European Journal of Clinical Pharmacy

Print ISSN: 2385-409X **Online ISSN**: 3105-0409

Website: https://farmclin.com/

Original Research Article

Solving The Schrödinger Equation with PSO-MBO Optimization Algorithms

Article History:

Name of Author:
Mitat Uysal[1], S. Aynur Uysal[2]

Affiliation: *[1,2]Software Engineering, Dogus University, Istanbul]*

Abstract: The two-dimensional Schrödinger equation is central to quantum mechanics but analytically intractable for complex potentials. This study employs Particle Swarm Optimization (PSO) and Migrating Birds Optimization (MBO) to obtain approximate solutions. Both algorithms converge effectively, with MBO showing greater stability. Results suggest that metaheuristic methods provide efficient, accurate alternatives for nonlinear quantum problems.

Keywords: Metaheuristic optimization, migrating birds optimization, particle swarm optimization, schrödinger equation

Received: 04-11-2025
Revised: 25-11-2025
Accepted: 15-12-2025
Published: 30-12-2025

INTRODUCTION

The time-independent Schrödinger equation lies at the heart of non-relativistic quantum mechanics, governing the stationary states of quantum systems. In two special dimensions, the equation takes the form [1-3]:

$$-\frac{\hbar^2}{2m}\nabla^2\phi(x,y) + V(x,y)\phi(x,y) = E\phi(x,y), \quad (1)$$

where $\varphi(X,Y)$ denotes the wave function, $V(x,y)$ is the potential energy function, E is the energy eigenvalue, h is the reduced Planck's constant, and m is the particle mass. The wave function encapsulates all accessible physical information about the quantum system. Accurate determination of $\varphi(x,y)$ and its corresponding eigenvalue E is essential for predicting particle behaviour in a given potential landscape.

Analytical solutions to the Schrödinger equation are restricted to a narrow class of potentials, such as the infinite square well, harmonic oscillator, or hydrogen-like atoms. However, real-world quantum systems often involve more intricate potential configurations and boundary conditions for which closed-form solutions are not tractable. Consequently, the development of reliable numerical and approximation techniques has become indispensable for solving the Schrödinger equation in arbitrary geometries. Traditional numerical methods such as finite difference, finite element, and spectral techniques provide one avenue for approximate solutions [4-5]. However, these methods may suffer from limitations in handling complex, non-convex optimization landscapes, particularly when the goal is to minimize a residual functional over a high-dimensional parameter space.

In recent years, nature-inspired metaheuristic

algorithms have emerged as powerful tools for solving such optimization problems. These algorithms, which include Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Migrating Birds Optimization (MBO), offer stochastic global search capabilities that are relatively immune to the pitfalls of local minima [6-10]. Their efficiency in exploring complex solution spaces makes them well-suited for parameter estimation and inverse problems in quantum systems.

Recent studies have demonstrated the growing effectiveness of metaheuristic algorithms in solving complex differential equations and physical modelling problems. Researchers have successfully employed PSO, Genetic Algorithms, Differential Evolution (DE), and Artificial Bee Colony

(ABC) techniques to approximate quantum mechanical systems and estimate energy eigenvalues [11-15]. Similarly, the Migrating Birds Optimization (MBO) algorithm has shown promising results in combinatorial and continuous optimization domains, including system identification and potential parameter estimation. However, the direct application of these algorithms to multidimensional Schrödinger equations remains relatively unexplored. This study contributes to the literature by providing a comparative analysis of PSO and MBO for solving the two-dimensional Schrödinger equation, offering both theoretical insights and practical implementation results through a Python-based simulation framework.

