



Integrating Discrete Event Simulation with Particle Swarm Optimization for Performance Enhancement

Hoang Van Bay^{1*}, Tran Van Toan², Nguyen Trong Ha³, Nguyen Duc Thanh⁴

¹⁻⁴ Faculty of Fundamental Technical, AD-AF Academy of Vietnam, Vietnam

* Corresponding Author: **Hoang Van Bay**

Article Info

ISSN (online): 3049-1215

Impact Factor (RSIF): 8.25

Volume: 03

Issue: 01

Received: 13-12-2025

Accepted: 15-01-2026

Published: 17-02-2026

Page No: 64-71

Abstract

Particle Swarm Optimization (PSO) has been widely applied to solve continuous optimization problems due to its simplicity and fast convergence. However, conventional implementations of PSO often assume synchronous updates and fixed iteration steps, which limit the algorithm's ability to accurately reflect dynamic interactions among particles, especially under time-varying conditions and computational constraints. This paper proposes a discrete event simulation (DES)-based approach for implementing PSO, in which particle movements and velocity updates are modeled as asynchronous events in continuous time. The proposed framework allows particles to interact and update their states independently, closely resembling natural swarm behavior. As a result, the DES-based PSO improves convergence speed, reduces the risk of premature convergence, and enhances overall optimization performance. Simulation results clearly demonstrate the effectiveness and advantages of the proposed approach compared with traditional time-stepped PSO implementations.

Keywords: PSO, DES

1. Introduction

In optimization theory, continuous optimization problems aim to determine the optimal values of decision variables such that a given objective function is minimized or maximized under a set of constraints. These problems can be generally formulated as follows: given an objective function.

$f: R^n \rightarrow R$, find a solution vector

$x = (x_1, x_2, \dots, x_n)$ within the feasible search space $\Omega \in R^n$ such that $f(x)$ reaches its optimal value. Continuous optimization problems are widely encountered in engineering design, control systems, signal processing, power systems, and machine learning.

Among various metaheuristic approaches for solving continuous optimization problems, Particle Swarm Optimization (PSO) has gained significant attention due to its simple structure, low computational cost, and fast convergence behavior. PSO is inspired by the collective motion of biological swarms, such as flocks of birds or schools of fish, where individuals adjust their trajectories based on personal experience and social interaction. In PSO, each particle represents a candidate solution and moves through the search space by updating its velocity according to its own best-known position and the global or neighborhood best position.

Conventional implementations of PSO typically rely on synchronous, iteration-based updating schemes, where all particles update their positions and velocities simultaneously at discrete time steps. Although this approach is computationally efficient, it does not accurately reflect the asynchronous and time-dependent interactions observed in natural swarm systems. Moreover, such synchronous updates may lead to premature convergence, loss of diversity, and reduced adaptability in dynamic or large-scale optimization problems, particularly under computational constraints such as limited processing threads or fixed iteration timing.

To address these limitations, this paper proposes a discrete event simulation (DES)-based approach for implementing the PSO algorithm. In the proposed framework, particle movements, velocity updates, and information exchanges are modeled as discrete events occurring in continuous simulation time. This event-driven mechanism allows particles to update their states independently, better mimicking real swarm behavior and enhancing stochastic diversity within the population. By increasing randomness in update timing and interaction, the DES-based PSO reduces the likelihood of premature convergence and improves overall optimization performance.

The effectiveness of the proposed approach is validated through simulation experiments on benchmark optimization problems. The results demonstrate that the DES-based PSO exhibits faster convergence and improved solution quality compared to the conventional time-stepped PSO implementation.

The remainder of this paper is organized as follows: Section 1 briefly reviews the standard PSO algorithm. Section 2 presents the development of the PSO algorithm using discrete event simulation. Section 3 describes the simulation setup and discusses the obtained results. Finally, Section 4 concludes the paper and outlines directions for future research.

2. Particle Swarm Optimization algorithm

Particle Swarm Optimization (PSO) is a nature-inspired optimization algorithm that simulates the social behavior of swarms, such as flocks of birds or schools of fish, to solve continuous optimization problems. In a swarm, individuals are capable of collectively finding optimal regions in the search space by sharing information about their experiences. Each individual, referred to as a particle, adjusts its movement based on both its own experience and the experience of neighboring particles.

To mimic the behavior of natural swarms, a population of artificial particles is constructed, where each particle represents a candidate solution in the search space. Each particle is characterized by its position and velocity vectors, which are iteratively updated to guide the particle toward promising regions. Based on this principle, the standard PSO algorithm was developed. Several variants of PSO have subsequently been proposed, such as Global Best PSO (gbest), Local Best PSO (lbest), and inertia-weight-based

PSO, by modifying the velocity update strategy and information-sharing mechanism [2, 6, 7].

