

Performance Analysis of PSO-Based Path Planning Using Chaotic Systems with Different Chaoticity Levels

1 Bilal Gurevin¹, Ihsan Pehlivan², Parisa Mostafazadeh³, Emin Guney^{4*}, Trung
2 Thanh Nguyen⁵

3
4 ¹ Department of Robotics and Autonomous Systems Engineering, Faculty of Engineering, Fatih
5 Sultan Mehmet Vakif University, Istanbul, Turkey, bgurevin@fsm.edu.tr

6 ² Department of Electrical and Electronics Engineering, Faculty of Technology, Sakarya
7 University of Applied Sciences, Sakarya, 54100, Turkey, ipehlivan@subu.edu.tr

8 ³ Liverpool Logistics, Offshore and Marine (LOOM) Research Institute, Liverpool John Moores
9 University, Liverpool, L3 3AF, United Kingdom, P.Mostafazadeh@2023.ljmu.ac.uk

10 ⁴ Department of Computer Engineering, Faculty of Technology, Sakarya University of Applied
11 Sciences, Sakarya, 54100, Turkey, eminguney@subu.edu.tr

12 ⁵ Faculty of Engineering and Technology, Liverpool John Moores University, Liverpool, L3
13 3AF, United Kingdom, t.nguyen@ljmu.ac.uk

14

15 Corresponding Author: **Emin Güney**

16 Corresponding Author's Email id: **eminguney@subu.edu.tr**

Abstract

In this study, the performance of a particle swarm optimization (PSO)-based metaheuristic path planning algorithm integrated with different chaotic systems was evaluated. Unlike previous studies in the literature, this research specifically investigated the relationship between the chaoticity of chaotic systems and path planning performance. Lorenz, modified-Chameleon, and scaled-Zhongtang chaotic systems were integrated into the PSO algorithm by replacing the random parameter values required during each iteration with chaotic signals generated from these systems. The chaotic characteristics of the selected systems were analyzed using phase portraits, Lyapunov exponent spectra, and fast Fourier transform (FFT) analysis to determine their chaotic behavior levels. To evaluate the proposed approaches, seven different 20×20 m. test environments containing circular obstacles with varying positions and sizes were generated. Each algorithm was independently tested 30 times in every environment, resulting in a total of 840 experiments. The obtained results demonstrated that the scaled-Zhongtang-based PSO algorithm achieved the shortest path lengths and the best overall path planning performance in all environments, whereas the conventional PSO algorithm produced the lowest performance. Considering both the chaoticity analyses and the path planning results, it was observed that systems exhibiting stronger chaotic behavior provided greater improvements in PSO-based path planning performance.

Keywords: Chaos, Metaheuristic, Optimization, Path planning, ROS.

1. Introduction

With the technological developments, the use of robots has become widespread in all sectors. The energy consumption problem of mobile robots, which are constantly in motion compared to stationary robots, has become an important field of study in both academic and industrial fields. The search for optimum paths in complex environments is one of the most important problems in mobile robotics [1]. Effective path planning not only ensures efficient navigation, but also protects against collisions and optimizes energy consumption. Meta-heuristic algorithms have emerged as powerful tools for handling such complex optimization tasks [2]. These algorithms are inspired by natural events that often simulate behavior observed in biological, physical or social systems. Although they do not guarantee global optimality, their

ability to navigate wide solution spaces and converge to optimal solutions has positioned them at an important point in solving a variety of real-world problems.

In parallel, combining chaotic systems with optimization techniques has been the focus of attention of researchers. Chaotic systems, characterized by sensitivity to initial conditions and aperiodic behavior, exhibit complex and seemingly random trajectories in confined regions. Recent research has highlighted that taking advantage of the inherent chaotic properties of such systems can potentially add further exploration capabilities to optimization algorithms [3]. This has led to a growing interest in exploring how the chaoticity of systems can affect the performance of meta-heuristic algorithms [4].

While existing studies examine chaotic-based path planning approaches, they primarily focus on improving algorithmic performance using different chaotic maps. However, they do not explicitly and systematically analyze the relationship between the degree of chaoticity of chaotic systems and optimization performance. This research aims to shed light on the relationship between chaos and optimization in the field of meta-heuristic path planning by optimizing the PSO algorithm with systems exhibiting different levels of chaoticity. Specifically, this study investigates whether higher chaoticity leads to improved performance in terms of finding optimal or near-optimal paths for mobile robots.

The remainder of the article is as follows: Section 3 covers chaotic systems and their analysis, Section 4 focuses on integrating chaotic systems into PSO, Section 5 presents the experimental studies, and Section 6 evaluates the results.

2. Literature Review

The use of meta-heuristic algorithms has led to significant advances in the field of optimization, providing versatile tools to tackle complex problems that traditional path planning methods try to solve efficiently. Some examples in the literature where metaheuristic algorithms are used in the path planning problem are as follows. Pattnaik et al. utilized the hybrid particle swarm and chemical reaction optimization (HPCRO) algorithm in local planning, considering the criteria of optimal length and safety in path planning for a mobile robot. They performed path planning in an environment with unknown surroundings and objects. The results of the HPCRO algorithm were compared with genetic algorithms (GA), chemical reaction optimization, and particle

swarm optimization (PSO) algorithms. The hybrid algorithm provided successful results in terms of cost function value and computational cost [5]. Xin et al. proposed a technique to overcome the issues of slow convergence speed, limited local search capability, and early convergence that are associated with GA. They compared their method with five examples from the traveling salesman problem library. Additionally, they tested the algorithm on a physical environment using a marine vehicle [6]. Elhoseny et al. proposed a Bezier curve-based modified GA approach for path planning of mobile robots in dynamic environments. They demonstrated the calculation of the optimal distance between the start and end points in six different environments, achieving energy efficiency [7].

Lamini et al. developed a GA-based application that performs path planning by overcoming obstacles between two points in a static environment. In their study, they employed a different crossover operator to avoid premature convergence. As a result, they obtained a more efficient algorithm in terms of distance, safety, and energy. They compared their work with three existing studies in the literature [8]. Kang et al. conducted a study on the optimization of path planning for mobile robots using the enhanced Dijkstra algorithm with PSO. They compared the optimal path results obtained from Dijkstra, Enhanced Dijkstra, and Enhanced Dijkstra with PSO using the MAKLINK graph. The shortest path was achieved in the study where Enhanced Dijkstra was used in conjunction with PSO [9].

The inherent unpredictability and complex dynamics of chaotic systems have shown that combining these dynamics with optimization algorithms can increase their search capabilities, especially in complex and high-dimensional problem spaces. Furthermore, chaotic systems are utilized to improve the issues of early convergence and getting trapped in local optima in metaheuristic algorithms. In this context, the studies have been conducted in the literature: Lian et al. presented the chaotic adaptive PSO algorithm in their research article, designed for use in static environments. They considered the fitness function based on path length and obstacle avoidance. Chaotic maps were employed to optimize the random parameters of PSO. The most successful results were obtained with the Kent Map. Through their study, they observed improvements in global search capability and speed [10]. Nayar et al. conducted a study on the enhancement of PSO using a chaotic approach. They utilized the Logistic map as the chaotic system and performed their work in MATLAB. Their study demonstrated that the chaotic-based PSO found shorter paths with fewer iterations [11]. Yuan et al. conducted a study on the optimization of the reptile search algorithm (RSA). They focused on addressing issues such as

low population diversity, imbalanced exploration, and getting trapped in local optima. They developed a chaotic inverse learning strategy using chaotic maps to improve population diversity. They also proposed an alternative pool strategy to balance exploitation and exploration. A shifted distribution estimation strategy was used to modify all individuals and guide evolution. They validated the capabilities of MRSA using 23 benchmark functions, IEEE CEC2017 benchmark functions, and robot path planning problems [12].

Ait Saadi et al. worked on the optimization of unmanned aerial vehicle path planning. They employed a hybrid approach combining Chaotic Aquila Optimization and Simulated Annealing algorithms. They used the Singer chaotic map to determine random values in Aquila. They compared the performance of the proposed algorithm with three benchmark maps and nine metaheuristic algorithms. They demonstrated that the proposed algorithm outperformed others in terms of fitness and cost [13]. Cheng et al. focused on optimizing the early convergence, weakness in global search, and finding the optimal path in PSO. In their study, they combined PSO with Grey Wolf Optimization (GWO) to create the IPSO-GWO algorithm. They utilized the logistic map chaotic system to determine the velocity and positions of particles. Additionally, they employed random chaotic sequences to rescue particles from local optima. Simulation results showed that the proposed IPSO-GWO outperformed the conventional PSO-GWO in terms of performance [14].

Wu et al. presented the Sparrow Search Algorithm for the path planning problem of mobile robots. They increased the diversity in the initial population by using the Tent Map chaotic map. Through simulation experiments, they demonstrated that the Sparrow Search Algorithm with Tent Map produced more optimal paths [15]. Ntakolia et al. proposed a hierarchical approach that combines global and local search for path planning. They utilized the chaotic ant colony optimization algorithm with fuzzy logic for generating the global path and employed deep convolutional neural networks for obstacle detection and local navigation. In their study focused on wearable technologies, the proposed path planning algorithms demonstrated computational efficiency and capability for real-time applications [16]. Chu et al. introduced a cost function for safe flights of unmanned aerial vehicles in a 3D environment. They combined PSO with enhanced nonlinear dynamic inertia weights. The proposed algorithm improved the convergence and fitness function value of PSO. They further utilized the logistic map chaotic system to initialize the velocity data chaotically. Through simulation studies, they demonstrated that the

proposed algorithm reduced convergence speed in complex environments and generated smoother paths [17].