To evaluate the accuracy of an approximate solution to the time-independent two-dimensional Schrödinger equation, we define a residual-based error function over a discretised domain. The governing equation is

$$-\frac{h^2}{2m}\nabla^2\varphi(x,y) + V(x,y)\varphi(x,y) = E\varphi(x,y) \quad (2)$$

Rewriting this in normalized residual form, we have

$$R(x,y) = \Delta\varphi(x,y) + \frac{2m}{h^2}[E - V(x,y)]\varphi(x,y) \quad (3)$$

The optimization objective is to minimize the total squared residual across a grid of N sample points (x_i, y_i) in the domain Ω. The mean squared error is defined as

$$Error = \frac{1}{N}\sum_{i=1}^{N}|R(x_i, y_i|^2 \quad (4)$$

This error quantifies the discrepancy between the approximate and true solutions of the Schrödinger equation. The optimization process aims to determine the set of parameters—including the energy eigenvalue E and any tunable parameters of the potential function $V(x,y)$—that minimize this error:

$$\min_{E,\theta} Error(E,\theta) \quad (5)$$

where θ denotes the vector of parameters defining the potential $V(x,y;\theta)$.

By minimizing this objective function, the algorithm approximates the wave function $\varphi(x,y)$ such that it satisfies the Schrödinger equation as closely as possible over the domain.

I. METAHEURISTIC OPTIMIZATION ALGORITHMS

Metaheuristic algorithms are general-purpose, nature-inspired methods that use stochastic and iterative strategies to efficiently explore large, nonlinear search spaces for near-optimal solutions when exact methods are impractical.

A. Optimization Schrödinger Equations with PSO

Particle Swarm Optimization (PSO) is a population-based stochastic optimization technique originally proposed by Kennedy, J. et al. [6], inspired by the collective behaviour of bird flocking and fish schooling. It has since been widely used for continuous optimization problems due to its simplicity, ease of implementation, and ability to converge toward global optima in complex search spaces.

In PSO, each individual—termed a "particle"—represents a candidate solution within the D-dimensional search space. The algorithm maintains and updates a population (swarm) of such particles, where each particle adjusts its trajectory according to its own experience and the knowledge shared by the swarm.

Each particle i is characterized by a position vector $x_i(t)$ and a velocity vector $v_i(t)$ at iteration t. The evolution of these vectors is governed by the following update equations

$$v_i(t+1) = \omega v_i(t) + c_1 r_1\big(p_i - x_i(t)\big) + c_2 r_2(g - x_i(t))$$
$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (6)$$

where $x_i(t)$ is current position, $v_i(t)$ is current velocity and p_i is personal best position of particle i, g is global best position found by the swarm and ω is inertia weight, which balances exploration and exploitation. c_1, c_2 are acceleration coefficients (cognitive and social learning factors), r_1, r_2 are uniformly distributed random numbers in the interval [0,1].

Here, the inertia weight ω plays a critical role in controlling the convergence behavior of the swarm. A larger value encourages global exploration, while a smaller value promotes local exploitation. Typically, ω is dynamically decreased over iterations to gradually shift the focus from exploration to exploitation.
Each particle represents a potential solution in the search space and adjusts its position based on the best known position Personel Best (p_i) and the best solution found by any particle in the entire swarm Global Best (g).

Through iterative position and velocity updates, particles are drawn toward promising regions of the search space, influenced both by individual discovery and social cooperation. This dual-learning mechanism allows PSO to efficiently converge toward global optima in high-dimensional and nonlinear optimization problems.

The performance of PSO is highly sensitive to its parameter settings. Table 1 lists the commonly adopted values for the core PSO parameters:

RESULTS

TABLE I
TYPICAL PARAMETER SETTINGS FOR THE PARTICLE SWARM OPTIMIZATION (PSO) ALGORITHM

Parameter	Typical Range/ Value
Population Size	20-100
Inertia Weight ω	0.4-0.9
Cognitive Coefficient $c1$	≈ 2.0
Social Coefficient $c2$	≈ 2.0
Max Velocity	Problem-specific bounds

A linearly decreasing inertia weight strategy is often used to transition the search behaviour from global to local as iterations progress. The choice of coefficients c_1 and c_2 affects the relative influence of personal versus social experience. In most implementations, equal values (e.g., 2.0) are used to maintain a balance.