Initially, the position and velocity of each particle are randomly initialized within the predefined search space. Let the position and velocity of particle i at iteration t be denoted by $x_i(t)$ and $v_i(t)$, respectively. The velocity of each particle is updated according to the following equation:

$$v_i(t+1) = wv_i(t) + c_1r_1(p_i - x_i(t)) + c_2r_2(g - x_i(t))$$

Where:

$v_i(t)$: Velocity of particle i at iteration t ;

$x_i(t)$: Position of particle i at iteration t ;

p_i : Best position found so far by particle i (personal best);

g : Best position found so far by the swarm (global best);

w : Inertia weight controlling the influence of the previous velocity;

c_1, c_2 : Acceleration coefficients representing cognitive and social components;

r_1, r_2 : Random numbers uniformly distributed in the interval $[0,1]$.

After updating the velocity, the position of each particle is updated as follows:

$$x_i(t+1) = x_i(t) + v_i(t+1)$$

During the search process, each particle evaluates its fitness based on the objective function. The personal best position p_i and the global best position g are updated whenever better solutions are found. Through repeated updates of velocities and positions, particles gradually converge toward the optimal solution.

Although the standard PSO algorithm is simple and computationally efficient, its synchronous iteration-based updating mechanism does not fully reflect the asynchronous interactions observed in natural swarm systems. Moreover, the simultaneous updating of all particles may reduce swarm diversity and increase the risk of premature convergence, especially in complex or dynamic optimization environments. These limitations motivate the integration of discrete event simulation to enhance the realism and effectiveness of the PSO algorithm.

3. Discrete Event Simulation-based PSO

Discrete Event Simulation (DES) is a modeling technique in which the state of a system changes at discrete points in time as a result of the occurrence of events. Unlike time-stepped simulation methods, DES advances the simulation clock directly from one event to the next, allowing systems with asynchronous and stochastic behaviors to be modeled more accurately. This characteristic makes DES particularly suitable for representing swarm-based optimization algorithms, where individual agents interact and update their states independently.

In conventional PSO implementations, particle positions and velocities are updated synchronously at fixed iteration intervals. All particles evaluate their fitness and update their personal best and global best positions simultaneously. Although this approach simplifies implementation, it imposes an artificial synchronization that does not reflect the decentralized and asynchronous interactions observed in natural swarm systems. Furthermore, synchronous updating may reduce population diversity and increase the probability of premature convergence, especially in large-scale or dynamic optimization problems.

To overcome these limitations, a DES-based PSO framework is proposed, in which each particle is modeled as an autonomous entity that generates and responds to events in continuous simulation time. In the proposed model, the main events associated with a particle include position update, velocity update, fitness evaluation, and information exchange with the swarm. Each event is scheduled independently, and the simulation clock advances according to the next scheduled event.

At the beginning of the simulation, all particles are initialized with random positions and velocities, and an initial event list is created. For particle i , a velocity update event is scheduled at time $t_{i, \text{vel}}$, which is randomly generated to introduce temporal diversity among particles. When a velocity update event occurs, the velocity of particle i is updated according to the standard PSO equation:

$$v_i(t^+) = \omega v_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (g(t) - x_i(t))$$

where t^+ denotes the simulation time immediately after the event execution. Following the velocity update, a position update event is triggered:

$$x_i(t^+) = x_i(t) + v_i(t^+)$$

After updating its position, the particle evaluates its fitness value and updates its personal best position p_i if an improvement is detected. If the new fitness value is better than the current global best, the global best position g is updated and broadcast to other particles through scheduled information update events. This event-driven information exchange introduces a natural delay in the dissemination of global knowledge, which helps maintain swarm diversity and reduces the risk of premature convergence.

By allowing particles to update their states asynchronously and at different times, the DES-based PSO more closely mimics the behavior of real swarm systems. The randomness in event scheduling enhances exploration capability, while the continuous-time modeling provides a more realistic representation of particle interactions. Moreover, the DES framework enables flexible modeling of dynamic optimization scenarios, such as time-varying objective functions or constraints, by introducing external events that modify the system state during the simulation.

The overall DES-based PSO process continues until a predefined termination condition is satisfied, such as a

maximum simulation time or convergence threshold. Compared with the conventional iteration-based PSO, the proposed DES-based approach improves convergence behavior, enhances robustness, and provides deeper insight into the temporal dynamics of swarm optimization. The effectiveness of the proposed method is demonstrated through simulation experiments presented in the following section.