Vikas and Parhi conducted a study on optimizing the path planning of humanoid robots in complex terrain conditions. They combined chaos with the gravitational search algorithm (GSA), using linear regression-based RGSA. They employed chaotic systems such as the logistic map, Gauss map, piecewise linear chaotic map, and sinusoidal map. They tested their approach in real-time simulations and a laboratory environment. The study successfully enabled the humanoid robot to navigate dynamic and static obstacles and reach its goal [18]. Hosseinpourfard and Javidi utilized PSO and GA for path planning optimization. They generated the random parameters used in PSO and GA using chaotic systems such as the Lorenz, Tent map, and Henon map. In their results, they showed that CPSO using the Lorenz system outperformed PSO, GA, and CGA in generating more optimal paths [19].

Cheng and Liu improved Grey Wolf Optimization (GWO) algorithm, named CGGWO, was proposed to address path planning problems by integrating chaotic mapping, gold mining optimization strategies, vertical and horizontal crossover, and Gaussian mutation. In this approach, chaotic mapping is used to initialize the population, improving its distribution across the search space and enhancing diversity. Additional mechanisms, such as crossover strategies and mutation, are employed to prevent premature convergence and improve global search capability. Experimental results on benchmark functions and path planning scenarios demonstrated that the proposed CGGWO algorithm achieves faster convergence, higher stability, and shorter path lengths compared to conventional methods [20].

Ugwoke et al. presented a simulation-based review study that categorizes classical, heuristic, and metaheuristic path planning algorithms while analyzing their fundamental principles, characteristics, and application domains. The study provides a comparative evaluation of different algorithms in simulated environments with varying obstacle configurations. The results indicate that no single algorithm consistently outperforms others across all scenarios, as performance depends on factors such as environment complexity, obstacle distribution, and problem structure. Furthermore, the study highlights the trade-offs between path length, computational time, and planning efficiency [21].

Li et al. proposed an improved Harris Hawks Optimization (HHO) algorithm, referred to as MIHHO, to address path planning problems in mobile robots. In this approach, a Chebyshev chaotic mapping strategy is employed to initialize the population, thereby increasing diversity and enhancing global search performance. In addition, a fusion exploration mechanism combining the discovery strategy of the Sparrow Search Algorithm with the exploration phase of HHO is introduced to improve search efficiency. Furthermore, modifications such as adaptive control of the parameter E and a dynamic Lévy flight strategy are incorporated to balance exploration and exploitation and accelerate convergence. Simulation results demonstrate that the proposed MIHHO algorithm achieves improved search performance, higher planning efficiency, and better path quality compared to conventional methods [22].

Zhao et al. propose MYIGWO, an enhanced Grey Wolf Optimizer (GWO) designed to address limitations in global exploration, the exploitation–exploration balance, and static neighbor selection in the improved GWO (IGWO). The algorithm introduces a dual mutation strategy: Cauchy mutation is applied to lower-performing individuals to enhance global search, while polynomial mutation accelerates convergence and strengthens local exploitation. A directional correction mechanism further stabilizes mutation outcomes. Additionally, the DLH strategy's neighbor radius is dynamically adapted using chaotic mapping, allowing flexible neighbor selection. MYIGWO was extensively evaluated against eighteen advanced metaheuristic algorithms on multiple benchmark functions and applied to four classical engineering optimization problems as well as robot path planning. Results demonstrate that MYIGWO significantly improves global search capability, convergence speed, and robustness, consistently yielding near-optimal solutions and effectively overcoming several shortcomings of the original GWO [23].

Huang and Fu present a hybrid path planning algorithm for unmanned aerial vehicles (UAVs) that integrates the artificial potential field (APF) method with a chaotic bat algorithm (CBA). While APF alone can lead UAVs into local minima, the CBA introduces a metaheuristic global search mechanism inspired by bat foraging behavior, enhanced with chaos theory to improve exploration and exploitation. The algorithm generates initial populations using chaotic mapping, guides UAVs along the potential field gradient, and dynamically calibrates parameters to optimize path quality and search efficiency. Comparative simulations against standard bat

algorithm, DE-based hybrid bat algorithm, and other CBA variants demonstrate that the proposed fusion method achieves shorter paths, higher success rates, reduced computation time, and improved navigation safety, particularly in complex environments [24].

3. Motivation, Novelty and Significance

Although the use of chaotic systems in the optimization of meta-heuristic algorithms is an emerging research area, studies that explicitly focus on the interaction of chaotic systems and meta-heuristic algorithms for mobile robot path planning are relatively limited. The available literature provides a basis for the research presented in this work, but there are notable gaps in a comprehensive understanding of how systems of different chaoticity affect the performance of a metaheuristic algorithm in solving the path planning problem.

Existing studies examine chaotic-based path planning approaches, but they do not analyze chaoticity-performance relationship. In contrast to other existing studies mentioned in the literature, this particular research focused on investigating the impact of chaotic systems' chaotic properties on conventional PSO-based metaheuristic path planning algorithm. The question addressed is whether systems exhibiting more chaotic features would lead to better optimization of the metaheuristic algorithm, and why.

The chaotic systems considered in this study were selected to represent different levels of chaoticity. The Lorenz system is a classical benchmark widely used in the literature, whereas the modified-Chameleon and scaled-Zhongtang systems exhibit distinct dynamical structures and higher chaoticity characteristics.

Moreover, to the best of our knowledge, the modified-Chameleon and scaled-Zhongtang chaotic systems have not been previously utilized in PSO-based path planning problems. Therefore, this study not only investigates the effect of chaoticity levels on optimization performance but also contributes to the literature by evaluating the applicability of these less-explored chaotic systems in path planning. This selection enables a systematic investigation of how varying degrees of chaoticity influence the performance of PSO-based path planning.

Lorenz [25] chaotic system, which is the starting point of chaotic systems and known to have good chaotic performance, was used to develop the PSO for path planning. On the other hand, modified-Chameleon [26] and scaled-Zhongtang [27] chaotic systems, which are known not to be used in path planning optimization before, although they have high chaoticity, were used. The chaotic systems were discretized using the Runge-Kutta 4 (RK-4) numerical method to convert them into discrete-time systems. The random parameter values required to update the velocity data in each iteration of PSO replaced with float numbers ranging from 0 to 1 obtained from chaotic systems. As a result, 3 more algorithms were derived: Lorenz PSO (LPSO), modified-Chameleon PSO (CPSO), and scaled-Zhongtang PSO (ZPSO), in addition to the conventional PSO.

To test these four different algorithms, seven different environments were created. Each environment had obstacles of varying sizes and locations. Each algorithm was tested 30 times for each environment between the selected start and target points. This resulted in a total of 840 different tests that could be used for comparison. A comparison was made between the results obtained from the chaoticity analysis and the results obtained from the path planning tests. According to the path planning tests, ZPSO achieved the most successful results. Likewise, in the chaoticity analysis, the scaled-Zhongtang system was found to be the most chaotic compared to the other two chaotic systems. On the other hand, conventional PSO-based path planning tests gave the lowest performing results. According to all test results, the improvement effect of the systems with higher chaoticity properties on the PSO-based metaheuristic algorithm was clearly observed.

3.1. Chaotic Systems Used

In this section, the chaotic systems used in the study are introduced by providing their parameter and initial values. The equation for the Lorenz chaotic system is shown in Equation 1 [25]. Additionally, the parameter values for the Lorenz chaotic system were set as $a = 10$, $b = 49.16$, $c = 8/3$ and the initial conditions were $x_0 = 0.1$, $y_0 = -0.1$, $z_0 = 9$. The decimal numbers in the range of 0-1, generated using the RK-4 solution method, were utilized in the production of the PSO algorithm incorporating the Lorenz system.

$$\begin{aligned}\dot{x} &= a(y - x) \\ \dot{y} &= bx - y - xz \\ \dot{z} &= xy - cz\end{aligned}\tag{1}$$

273 The equation for the Modified-Chameleon chaotic system is shown in Equation 2 [26]. The
 274 mentioned chaotic system has parameter values of $a=1.9$, $b=1.1$, $c=11.1$, $d=0.7$ and initial
 275 conditions of $x_0=0.4$, $y_0=0.1$, $z_0=0$. The decimal numbers in the range of 0-1, generated using
 276 the RK-4 solution method, were used in the production of the PSO algorithm incorporating the
 277 Modified-Chameleon system.

$$\begin{aligned} \dot{x} &= ay \\ \dot{y} &= -x + byz \\ \dot{z} &= -x - cxy - dxz \end{aligned} \quad (2)$$

279 The equation for the scaled-Zhongtang chaotic system is shown in Equation 3 [27]. Additionally,
 280 the scaled-Zhongtang chaotic system has parameter values of $a=80$, $b=40$, $c=5$, $d=10$, $e=2$
 281 , $f=10$, $g=48.83$ and initial conditions of $x_0=1$, $y_0=0$, $z_0=1$. The decimal numbers in the
 282 range of 0-1, generated using the RK-4 solution method, were used in the production of the PSO
 283 algorithm incorporating the scaled-Zhongtang system.

$$\begin{aligned} \dot{x} &= ay - bx \\ \dot{y} &= cx + dy - exz^2 \\ \dot{z} &= -fx - gz + zx^2 \end{aligned} \quad (3)$$

285 3.2. PSO and Use of Chaotic Signals in PSO

286 The general logic of the PSO algorithm is based on each individual in the swarm trying to move
 287 closer to the best individual in the swarm [28]. The proximity of each individual to the solution
 288 is measured based on the defined objective function. Through the objective function, each
 289 individual is directed towards its own best solution and the global best solution. As a result, over
 290 time, each individual starts searching for a solution around the individual with the best state and
 291 gets closer to the solution. The best solution found so far by each individual during the search
 292 process is referred to as the local best (pbest) value. The value of the individual in the swarm that
 293 is closest to the solution is referred to as the global best value [29].

