, Mathematical Biology I: An Introduction, 3rd ed. Springer, 2002.

[8] Y. T. Kao and E. Zahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, Applied Soft Computing, 8(2), 849–857, (2008).

[9] R. A. Fisher, The wave of advance of advantageous genes, Annals of Eugenics, 7(4), 355–369, (1937).

[10] N. A. Kudryashov, On exact solutions of families of Fisher equations, Theoretical and Mathematical Physics, 94(2), 211–218, (1993).

[11] R. V. Rao, V. J. Savsani, and D. P. Vakharia, Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems, Computer-Aided Design, 43, 303–315, (2011).

[12] G. D. Smith, Numerical Solution of Partial Differential Equations. Oxford University Press, New York, 1965.

[13] A. A. Boroujeni, R. Pourgholi, and S. H. Tabasi, Numerical solutions of KdV and mKdV equations: using sequence and multi-core parallelization implementation, Journal of Computational and Applied Mathematics, 454, 116184, (2025).

[14] R. Pourgholi, H. Dana, and S. H. Tabasi, Solving an inverse heat conduction problem using genetic algorithm: sequential and multi-core parallelization approach, Applied Mathematical Modelling, 38(7–8), 1948–1958, (2014).

[15] H. D. Mazraeh, K. Parand, M. Hosseinzadeh, et al., An improved water strider algorithm for solving the inverse Burgers–Huxley equation, Scientific Reports, 14, 28797, (2024).

[16] H. D. Mazraeh and K. Parand, GEPINN: An innovative hybrid method for a symbolic solution to the Lane–Emden type equation based on grammatical evolution and physics-informed neural networks, Astronomy and Computing, 48, 100846, (2024).

[17] H. D. Mazraeh, K. Parand, H. Farahani, and S. R. Kheradpisheh, An improved imperialist competitive algorithm for solving an inverse form of the Huxley equation, Iranian Journal of Numerical Analysis and Optimization, 14(3), (2024).

[18] H. Dana Mazraeh, P. Motaharinezhad, N. Daneshian, and K. Parand, Solving the Fokker–Planck equation with neural networks: a performance improvement approach, Analytical and Numerical Solutions for Nonlinear Equations, 9(1), 1–11, (2025).