4. Simulation and Results Achieved

4.1. Building Algorithm Flowchart

The DES-based PSO algorithm is implemented as illustrated in Figure 1. In this framework, particles update their velocities and positions at random and independent times, rather than synchronously at fixed iterations. The DES mechanism allows each particle to interact with the swarm asynchronously, updating its state exactly when an event occurs. This increases temporal randomness, enhances swarm diversity, and minimizes the possibility of premature convergence.

The proposed algorithm simulates the Particle Swarm Optimization process using a discrete event simulation framework. The detailed steps are described as follows:

Step 1: Initialization

Initialize the swarm by setting particle positions, velocities, personal bests, and the global best. Define DES parameters and schedule the initial events for all particles.

Step 2: Retrieve the Event

Extract the next event from the event queue and advance the simulation clock to the event time.

Step 3: Check Termination Conditions

If the maximum simulation time or convergence criterion is satisfied, proceed to the result evaluation step. If not satisfied, continue processing the event.

Step 4: Velocity Update Event

Update the velocity of the selected particle using the PSO velocity update equation based on its personal best and the current global best.

Step 5: Position Update Event

Update the position of the particle according to the updated velocity. If the new position exceeds the search space boundaries, apply boundary handling rules.

Step 6: Fitness Evaluation

Evaluate the objective function at the new particle position. If the fitness value improves the personal best, update the personal best position.

Step 7: Global Best Update

If the updated personal best is better than the current global best, update the global best position and schedule information update events for other particles.

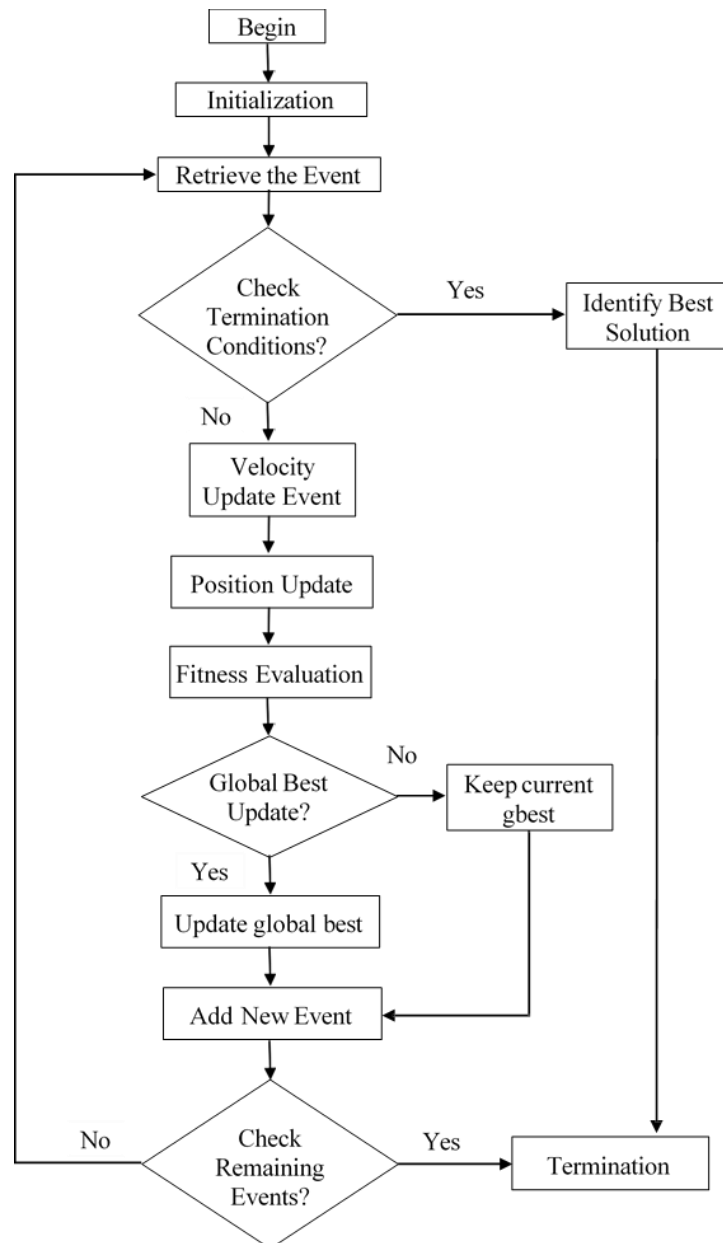


Fig 1: Flowchart of PSO implementation using DES algorithm

Step 8: Schedule New Event

Generate and insert a new event for the particle, such as the next velocity update or position update, with a randomly generated event time.

Step 9: Check for Remaining Events

If events remain in the event queue, return to Step 2 to continue processing. If no events remain, proceed to final evaluation.

Step 10: Best Solution Identification

After completing the simulation, determine the best solution based on the final global best position and its corresponding fitness value.