294
 295 The basic steps of the PSO algorithm can be summarized as follows:

- 296 • Random velocity and position values are assigned to each individual to create an initial
 297 swarm.
- 298 • The fitness values of each individual are calculated.
- 299 • In each iteration, each individual is compared to the previous iteration to select the best
 300 one. This value is assigned as the local best value for each individual.
- 301 • The individual with the best value in the current iteration is assigned as the global best

value.

- The process is repeated until the termination condition is met.

Based on Equation 4 and Equation 5, the velocity and position information of individuals are updated in each iteration.

$$V_{ij}^{t+1} = wV_{ij}^t + c_1r_1(P_{ij}^t - x_{ij}^t) + c_2r_2(G_{gbest}^t - x_{ij}^t) \quad (4)$$

$$x_{ij}^{t+1} = x_{ij}^t + V_{ij}^{t+1} \quad (5)$$

According to the PSO algorithm, random numbers in the range of 0 to 1, represented by the parameters r_1 and r_2 in Equation 4, are used in each iteration. In conventional implementations, these values are generated using pseudo-random number generators provided by the programming language.

In this study, instead of conventional pseudo-random numbers, chaotic sequences generated from deterministic chaotic systems were utilized. Specifically, normalized values obtained from the x-dimension of each chaotic system – Lorenz, modified-Chameleon, and scaled-Zhongtang – were used to replace the random parameters r_1 and r_2 in the PSO algorithm. These chaotic signals were obtained by discretizing the corresponding systems using the RK-4 numerical method and scaling the outputs into the range of 0 to 1. It should be noted that each chaotic system was used independently to construct a separate PSO variant. During a single optimization run, only one chaotic system was employed, and no switching between different chaotic systems was performed during iterations.

In Equation 6, X_1^t and X_2^t denote chaotic sequences generated from the selected chaotic system. These values are obtained from the x-dimension of the corresponding chaotic system and normalized into the range of [0, 1]. In contrast to conventional PSO, where r_1 and r_2 are pseudo-random numbers, the proposed approach employs deterministic chaotic signals to enhance the diversity and exploration capability of the algorithm.

$$V_{ij}^{t+1} = wV_{ij}^t + c_1X_1^t(P_{ij}^t - x_{ij}^t) + c_2X_2^t(G_{gbest}^t - x_{ij}^t) \quad (6)$$

Accordingly, three different chaos-based PSO variants were developed: LorenzPSO (LPSO), modified-ChameleonPSO (CPSO), and scaled-ZhongtangPSO (ZPSO). This design ensured that a fair and consistent comparison of the effects of different chaotic systems and their chaoticity levels on path planning performance was achieved. The overall working principle of the PSO algorithm is preserved, and the integration of chaotic signals into the PSO framework is

illustrated in detail in the flowchart presented in Figure 1. In addition to the flowchart presented in Figure 1, the implementation procedure of the proposed chaotic PSO framework is summarized in Algorithm 1 for improved methodological clarity.

According to the PSO algorithm, the random parameters r_1 and r_2 in Equation 4 are generated within the range of 0 to 1 at each iteration. In the conventional PSO algorithm, these values are produced using pseudo-random functions provided by the programming language. In this study, instead of pseudo-random generators, chaotic sequences obtained from chaotic systems were used to generate these parameters.

For each PSO variant, only one chaotic system was employed throughout the entire optimization process. Specifically, the Lorenz system was used in LPSO, the modified-Chameleon system in CPSO, and the scaled-Zhongtang system in ZPSO. The chaotic systems were discretized using the RK-4 numerical method, and the x-dimension signals obtained from these systems were normalized into the range of 0 to 1. During optimization, the r_1 and r_2 parameters required by PSO were selected from the chaotic number sequence generated by the corresponding chaotic system at each iteration. No dynamic switching between different chaotic systems was performed during the optimization process.

Algorithm 1: Proposed Chaotic PSO Framework

-
- 1: Initialize particle positions (x_{ij}^t) and velocities (V_{ij}^t)
 - 2: Initialize pbest and gbest values
 - 3: Select one chaotic system (Lorenz / Modified-Chameleon / Scaled-Zhongtang)
 - 4: Generate chaotic sequence using RK-4 method
 - 5: Normalize chaotic sequence into the range [0,1]
 - 6: for each iteration t do**
 - Select chaotic values X_1 and X_2 from the generated chaotic sequence
 - Update particle velocity
 - Update particle position
 - Evaluate fitness value
 - Update pbest, gbest
 - 7: end for**
 - 8: Return optimal path
-

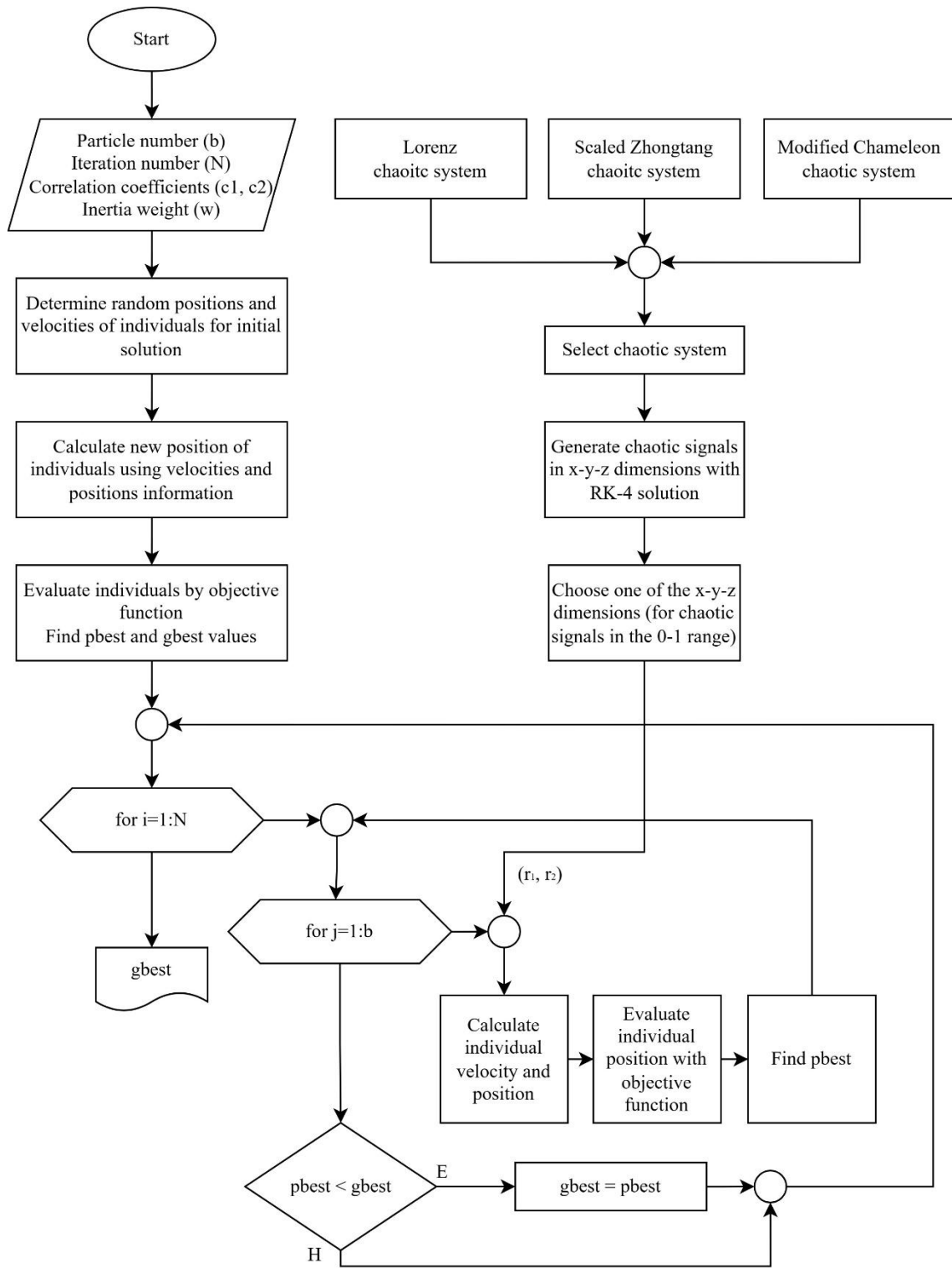


Fig. 1. Generating random numbers from chaotic systems and integrating them into the PSO

In the PSO algorithm employing chaotic sequences, the parameter settings were defined as follows: inertia weight (w) = 0.8, acceleration coefficients (c_1 and c_2) = 1.8, population size = 100, and maximum number of iterations = 100. The parameter settings used in all experiments are summarized in Table 1.

Parameter	Value
Inertia weight (w)	0.8
Acceleration coefficient (c1)	1.8
Acceleration coefficient (c2)	1.8
Population size	100
Maximum iteration number	100

Table 1. PSO parameter settings used in experiments

3.2.1 Objective Function and Constraint Handling

In the proposed PSO-based path planning approach, particles were evaluated using an objective (fitness) function designed to minimize the total path length while penalizing infeasible solutions such as obstacle collisions, environment violations, and invalid start/goal positions. Thus, the optimization process aimed to generate the shortest feasible and collision-free path between the defined start and target points. The total path length was calculated as follows:

$$L = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \quad (7)$$

where L represents the total path length, N denotes the number of path points, and (x_i, y_i) represents the coordinates of the i th path point. To ensure feasible path generation, penalty terms were incorporated into the fitness function for constraint violations. The final objective (fitness) function used during optimization is defined as follows:

$$F = L(1 + P_s)(1 + P_g)(1 + N_e\beta)(1 + N_c\gamma) \quad (8)$$

where F represents the fitness value, P_s and P_g denote the penalty terms associated with start-point and goal-point violations, respectively, N_e represents the number of environment violations, and N_c represents the number of collision violations. The coefficients β and γ correspond to the environment violation and collision violation penalty coefficients, respectively.