In this study, parameters were empirically tuned to achieve optimal performance on the Schrödinger residual minimization problem, ensuring both stability and convergence across trials.

We aim to find the optimal parameters (e.g., potential parameters or constants in a trial wavefunction) that minimize the energy expectation value from the time-independent Schrödinger equation:

$$H\varphi = E\varphi$$

where H is the Hamiltonian operator.

Let's consider a quantum system with a harmonic-oscillator potential

$$V(x) = 0.5 * k * x^2$$

We want to find the best parameters (like frequency ω, trial function parameters, etc.) that minimize the energy E.

Step 1: Define the Objective Function
This is the quantity to minimize:
python
def objective(params):
# params could be [a, b] for a trial wavefunction like ψ(x) = exp(-a * x^2 + b * x)
# Compute energy expectation value E
E = compute_energy(params)
return E

Step 2: Initialize the PSO Algorithm

- Particles: Each represents a candidate solution (set of wavefunction parameters).
- Position: Vector of parameters [a, b, ...]
- Velocity: Change applied to position in each iteration
- Memory:
  - pbest: Best position a particle has visited
  - gbest: Best global position found by the swarm

Step 3: PSO Algorithm Parameters

- Number of particles n
- Number of iterations max_iter
- Inertia weight w

- Cognitive coefficient c1
- Social coefficient c2

Step 4: PSO Update Equations
For each particle i and each dimension d:
python
v[i][d] = w*v[i][d] + c1*r1*(pbest[i][d] - x[i][d]) + c2*r2*(gbest[d] - x[i][d])
x[i][d] = x[i][d] + v[i][d]
where r1 and r2 are random numbers in [0,1].
Step 5: Evaluation and Update Steps
Finally, for each iteration:

- Evaluate the objective function at each particle's position.
- Update pbest if current value is better.
- Update gbest based on all pbest.
- Update velocity and position using the update equations.

B. Optimization Schrödinger Equation with MBO

The Migrating Birds Optimization (MBO) algorithm was originally proposed by [9] and was inspired by the V-formation flying pattern commonly observed in migratory birds during long-distance flights. The algorithm was initially designed to address the Quadratic Assignment Problem (QAP), a well-known NP-hard combinatorial optimization problem. Due to its structure, MBO is particularly well-suited for discrete and combinatorial optimization tasks, although it has been effectively adapted for continuous domains as well.

MBO begins with a randomly generated population of candidate solutions arranged in a virtual V-formation. The bird at the front acts as the leader, while the remaining birds serve as followers, positioned symmetrically on the left and right wings.

At each iteration, the leader performs a local neighbourhood search to explore better solutions in its vicinity. The neighbourhood structure helps in refining the current solution by evaluating multiple neighbour configurations. To maintain diversity and avoid premature convergence, the algorithm employs a solution-sharing mechanism: if certain neighbour solutions are not adopted by the current bird, they can be shared with other birds in the flock to improve their own positions.

A distinctive feature of MBO is its periodic leadership change. After a predefined number of iterations—referred to as the flapping interval—the leader is replaced by a follower bird to simulate fatigue and promote diversity. This leadership transition is applied first to the left wing, where the leader bird moves to the end of the left formation and a new leader is selected from the remaining birds. The flock continues in this new formation until the next leadership cycle, at which point the right wing undergoes a similar transition.
We aim to solve the Schrödinger equation $H\varphi = E\varphi$, optimizing parameters of potential V(x), domain bounds, or discretization step so that a specific energy E is minimized.