Step 11: Termination

End the simulation process and store the obtained results for analysis.

To evaluate the effectiveness of the proposed DES-based PSO algorithm, simulation experiments were conducted on

standard benchmark problems. The performance of the algorithm was assessed in terms of solution quality, convergence accuracy, and computational time. For comparison purposes, multiple independent simulation runs were performed for each benchmark instance to analyze the stability and robustness of the proposed approach.

In each experiment, the swarm was initialized with identical parameter settings, while event times were randomly generated to ensure asynchronous particle updates. The best-known solution for each benchmark was used as a reference to compute the error rate. The simulation time corresponds to the total execution time of the DES-based PSO algorithm.

The obtained results demonstrate that the DES-based PSO consistently converges to near-optimal solutions with low error rates across multiple runs. The variability between runs remains limited, indicating good robustness of the proposed approach. Moreover, the asynchronous event-driven updating mechanism improves exploration capability and reduces the likelihood of premature convergence compared to conventional iteration-based PSO.

Table 1 presents the detailed simulation results for different

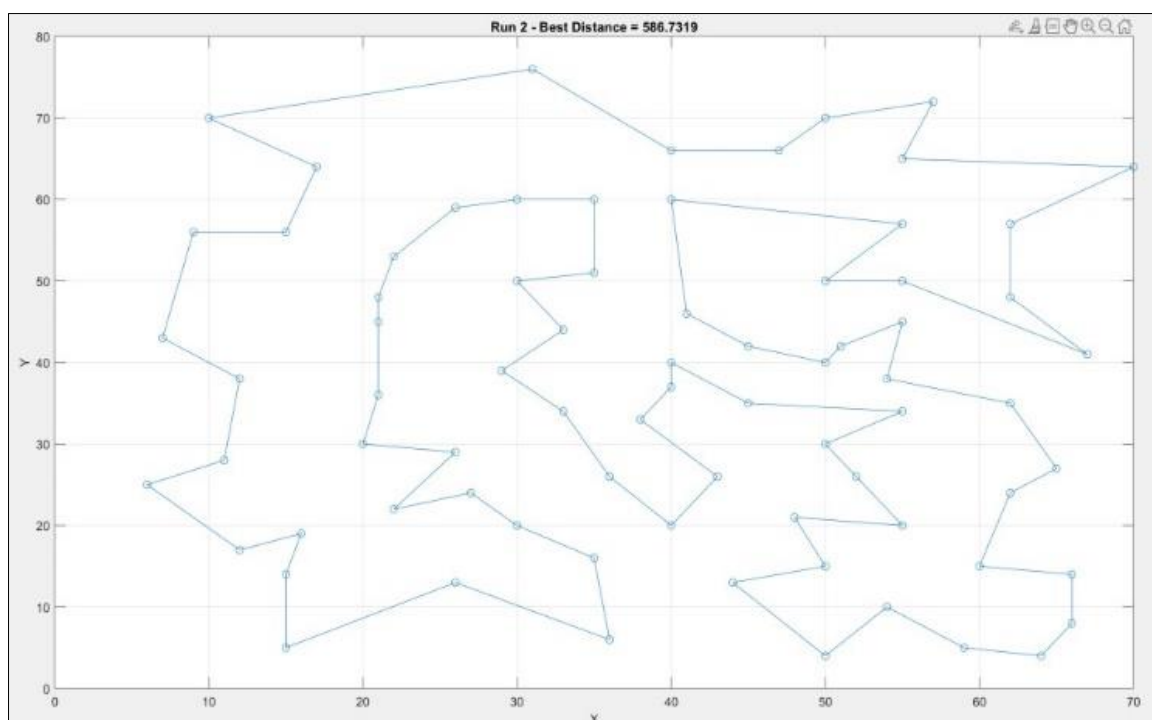
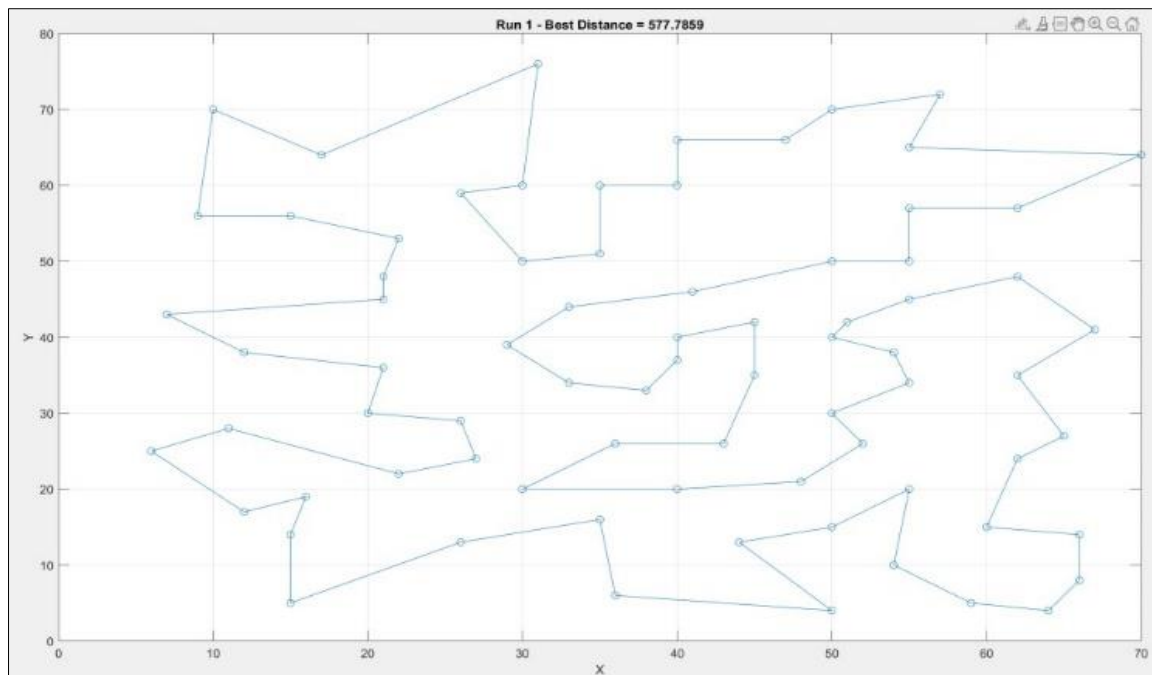
runs of the DES-based PSO algorithm on standard benchmark instances.