In this study, $P_s=P_g=1$ when the corresponding violation occurs; otherwise, the penalty value is zero. The environment violation penalty coefficient was selected as $\beta = 0.2$, while the collision violation penalty coefficient was selected as $\gamma = 1$. In addition, fixed penalty values of 1 were assigned for start-point and goal-point violations. During optimization, paths with higher

numbers of constraint violations produced larger fitness values and were therefore less likely to be retained as personal-best (pbest) or global-best (gbest) solutions. Since the optimization process directly minimizes the path cost value, no additional normalization procedure was applied. Consequently, lower fitness values correspond to shorter and more feasible collision-free paths.

4. Experiments

In this section, path planning experiments were conducted using both the conventional PSO and chaos-integrated PSO algorithms. Four different algorithms, namely PSO, LPSO, CPSO, and ZPSO, were comparatively evaluated. A total of seven different 2D test environments were generated, as presented in Fig. A1 (Appendix A). Each environment had dimensions of 20×20 m and included circular obstacles with different sizes and spatial distributions to create varying levels of navigation complexity. In the first five environments, obstacle positions and sizes were manually designed, whereas in environments 6 and 7, obstacle positions and diameters were randomly generated using Python code.

For all experiments, the start point and target point were fixed at coordinates (19,1) and (1,19), respectively, to ensure fair comparison among algorithms. The detailed parameter settings and experimental configurations used in all simulations are summarized in Table 2.

Parameter	Value
Environment size	20×20 m
Number of environments	7
Obstacle type	Circular static obstacles
Start point	(19,1)
Goal point	(1,19)
Population size	100
Number of iterations	100
Number of runs	30
Termination criterion	Maximum iteration count

Table 2. Experimental setup

Each algorithm was executed 30 times in every environment, resulting in a total of 840 independent experiments. The maximum number of iterations was set to 100, which was also

used as the termination criterion for the optimization process. As an example, Fig. A2 presents the first three and last three iterations obtained using the PSO algorithm in Environment-7, while Fig. A3 shows the iterations obtained using the ZPSO algorithm in the same environment. Figure 2 illustrates the final paths obtained after 100 iterations using PSO and ZPSO. The ZPSO-based path exhibits shorter distance and smoother turns compared to the conventional PSO result.

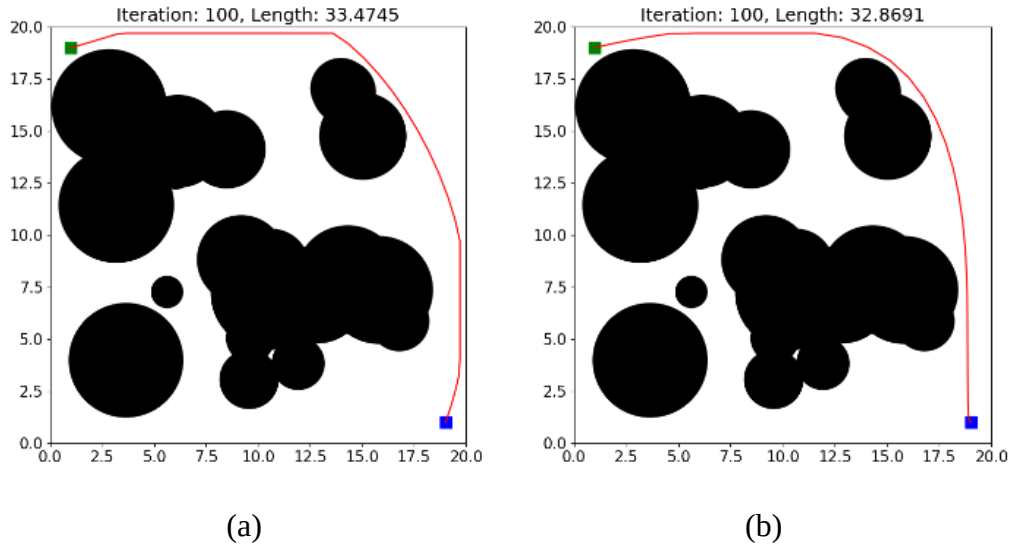


Fig. 2. Path results from PSO (a) and ZPSO (b) in environment-7

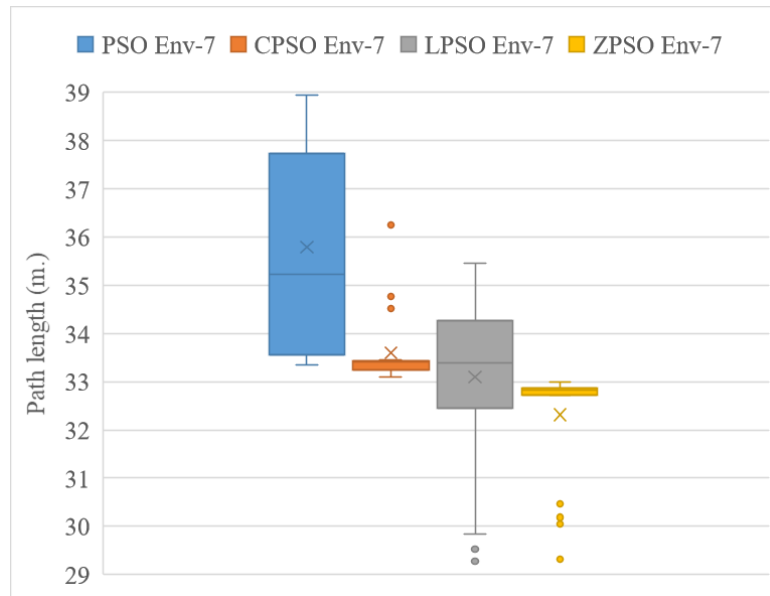


Fig. 3. Path length results for environment-7

The boxplot graphical of the path planning test results performed with 4 different path planning algorithms in Environment-7 is as given in Figure 3. Considering the average values and box sizes, it is seen that the ZPSO path planning algorithm calculates the shortest length paths and is

less dispersed. It is seen that the PSO path planning algorithm produces the longest paths on average and is more unstable than ZPSO. On the other hand, it is seen that the LPSO and CPSO path planning algorithms, respectively, come right after ZPSO in terms of performance. Likewise, the results of other test environments from environment-1 to environment-6 are given in Fig. A4 - Fig. A9, respectively. According to the test results, PSO containing the scaled Zhongtang system was the most successful algorithm in every environment, while conventional PSO without chaotic signals was the lowest performing algorithm in every environment.

The evaluation of the average results of the path lengths obtained from 30 different tests in each environment is represented with the graphs. The obtained results for environment-7 are shown in Figure 4. All other test results are as shown in Fig. A10 - Fig. A15. The tests conducted with ZPSO achieved the most successful results in each environment. Only in environment-6, CPSO performed better than LPSO. In all other environments, LPSO outperformed CPSO. PSO, on the other hand, resulted in the longest path calculations in all environment tests.

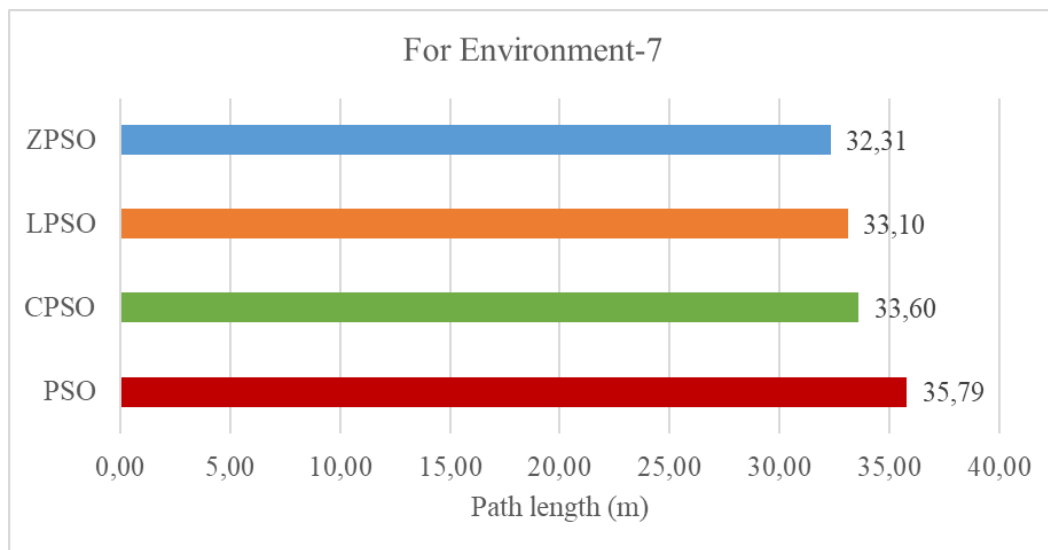


Fig. 4. Average path length results for environment-7

5. Algorithmic behavioral analysis

In this section, behavioral analysis of ZPSO, which includes scaled-Zhongtang chaotic signals and produces the shortest paths, and conventional PSO without chaotic signals, with convergence graph, standard deviation graph, T-test and path trajectory analysis were examined.