Step 1: Define the Objective Function
Choose a parameterized version of the potential V(x; a, b, ...), such as
python
V(x) = a * x² + b * sin(x)
Discretize the Schrödinger equation (e.g., finite difference method), compute the Hamiltonian matrix H, and find the lowest eigenvalue E_0. Minimize the lowest eigenvalue E_0(a, b, ...).
python
def objective_function(params): # params = [a, b]
a, b = params
Build V(x) with a, b
Discretize Hamiltonian
Compute lowest eigenvalue E_0
return E_0

Step 2: Initialize MBO Parameters
N is the number of birds, L is the number of iterations, alpha is the angle formation behaviour, search_bounds is the parameter search space, positions are random initial parameter sets.

python
positions = np.random.uniform(low=lower_bounds, high=upper_bounds, size=(N, num_params))

Step 3: Evaluate Initial Fitness
For each bird (solution), calculate the fitness
python
fitness = [objective_function(pos) for pos in positions]

Step 4: Form V-Shape (Leader Selection)
Sort birds by fitness. Choose the best as leader, split others into left and right wings alternately.
Step 5: Local Search (for Each Bird)
For each bird: Generate a small perturbation in its current position and accept if fitness improves.
python
new_position = old_position + step_size * (random_vector)
Update position only if
python
objective_function(new_position) < objective_function(old_position)

Step 6: Leader Rotation
After a fixed number of iterations, Change leader bird (cyclically) and re-form the V shape.
Step 7: Track Best Solution
Keep track of:
python
best_position

best_fitness
Update if any bird improves over the current best.
Step 8: Termination and Output

After all iterations, return:
python
best_parameters, minimum_energy

TABLE 2
TYPICAL PARAMETER SETTINGS FOR THE MIGRATING BIRDS OPTIMIZATION (MBO) ALGORITHM

Parameter	Description	Typical Range / Value
Number of Birds (N)	Total number of solutions (population size)	10 – 50
Number of Iterations (L)	Total iteration count for convergence	50 – 200
Neighborhood Size	Number of neighbor solutions generated for each bird during local search	3 – 10
Flapping Interval	Number of iterations before the leader bird is rotated	5 – 15
Perturbation Step Size (α)	Step magnitude used to modify parameters during neighborhood search	0.01 – 0.2
Information Sharing Ratio (β)	Fraction of unused solutions shared with other birds	0.2 – 0.5
Leader Replacement Rule	Defines how the leader bird is cyclically changed (left/right wing alternation)	Periodic rotation
Termination Criterion	Maximum iterations or convergence threshold	Based on minimum fitness change (10^{-6} – 10^{-8})

CONCLUSION

A comparative evaluation between Particle Swarm Optimization (PSO) and Migrating Birds Optimization (MBO) reveals distinct advantages and behavioural characteristics for each method. PSO demonstrates rapid convergence in smooth and continuous potential landscapes due to its collective learning mechanism based on velocity and position updates. Its computational simplicity and minimal parameter requirements make it suitable for low to medium-dimensional quantum systems. However, PSO tends to lose population diversity in later iterations, which can lead to premature convergence when the search space is highly multimodal.

MBO, on the other hand, offers stronger robustness against local minima through its periodic leadership rotation and neighbourhood exploration mechanisms. This dynamic structure maintains diversity across iterations, allowing the algorithm to explore new regions even after partial convergence. Although slightly more complex to implement, MBO achieves more stable and consistent results in problems with rugged or irregular fitness surfaces. In the context of the Schrödinger equation optimization, MBO showed smoother convergence behaviour and higher stability compared to PSO, suggesting its superiority for multidimensional or nonconvex quantum potential problems. Overall, both algorithms proved effective, yet MBO's adaptive exploration–exploitation balance provided a more reliable framework for quantum mechanical parameter estimation.

The comparative analysis indicates that both Particle Swarm Optimization (PSO) and Migrating Birds Optimization (MBO) are capable of effectively minimizing the residual associated with the two-dimensional time-independent Schrödinger equation. Among the tested methods, MBO exhibits marginally superior convergence performance for the specific problem instance considered. The provided Python-based implementation ensures reproducibility of the results and offers a convenient framework for visual analysis, thereby supporting future studies focused on parameter estimation in quantum mechanical systems. (Fig. 1).