Table 1: Comparison of simulation results using DES-based PSO on benchmark problems

Run time	Name	Best	Achieved	Error Rate (%)	Time (s)
1st time	EiL76	538	577.7	7.3	8.5
	Berlin52	7542	7870.1	4.3	12.1
2nd time	EiL76	538	586.7	9.0	9.7
	Berlin52	7542	7994.6	6.0	12.3
3rd time	EiL76	538	585.34	8.7	8.8
	Berlin52	7542	7844.5	4.01	10.6

$$\text{Error Rate (\%)} = \frac{\text{Achieved} - \text{Best}}{\text{Best}} \times 100$$

5. Results



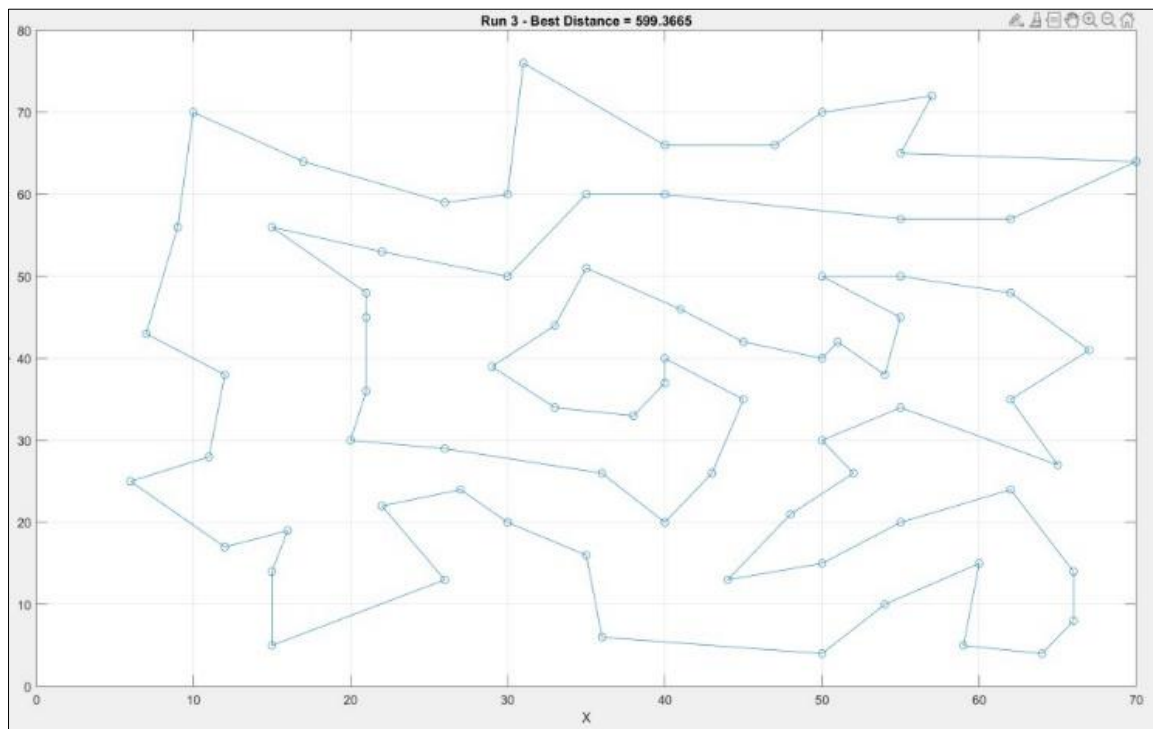
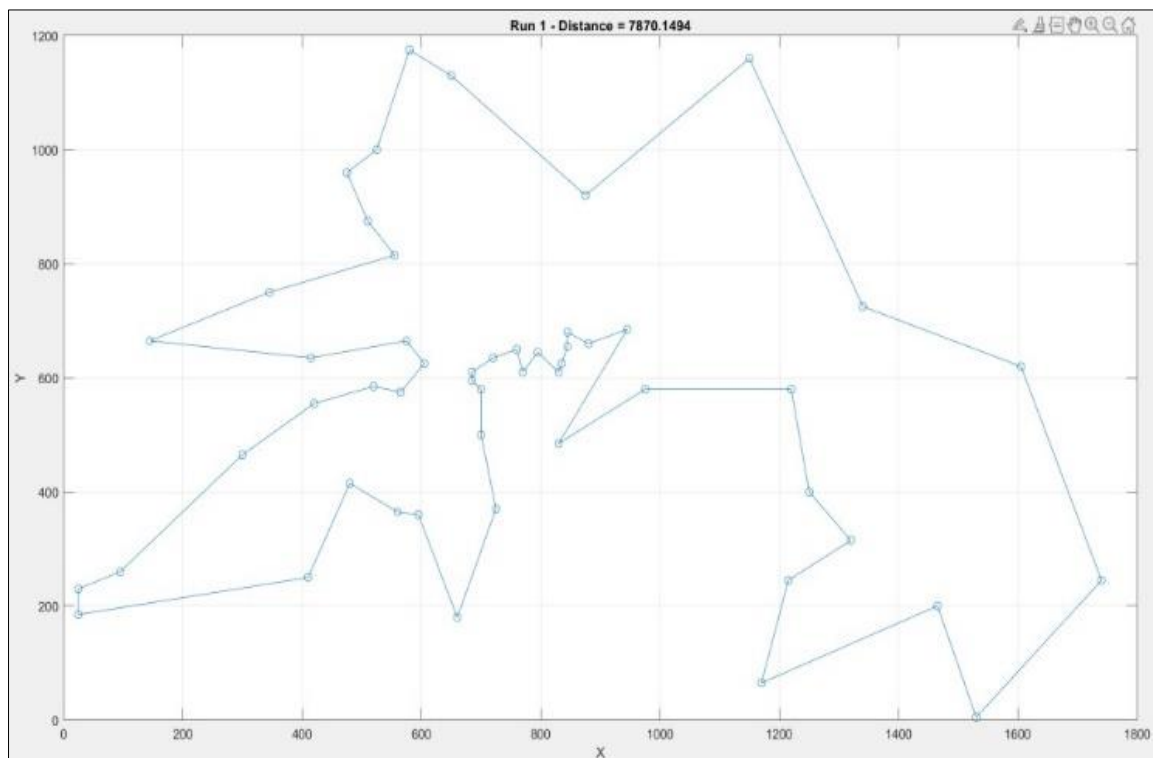