5.1. Convergence Graph Analysis

Since ZPSO achieved the best path planning performance in all test environments, while the conventional PSO algorithm produced the lowest performance, these two algorithms were

selected for detailed convergence analysis and statistical comparison. For each environment, both algorithms were independently executed 30 times, and each run consisted of 100 iterations. During the convergence analysis, the path length values obtained at each iteration were averaged over 30 independent runs to generate the convergence curves. Therefore, the convergence graphs presented in Figures 5 and 6 represent the iteration-versus-average path length behavior of the algorithms. Figure 5 compares the convergence characteristics of PSO and ZPSO in environment-3. It can be observed that ZPSO converges more rapidly and reaches a shorter path length with fewer iterations compared to the conventional PSO algorithm. Similarly, Figure 6 illustrates the convergence behavior of both algorithms in environment-7. The results demonstrate that ZPSO achieves faster convergence and more stable optimization performance.

In addition to the convergence analysis, statistical evaluations were performed using the final path lengths obtained at the 100th iteration of the 30 independent runs. The mean path length values were calculated by averaging the final path lengths obtained from these runs, while the standard deviation values were computed using the same final iteration results to evaluate the consistency and stability of the algorithms.

The statistical comparison results for all environments are summarized in Table 3. The obtained results indicate that ZPSO consistently achieved lower mean path lengths and significantly lower standard deviation values compared to the conventional PSO algorithm. Particularly in environment-7, PSO produced a result of 68.66 ± 48.70 , whereas ZPSO achieved 32.31 ± 1.10 . The substantially lower standard deviation obtained by ZPSO indicates a more stable and reliable optimization performance across independent runs.

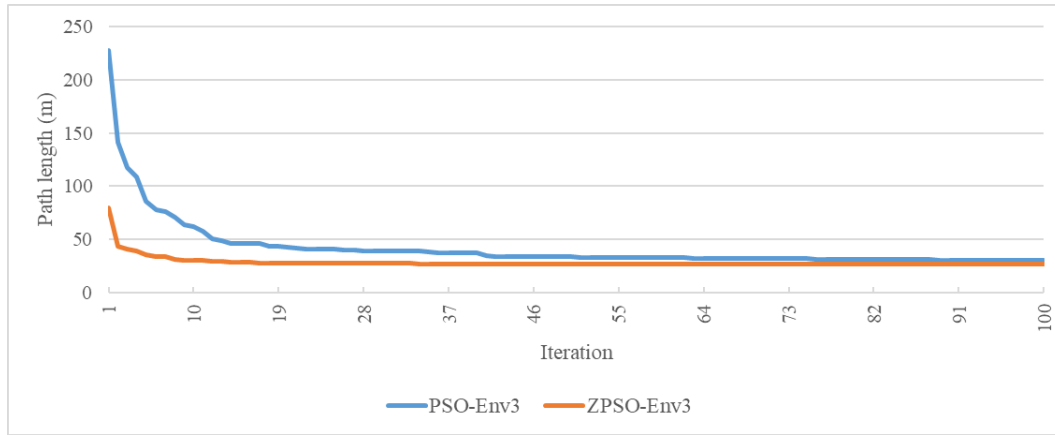


Fig. 5. Convergence comparison of conventional PSO and ZPSO in Environment-3 (20×20 m environment with circular static obstacles) using a population size of 100 and 100 iterations. The curves represent the average results obtained from 30 independent runs.

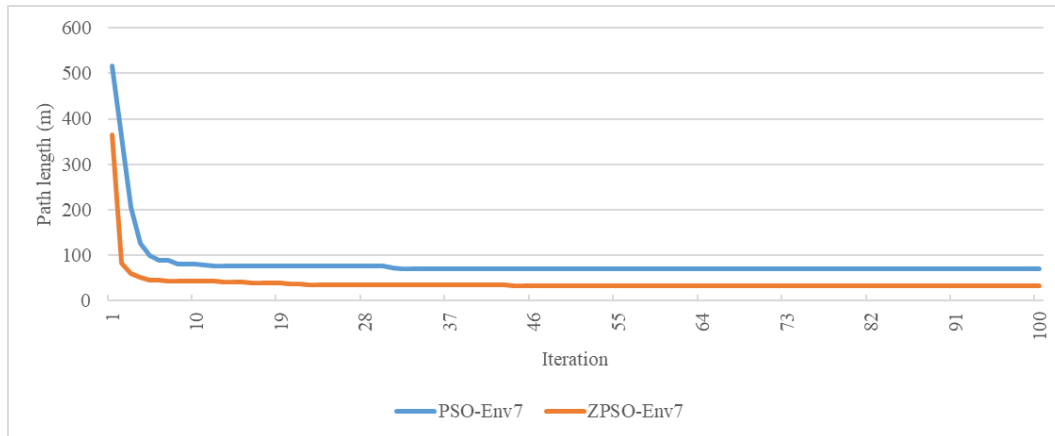


Fig. 6. Convergence comparison of conventional PSO and ZPSO in Environment-7 (20×20 m environment with circular static obstacles) using a population size of 100 and 100 iterations. The curves represent the average results obtained from 30 independent runs.

Environments	PSO	ZPSO
	Mean Path Length $\pm$ Std	Mean Path Length $\pm$ Std
Env-1	29,01 $\pm$ 1,57	27,22 $\pm$ 0,07
Env-2	29,80 $\pm$ 2,32	27,03 $\pm$ 0,47
Env-3	30,55 $\pm$ 4,13	26,61 $\pm$ 0,76
Env-4	32,44 $\pm$ 5,56	29,40 $\pm$ 0,15
Env-5	29,65 $\pm$ 1,67	26,60 $\pm$ 0,94
Env-6	36,72 $\pm$ 1,76	30,98 $\pm$ 0,43
Env-7	68,66 $\pm$ 48,70	32,31 $\pm$ 1,10

Table 3. Comparison of mean path length and standard deviation values obtained from 30 independent runs

5.2. Standard Deviation Analysis

The standard deviation graphs were plotted for 2 different algorithms Figure 7 and Figure 8: PSO and ZPSO. It is evident that the spread of data for ZPSO is smaller compared to PSO, indicating less variability in the performance of ZPSO. Furthermore, the data points in the ZPSO graph are tightly clustered around the mean, suggesting a relatively consistent and similar set of values. Conversely, the PSO graph shows a greater spread of data, indicating higher variability in the performance of the algorithm.

In summary, the difference between the means and other data points is small for the ZPSO algorithm, while it is high for the PSO algorithm. This discrepancy implies that the ZPSO algorithm exhibits a relatively consistent performance across different samples or data points, while the PSO algorithm demonstrates more variability. The small difference between the means and other data points for ZPSO suggests that its performance values tend to cluster closely around the mean, resulting in a consistent performance level across various samples or conditions. On the other hand, the PSO algorithm shows a higher degree of variability in its performance, with performance values exhibiting greater spread and variability around the mean. This indicates that the performance of PSO is less consistent across different samples or conditions. From a practical standpoint, these findings imply that the ZPSO algorithm is more likely to produce stable and predictable results, given its relatively consistent performance. In contrast, the PSO algorithm's higher variability suggests that its results may vary significantly across different samples or conditions.

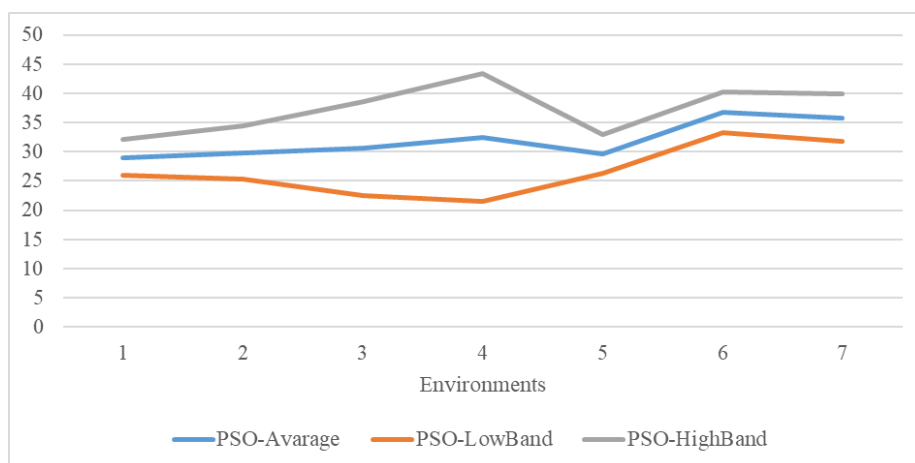


Fig. 7. Standard Deviation graph PSO for all environments

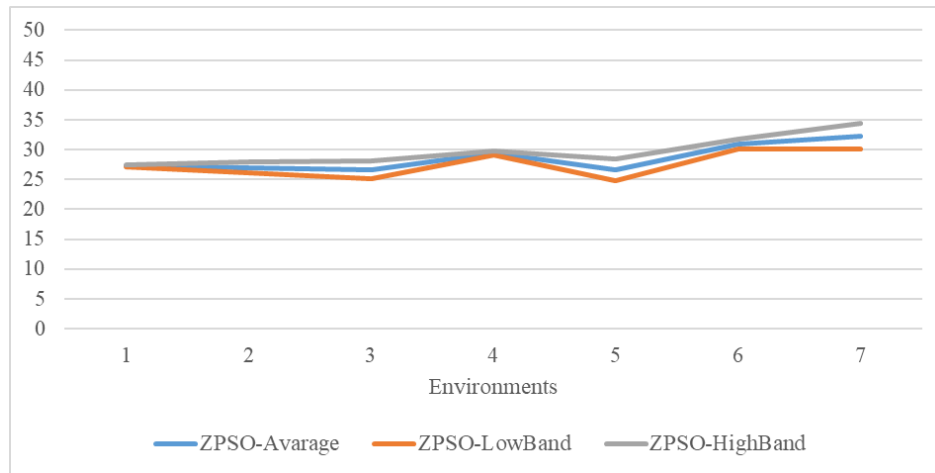


Fig. 8. Standard deviation graph ZPSO for all environments

5.3. Statistical T-test Analysis

To assess the performance disparity between the PSO and ZPSO algorithms across different environments (3 and 7), a two-tailed t-test was conducted using a sample size of 30. The null hypothesis assumed no significant difference between the groups. The resulting p-values for environments 3 and 7 were calculated as $8.79981E-07$ and $2.69118E-12$, respectively. These p-values provide substantial evidence to reject the null hypothesis, as they are significantly smaller than the conventional threshold of 0.05. Hence, it can be concluded that there is a significant difference in performance between the two algorithms across both environments.