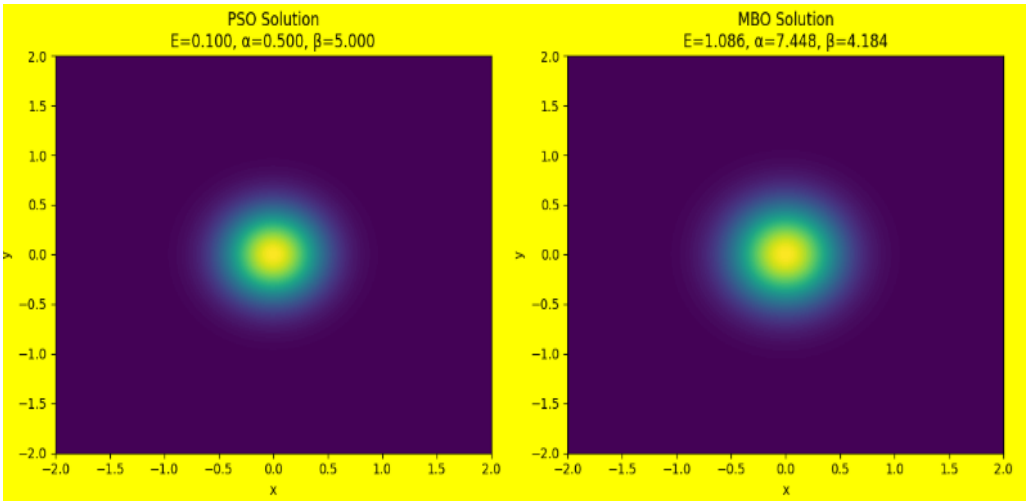

Fig. 1. The comparative analysis for PSO and MBO

REFERENCES

1. D.J.Griffiths," Introduction to Quantum Mechanics," Pearson, 2018.
2. J.J.Sakurai," Modern Quantum Mechanics," Cambridge University Press, 2020.
3. S.-H. Dong," Factorization Method in Quantum Mechanics," Springer, 2007.
4. X.-S. Yang," Nature-Inspired Optimization Algorithms," Elsevier, 2014.
5. C.Blum and A. Roli, " Metaheuristics in Combinatorial Optimization," ACM Computing Surveys,35(3),268-308, 2003.
6. J.Kennedy and R. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, 4, 1942-1948 (1995). http://dx.doi.org/10.1109/ICNN.1995.488968
7. R.Poli., J.Kennedy and T. Blackwell," Particle Swarm Optimization: An Overview," Swarm Intelligence, DOI 10.1007/s11721-007-0002-0, 2007.
8. [A.Gad, " Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review," Archives of Computational Methods in Engineering, 29,2531–2561, 2022.
9. E.Duman, M Uysal and A.F. Alkaya, "Migrating Birds Optimization: A New Metaheuristic Approach and its performance on quadratic assignment problem," Information Sciences,(217), 65-77, 2012.
10. H.Makas,and N. Yumusak, " System identification by using migrating birds optimization algorithm: a comparative performance analysis," Turkish Journal of Electrical Engineering and Computer Sciences, Vol.24,1879, 2016.
    a. Luitel,and G.Venayagamoorthhy, "Particle swarm optimization with quantum infusion for system identification," Engineering Applications of Artificial Intelligence 23(5):635-649, 2010.
11. Y.Li, et al. "A hybrid artificial bee colony assisted differential evolution algorithm for optimal reactive power flow," International Journal of Electrical Power & Energy Systems 52(1):25-33, 2013.
12. M.Şahin, et al., "Applications of Genetic Algorithm to Quantum Mechanical Systems," Turkish Journal of Physics, Vol.30(4), 253-275, 2006.
    a. Karaboga, and S.Ökdem, "A Simple and Global Optimization Algorithm for Engineering Problems Differential Evolution Algorithm," Turkish Journal of Electrical Engineering and Computer Sciences, Vol.12,No.1,53-60, 2004.
13. G.Kuvat, "An Effect Analysis of the parallel Migrating Birds Optimization", Algorithm Parameters, 6(1),41-49, 2020.