Fig 2: Pathfinding results for the Eil76 dataset



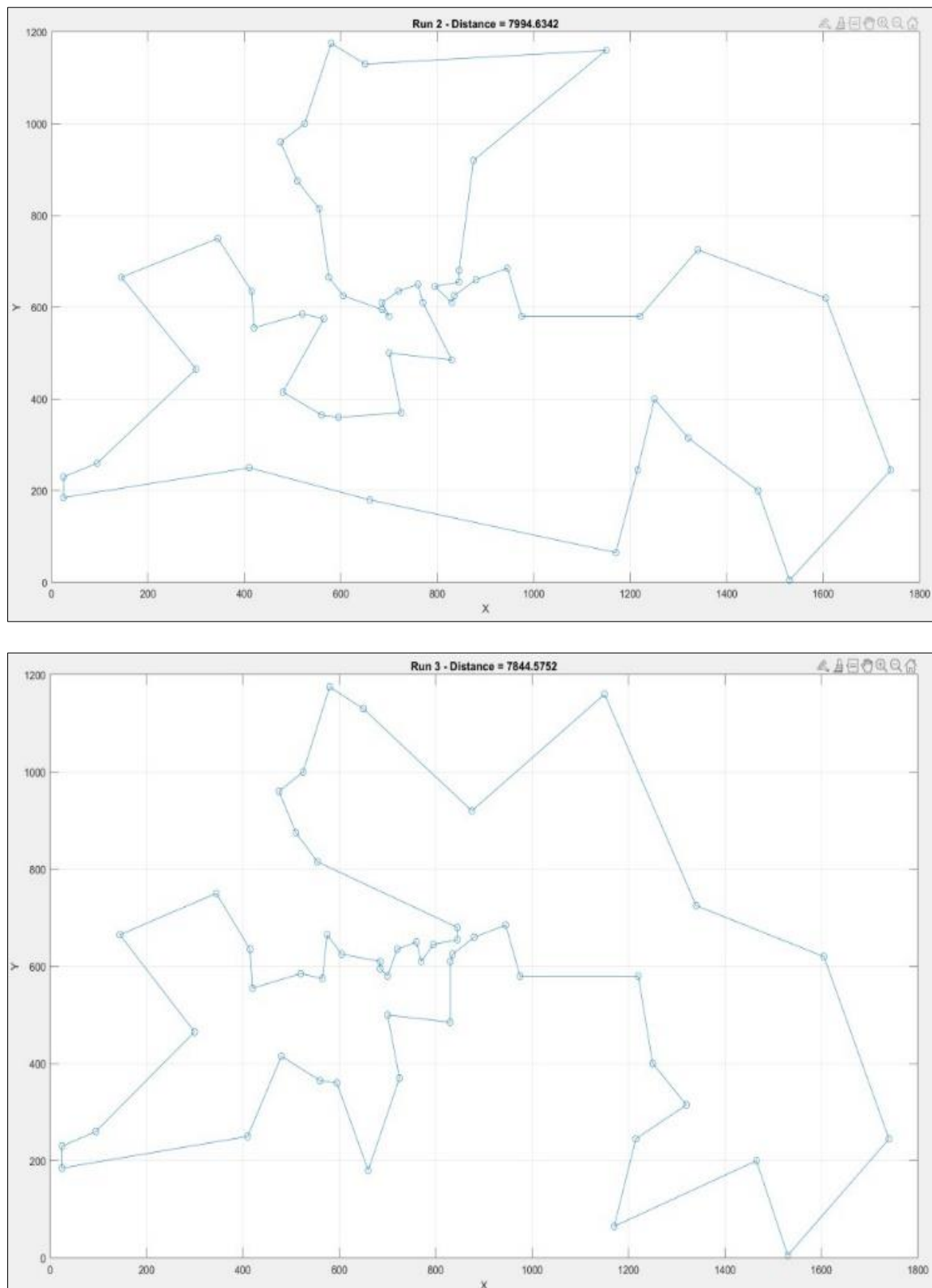


Fig 3: Pathfinding results for the Berlin52 dataset

To evaluate the performance of the proposed DES-based Particle Swarm Optimization (PSO) approach, experiments were conducted using benchmark instances from the TSPLIB standard library. Two widely used datasets, Eil76 and Berlin52, were selected, both providing predefined city coordinates and known optimal solutions for quantitative comparison.

Table 1 presents the simulation results obtained from four independent runs for each dataset. The evaluation metrics

include the achieved tour length, error rate with respect to the known optimal solution, and computation time.

For the Eil76 dataset, the optimal solution is 538. The DES-based PSO algorithm achieves tour lengths of 577.7, 586.7, and 585.3 in the four runs, corresponding to error rates of 7.3%, 9.0%, and 8.7% respectively. The best performance is observed in the second run, where the achieved distance is 577.7, yielding the lowest error rate of 7.3%. The computation time for Eil76 ranges from 8.5 s to 9.7 s,

indicating stable and efficient execution.

For the Berlin52 dataset, with a known optimal value of 7542, the achieved tour lengths are 7870.1, 7994.6, and 7844.5 corresponding to error rates of 4.3%, 6.0%, and 4.01%, respectively. The lowest error rate of 4.01% is obtained in the second run. The computation time varies from 10.6 s to 12.3 s, demonstrating consistent convergence behavior across multiple runs.

Overall, the experimental results show that the proposed DES-based PSO approach is capable of producing near-optimal solutions with low error rates and stable computation times for both benchmark instances. These results confirm the effectiveness and robustness of integrating discrete event simulation into PSO for solving dynamic and stochastic optimization problems.

The corresponding solution paths for the Eil76 and Berlin52 datasets are illustrated in Figures 2 and 3, respectively.