The small p-values suggest a statistically significant difference between the two algorithms regarding their ability to calculate the shortest path. It implies that the observed disparity in performance is unlikely to be a result of random chance or sampling variability. Instead, it points towards a genuine and meaningful distinction between the algorithms in terms of their effectiveness. In practical terms, these findings imply that one algorithm outperforms the other significantly in calculating the shortest path.

5.4. Path Trajectory Analysis

In this section, the smoothness of the paths obtained using ZPSO and PSO path planning algorithms were compared. Smoothness metric values of the optimal paths obtained as a result of 20 different runs from ZPSO and PSO were calculated. Low values indicate that the path is smoother. The smoothness metric values obtained from environment-3 and environment-7 as a result of 20 runs are as given in Figure 9 and Figure 10, respectively. When the mentioned figures are examined, it can be seen that ZPSO produces smoother paths, especially in environment-3.

In addition, t-test values of the smoothness metric values obtained as a result of 20 runs were calculated for environment-3 and environment-7. The resulting p-values for environments 3 and 7 were calculated as 9.02E-03 and 5.87E-04, respectively. These p values, like the T-test analysis in the previous section, provide significant evidence to reject the null hypothesis because they are significantly smaller than the conventional threshold of 0.05. Therefore, it can be concluded that there is a significant difference in performance between the two algorithms in both environments.

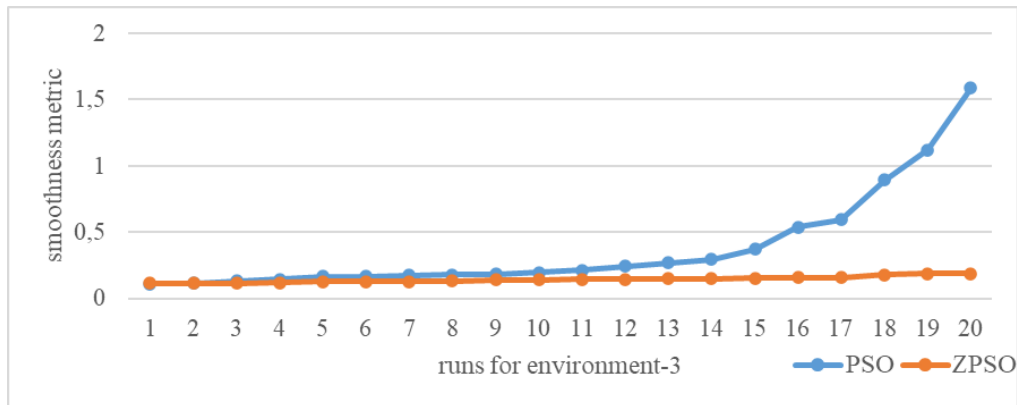


Fig. 9. Comparing of smoothness for ZPSO and PSO in environment-3

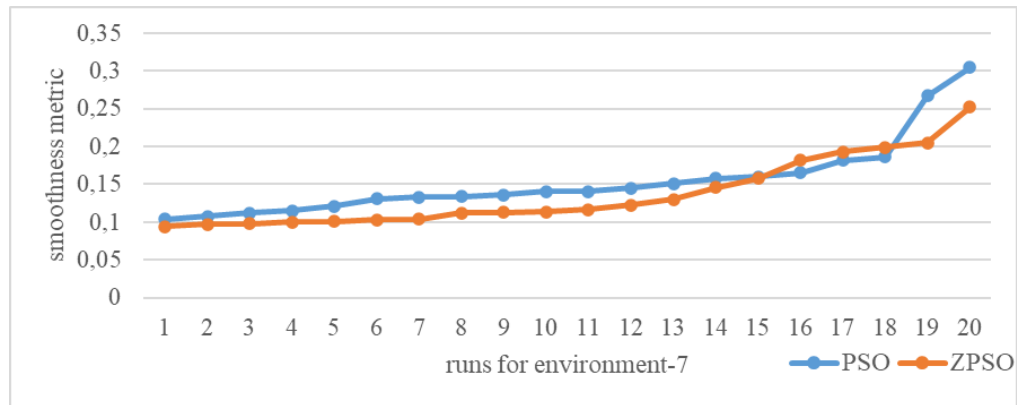


Fig. 10. Comparing of smoothness for ZPSO and PSO in environment-7

In summary, three points come to the fore in the comparison of PSO and ZPSO path planning algorithms. In general, ZPSO seems to have been able to smoothen the trajectory faster than PSO. E.g., ZPSO run 5 vs PSO run 3, which both started from similar trajectories (Fig. A16, Fig. A17). ZPSO is much better at getting out of a local optimum. Out of 10 runs, PSO got stuck forever in a local optimum that is infeasible very early in runs 2, 5, 6, 8 and never managed to escape (Fig.

A18, Fig. A19, Fig. A20, Fig. A21). It also got to the same local optimum in run 9 at iteration 2, but managed to escape in iteration 3 (Fig. A22). Out of 10 runs, ZPSO got to the same local optimum twice in runs 5 and 9, but in both times, it managed to escape in the next few iterations (Fig. A23, Fig. A24). ZPSO, however, is not immune to getting stuck at some local optima. The one that ZPSO ended up to in 9 out of 10 runs is not the absolutely best trajectory (see point 3 below).

The best trajectory is hard to find for both PSO and ZPSO, and was found only by ZPSO once (Fig. A25). The reason ZPSO could find this one and stick with it is likely because it was close to this trajectory right from the beginning (iteration 1, although as an infeasible trajectory at first). This trajectory however is hard to find, because its basin of attraction is very narrowed and covered by infeasible areas. Twice ZPSO started very near this trajectory (runs 2 and 6, Fig. A26, Fig. A27), but could not manage to find it. Instead ZPSO very quickly switched to the second best optimum (top right corner). Once ZPSO is in this trajectory, it doesn't get out.

6. Chaotic Analysis

In this section, analyses were made to measure why different chaotic systems affect the performance of the PSO path planning algorithm differently. In particular, to understand why ZPSO performs better, the chaotic behaviors of the systems were examined. There are several analyses that indicate which system exhibits more chaotic behavior among chaotic systems. The interpretation of whether one system possesses more chaotic characteristics than another can generally be made based on the following analyses [30].

- 3D systems tend to exhibit more chaotic behavior compared to 2D systems.
- When examining phase portraits, it is generally accepted that asymmetric systems possess more chaotic characteristics than symmetric systems.
- The first Lyapunov exponent must be positive. The higher this value, the more chaotic the system is.
- Additionally, a high Lyapunov dimension indicates a higher level of chaotic behavior.
- Systems with higher bandwidth tend to exhibit more chaotic characteristics.

In the upcoming sections, the chaoticity analyses of the selected chaotic systems will be conducted.

6.1. Phase Portrait Analysis

In this section, phase portraits obtained using the RK-4 solution method for the Lorenz, modified-

Chameleon, and scaled-Zhongtang chaotic systems used in PSO optimization are shown in Figure 11, Fig. A28, and Fig. A29, respectively. When examining the phase portraits of the Lorenz chaotic system, it can be observed that there are two phase portraits, XZ and YZ, which exhibit a symmetric appearance with respect to the y-axis. This indicates that the Lorenz system exhibits less chaotic behavior compared to asymmetric chaotic systems. When examining the phase portraits of the modified-Chameleon chaotic system, it can be observed that there are two phase portraits, XZ and YZ, which exhibit a symmetric appearance with respect to the y-axis. This indicates that the modified-Chameleon system exhibits less chaotic behavior compared to asymmetric chaotic systems.

On the other hand, when examining the phase portraits of the scaled-Zhongtang chaotic system, it can be seen that all phase portraits are asymmetric with respect to the y-axis. Based on the evaluation of phase portraits, it can be concluded that the scaled-Zhongtang system exhibits more chaotic characteristics compared to symmetric chaotic systems.

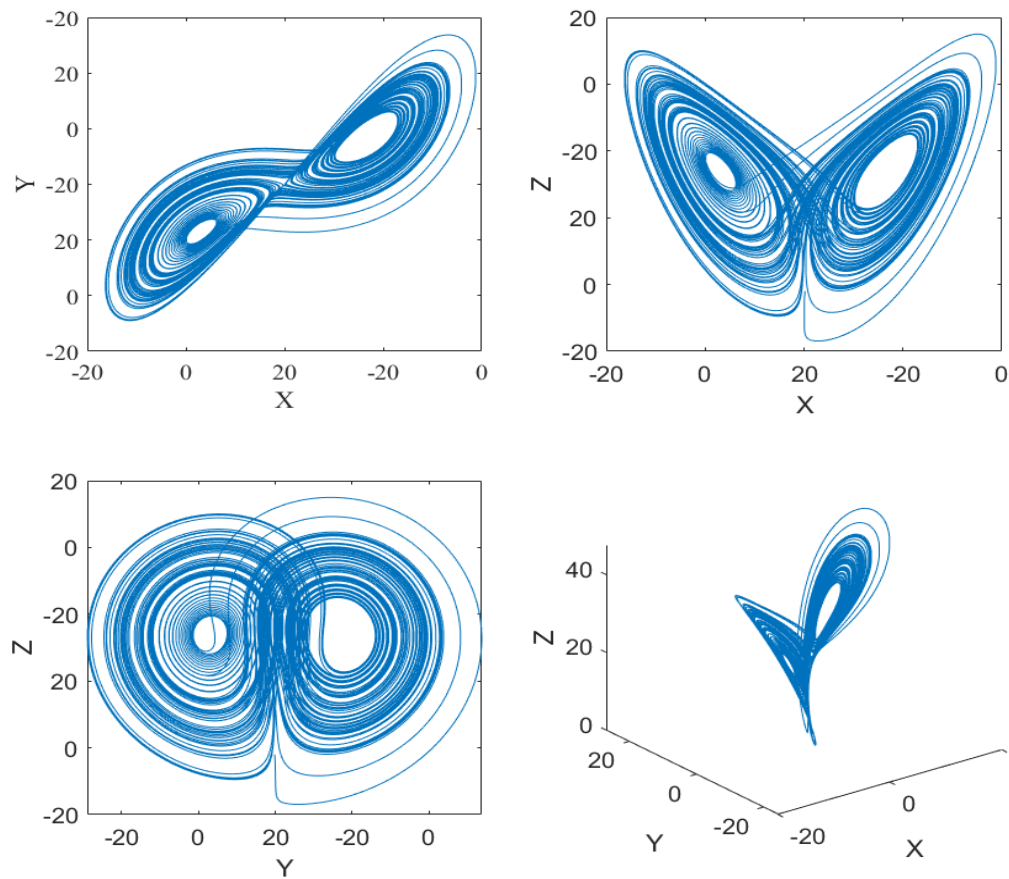


Fig. 11. Phase portraits of the Lorenz chaotic system: XY, XZ, YZ and XYZ projections

6.2. Lyapunov Dimension & Lyapunov Spectrum Analyses

In three-dimensional continuous-time systems, when calculating the Lyapunov exponents at a specific parameter value, it is expected that λ_1 , λ_2 and λ_3 , the Lyapunov exponents, have the sign sequence of (+, 0, -) for the system to exhibit chaotic behavior. In some cases, having a single positive Lyapunov exponent is sufficient [31]. Once a positive Lyapunov exponent is obtained, indicating the presence of chaotic behavior, the Lyapunov dimension is calculated using Equation 6, based on the Lyapunov exponents obtained at a fixed parameter value [32]. A higher Lyapunov dimension indicates a higher level of chaoticity.

The plotting of the Lyapunov exponents obtained for a selected parameter of three-dimensional continuous-time systems over a specific range is referred to as the Lyapunov exponent spectrum analysis. The Lyapunov exponent spectrum plots for the Lorenz, modified-Chameleon, and scaled-Zhongtang chaotic systems are shown in Figure 12, -Fig. A30, and -Fig. A31, respectively.

$$D = 2 + \frac{\lambda_1 + \lambda_2}{|\lambda_3|} \quad (6)$$

The Lyapunov exponent spectrum plot for the Lorenz chaotic system was obtained by plotting the spectrum over the range of 0-50 for the selected parameter b . Figure 12 shows the resulting graph. At the point where the parameter value b is 49.16, the first Lyapunov exponent λ_1 takes the highest value of 1.9287. For the parameter value $b = 49.16$, the Lyapunov exponents are as follows: $\lambda_1 = 1.9287$, $\lambda_2 = -0.0043$, $\lambda_3 = -15.5897$. Furthermore, using these Lyapunov exponents, the Lyapunov dimension of the Lorenz system is calculated to be $D = 2.12$.

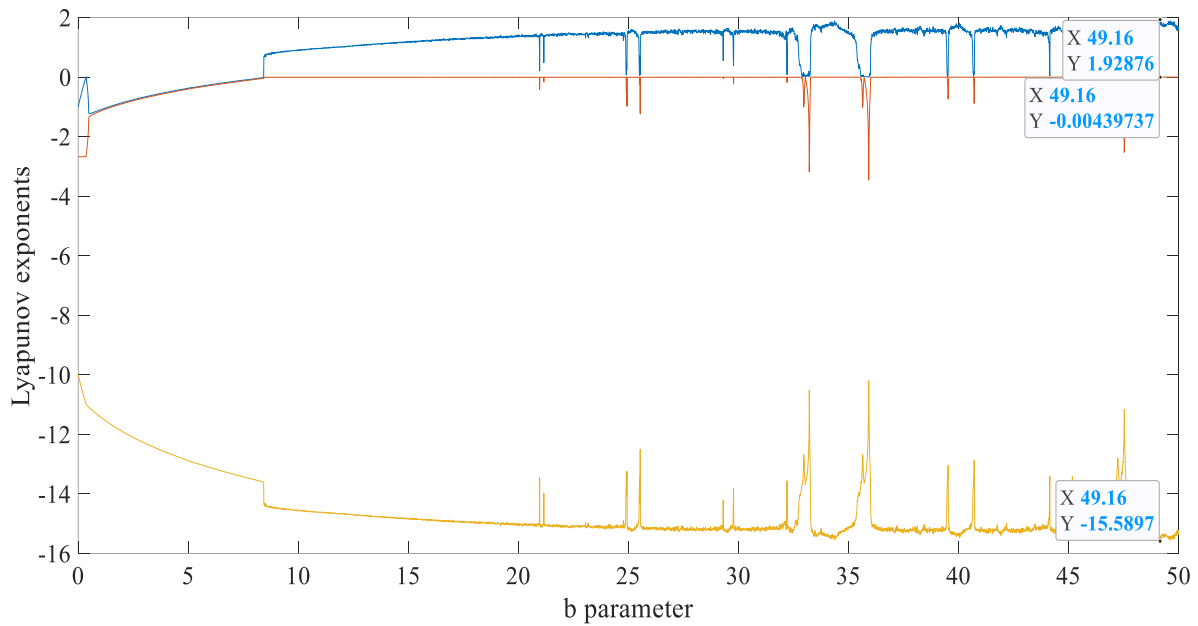


Fig. 12. Lyapunov Exponent Spectrum of the Lorenz Chaotic System.

The Lyapunov exponent spectrum graph for the Modified-Chameleon chaotic system was plotted in the range of 0-50 for the selected parameter c , resulting in the graph shown in Fig. A30. In this graph, at the value of $c = 11.1$, the first Lyapunov exponent λ_1 reaches its highest value of 0.1402. For the parameter value $c = 11.1$, the Lyapunov exponents are $\lambda_1 = 0.1402$, $\lambda_2 = 0.0017$, $\lambda_3 = -0.7995$, respectively. Furthermore, using these Lyapunov exponent values, the Lyapunov dimension of the modified-Chameleon system is calculated as $D = 2.17$.

The Lyapunov exponent spectrum of the scaled-Zhongtang chaotic system was obtained by plotting the graph in the range of 0-50 for the selected parameter value of g , resulting in the graph shown in Fig. A31. In this graph, at the point where $g = 48.83$, the first Lyapunov exponent $\lambda_1 = 9.6835$ reaches its highest value. For the parameter value of $g = 48.83$, the Lyapunov exponents are $\lambda_1 = 9.6835$, $\lambda_2 = -0.3105$, $\lambda_3 = -34.5711$. Furthermore, using these Lyapunov exponent values, the Lyapunov dimension of the scaled-Zhongtang system is calculated as $D = 2.27$.

The selected chaotic systems each contain multiple parameters. In order to enable experimental studies and comparisons of Lyapunov dimension values and positive Lyapunov exponents, these calculations have been performed using a single parameter value for each system. When calculating the Lyapunov dimension of each system, the parameter values at which they have the

highest positive Lyapunov exponent in the Lyapunov exponent spectrum graph have been taken into account.

According to the Lyapunov dimension calculations based on the mentioned criteria, the Lyapunov dimensions are $D = 2.27$ for the scaled-Zhongtang system, $D = 2.12$ for the Lorenz system, and $D = 2.17$ for the Modified-Chameleon system. Therefore, based on the Lyapunov dimension values, the scaled-Zhongtang system is the most chaotic system, followed by the Modified-Chameleon system and the Lorenz system.

According to the positive Lyapunov exponent values calculated based on the mentioned criteria, the positive Lyapunov exponents are $\lambda_1 = 9.6835$ for the scaled-Zhongtang system, $\lambda_1 = 1.9287$ for the Lorenz system, and $\lambda_1 = 0.1402$ for the Modified-Chameleon system. Therefore, based on the first positive Lyapunov exponent values, the scaled-Zhongtang system is the most chaotic system, followed by the Lorenz system and the Modified-Chameleon system.

Furthermore, when examining the Lyapunov exponent spectrum analysis, it can be observed that the scaled-Zhongtang system remains in chaos over a wider range of parameter values compared to the other two systems. Therefore, the scaled-Zhongtang system is the most chaotic among the selected systems. Similarly, the Lorenz system ranks second, followed by the Modified-Chameleon system.

6.3. FFT Analysis

The fast Fourier transform (FFT) analysis is an important criterion for detecting the chaotic behavior of a system. Through FFT analysis, information about the frequencies and magnitudes of the signals contained in the system at selected parameter values can be obtained [33]. Moreover, having signals in a wider bandwidth implies that the system may exhibit more chaotic behavior. In order to examine the bandwidth, FFT analysis of the chaotic systems used in the study have been conducted. The resulting graphs are shown in Figure 13 for Lorenz, Fig. A32 for Modified-Chameleon, and Fig. A33 for scaled-Zhongtang. When the same amplitude level is taken as reference in the FFT analysis, it is seen that the system that produces chaotic signals in the widest frequency range is the scaled-Zhongtang system, followed by the Lorenz and Modified-Chameleon chaotic systems, respectively. According to the FFT analysis criterion, the scaled-Zhongtang system exhibits the most chaotic characteristic.