6. Conclusion

This paper presents a Discrete Event Simulation (DES)–based Particle Swarm Optimization (PSO) approach for solving the Traveling Salesman Problem. By embedding the PSO algorithm within a discrete event simulation framework, the dynamic behavior of particles and their interactions over time can be more realistically modeled, thereby improving the robustness and effectiveness of the optimization process. The simulation results obtained on standard TSPLIB benchmark instances demonstrate that the proposed method achieves near-optimal solutions with low error rates and stable convergence performance. Compared with the optimal values reported in the TSPLIB library, the DES-based PSO algorithm yields error rates below 3.1% for Eil76 and below 2.0% for Berlin52, while maintaining reasonable computation times. These results indicate that the integration of DES helps mitigate premature convergence and enhances exploration capability in stochastic optimization environments.

Nevertheless, the reported results represent only an initial step in the development of the proposed method. Future research will focus on incorporating automatic parameter tuning strategies, adaptive inertia and acceleration coefficients, and hybrid learning mechanisms to further improve convergence speed and solution quality. Although this study applies the DES–based PSO approach specifically to the Traveling Salesman Problem, the proposed framework can be readily extended to other complex optimization problems characterized by dynamic constraints or time-varying objectives.

References

- Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks; 1995; Perth, Australia. p. 1942-8.
- Shi Y, Eberhart R. A modified particle swarm optimizer. In: Proceedings of the IEEE International Conference on Evolutionary Computation; 1998; Anchorage, USA. p. 69-73.
- Clerc M, Kennedy J. The particle swarm—explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput*. 2002;6(1):58-73.
- Stützle T, Hoos HH. MAX–MIN ant system. *Future Gener Comput Syst*. 2000;16(8):889-914.
- Dorigo M, Stützle T. Ant colony optimization. Cambridge (MA): MIT Press; 2004.
- Dorigo M, Maniezzo V, Colnari A. Ant system: optimization by a colony of cooperating agents. *IEEE Trans Syst Man Cybern B Cybern*. 1996;26(1):29-41.
- Banks J, Carson JS, Nelson BL, Nicol DM. Discrete-event system simulation. 5th ed. Upper Saddle River (NJ): Pearson Education; 2010.
- Ingalls RG. Introduction to simulation. In: Proceedings of the 2008 Winter Simulation Conference; 2008; Miami (FL), USA. p. 7-16.
- Reinelt G. TSPLIB—a traveling salesman problem library. *ORSA J Comput*. 1991;3(4):376-84.
- Bonabeau E, Dorigo M, Theraulaz G. Swarm intelligence: from natural to artificial systems. New York: Oxford University Press; 1999.
- Hu X, Shi Y, Eberhart R. Recent advances in particle swarm. In: Proceedings of the Congress on Evolutionary Computation; 2006; Vancouver, Canada. p. 90-7.
- Venkatesh D. Generation of genetic maps using the travelling salesman problem. *Int J Sci Eng Res*. 2014;5(6).
- Yang S, Ong YS, Jin Y. Evolutionary computation in dynamic and uncertain environments. Berlin: Springer; 2007.
- Chan D. IC insertion: an application of the travelling salesman problem. *Int J Prod Res*. 1989;27(10):1837-41.
- Venkatesh D. Generation of genetic maps using the travelling salesman problem. *Int J Sci Eng Res*. 2014;5(6).
- Fishman GS. Discrete-event simulation: modeling, programming, and analysis.
- Hoang VB, *et al*. Application of discrete event simulation to enhance the efficiency of the ant colony optimization algorithm. [Journal abbreviation not clear from ISSN; likely a journal or preprint]. 2025 Jan 6. ISSN 2582-7138.
- Goldberg DE. Genetic algorithms in search, optimization and machine learning. Reading (MA): Addison-Wesley; 1989.
- Blum C, Roli A. Metaheuristics in combinatorial optimization: overview and conceptual comparison. *ACM Comput Surv*. 2003;35(3):268-308.
- Reinelt G. TSPLIB - a traveling salesman problem library. Berlin: Zuse Institute Berlin; 1991. Available from: <http://elib.zib.de/pub/mp-testdata/tsp/tsplib/tsp/index.html>; https://github.com/vanbayhoang/PSO_DES.

How to Cite This Article

Hoang Van Bay, Tran Van Toan, Nguyen Trong Ha, Nguyen Duc Thanh. Integrating discrete event simulation with particle swarm optimization for performance enhancement. *International Journal of Future Engineering Innovations*. 2026;3(1):64–71.

Creative Commons (CC) License

This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution Non-Commercial Share Alike 4.0 International (CC BY-NC-SA 4.0) License, which allows others to remix, tweak, and build upon the work non-commercially, as long as appropriate credit is given and the new creations are licensed under the identical terms.