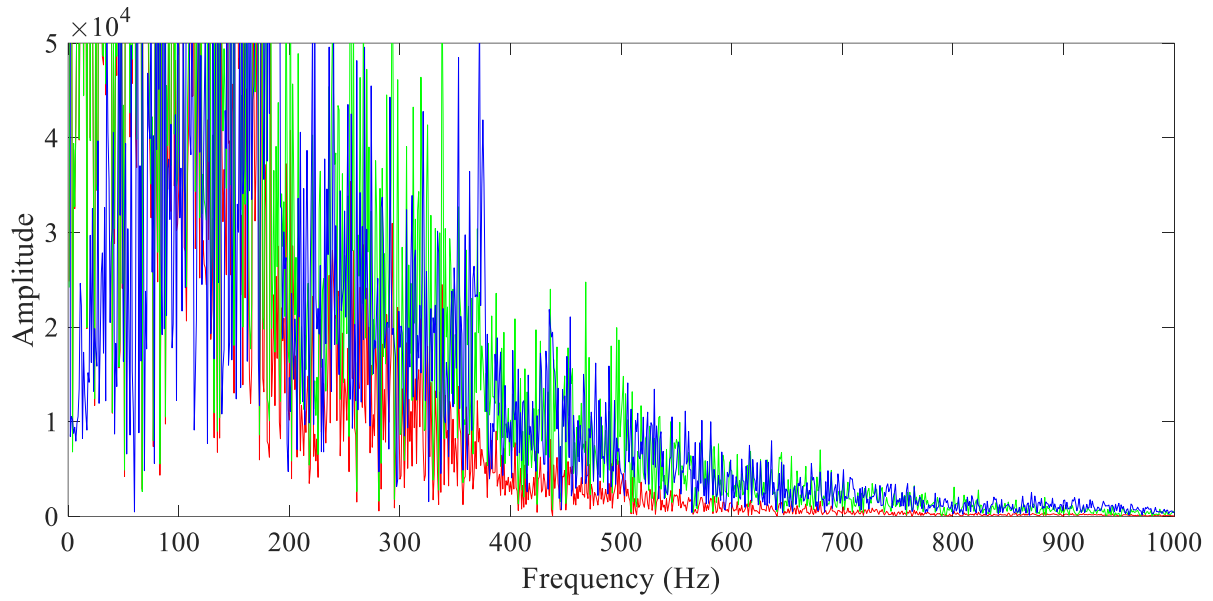


Fig. 13. FFT analysis of the Lorenz chaotic system

6.4. Comparison of The Chaoticity of The Systems Used

The results of the analysis indicate the chaotic characteristics of the selected systems, as shown in Table 4. The number of stars indicates the relative level of chaoticity based on the evaluation criteria. In determining which system is more chaotic among the selected ones, considerations were given to their symmetry in phase portraits with respect to the y-axis, positive Lyapunov exponent values, Lyapunov dimensions, having chaotic signals in a wider frequency range in FFT analysis, and their ability to maintain chaos over a wider range at the same parameter value. According to the conducted chaoticity tests, it can be observed that the scaled-Zhongtang chaotic system exhibits the highest level of chaoticity among the three systems: Lorenz, Modified-Chameleon, and scaled-Zhongtang. Following that, the Lorenz system ranks second, and the Modified-Chameleon system ranks third in terms of chaoticity properties.

	Asymmetry in phase portraits	Positive Lyapunov exponent values	Lyapunov dimension	Having chaotic signals over a wider frequency range (FFT analyses).	Staying in chaos with larger parameter value range
Lorenz chaotic system	---	★ ★	★	★ ★	---
Modified-Chameleon chaotic system	---	★	★ ★	★	---
Scaled-Zhongtang chaotic system	★ ★ ★	★ ★ ★	★ ★ ★	★ ★ ★	★ ★ ★

Table 4. Comparison of chaoticity levels of the systems

7. Discussions

In this study, the effects of chaotic systems on the performance of the PSO-based path planning algorithm have been extensively investigated. By integrating three different chaotic systems (ZPSO, LPSO, and CPSO) with the conventional PSO, a comprehensive comparison was conducted across seven different environments.

Chaotic systems with higher chaoticity exhibit more irregular and complex dynamics, which can enhance the exploration capability of metaheuristic algorithms. In the context of PSO, replacing pseudo-random numbers with highly chaotic sequences introduces greater diversity into the search process. This increased diversity enables particles to explore a wider region of the search space and reduces the likelihood of being trapped in local optima. Therefore, higher chaoticity can be considered a contributing factor to improved optimization performance, particularly in complex path planning problems.

Among the compared algorithms, ZPSO demonstrated the best performance in most scenarios. This can be attributed to the higher chaoticity of the system used in ZPSO, which enhances the exploration capability of PSO and allows the algorithm to traverse the solution space more effectively. In contrast, the conventional PSO algorithm exhibited the weakest performance across several scenarios, supporting the idea that the integration of chaotic systems can significantly improve optimization outcomes.

However, although ZPSO shows improved capability in exploring the search space and escaping local optima, it is not entirely immune to premature convergence in certain cases. Further

improvements to the ZPSO approach may enhance its performance, particularly in more complex environments such as Environment 7.

8. Conclusions

In this study, chaoticity analysis was performed on selected chaotic systems from the literature, namely the scaled-Zhongtang, Lorenz, and modified-Chameleon systems. The results indicate that, compared to the other two systems, the scaled-Zhongtang system exhibits several notable characteristics. It remains chaotic over a wider parameter range and demonstrates a broader bandwidth according to FFT analysis. In addition, its phase portraits show a more asymmetric structure compared to the other systems. For the parameter values used in this study, it also yields the highest Lyapunov dimension and the highest positive Lyapunov exponent among the considered systems.

In the analyzed chaotic systems used in the study, the scaled-Zhongtang system was found to exhibit more chaotic behavior compared to the other two systems based on chaoticity analysis. The Lorenz system ranked second in terms of chaoticity, while the modified-Chameleon system ranked third. Furthermore, the performance of these three different chaotic systems integrated into the PSO algorithm for the path planning optimization was compared. As in the results of the chaoticity analysis, the ZPSO algorithm, which incorporates the scaled Zhongtang system, which was used for the first time in PSO optimization, achieved the most successful results. The LPSO algorithm, which includes the Lorenz system, obtained moderately successful results. The CPSO algorithm, which incorporates the modified-Chameleon system, which was used for the first time in PSO optimization, achieved relatively lower success. The conventional PSO algorithm consistently performed poorly in each test. Considering the results of the chaoticity analysis and path planning optimization tests, it can be observed that chaotic systems with more levels of chaoticity are more successful in optimizing the PSO algorithm.

Future studies aim to test the developed algorithms in the Robot Operating System (ROS) environment. Instead of using ready-made path planning algorithms provided by ROS during autonomous navigation, the plan is to utilize the PSO-based path planning algorithm optimized with chaotic systems. This is expected to contribute to autonomous path planning research in the industry and academia within the popular ROS ecosystem.

Acknowledgements

The authors gratefully acknowledge the Scientific and Technological Research Council of Türkiye (TÜBİTAK) for the support provided during this study.

Funding

This work was supported by TÜBİTAK under the 2214-A International Research Fellowship Programme for PhD Students (Grant No: 1059B142200681).

Data Availability Statement

The datasets used and/or analysed during the current study are available from the corresponding author upon reasonable request.

Conflict of Interest

The authors declare that they have no known competing financial or non-financial interests that could have appeared to influence the work reported in this paper.

REFERENCES

- [1] L. Dong, Z. He, C. Song, and C. Sun, “A review of mobile robot motion planning methods: From classical motion planning workflows to reinforcement learning-based architectures”, *Journal of Systems Engineering and Electronics*, vol. 34 no. 2, pp. 439-459, 2023, doi: 10.23919/JSEE.2023.000051
- [2] G. Atalı, İ. Pehlivan, B. Gürevin and H. İ. Şeker, “Chaos in metaheuristic based artificial intelligence algorithms: A short review”, *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 29 no. 3, 2021, doi: 10.3906/elk-2102-5
- [3] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, & S. Mirjalili, “An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field”, *IEEE access*, vol. 8, pp. 186638-186652, 2020, doi: 10.1109/ACCESS.2020.3029728.
- [4] A. Mozaffari, M. Emami, A. Fathi, “A comprehensive investigation into the performance, robustness, scalability and convergence of chaos-enhanced evolutionary algorithms with boundary constraints”, *Artificial Intelligence Review*, vol. 52, pp. 2319-2380, 2019.
- [5] S. K. Pattnaik, D. Mishra and S. Panda, “A comparative study of meta-heuristics for local path planning of a mobile robot”, *Engineering Optimization*, vol. 0 no. 0, pp. 1-19, 2021, doi: 10.1080/0305215X.2020.1858074

- [6] J. Xin, J. Zhong, F. Yang, Y. Cui and J. Sheng, "An Improved Genetic Algorithm for Path-Planning of Unmanned Surface Vehicle", *Sensors*, vol. 19 no. 11, 2019, doi: 10.3390/s19112640
- [7] M. Elhoseny, A. Tharwat and A. E. Hassanien, "Bezier Curve Based Path Planning in a Dynamic Field using Modified Genetic Algorithm", *Journal of Computational Science*, vol. 25, pp. 339-350, 2018, doi: 10.1016/j.jocs.2017.08.004
- [8] C. Lamini, S. Benhlima and A. Elbekri, "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning", *Procedia Computer Science*, vol. 127, pp. 180-189, 2018, doi: 10.1016/j.procs.2018.01.113
- [9] H. I. Kang, B. Lee, and K. Kim, "Path Planning Algorithm Using the Particle Swarm Optimization and the Improved Dijkstra Algorithm", *2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application*, vol. 2, pp. 1002-1004, 2008, doi: 10.1109/PACIIA.2008.376
- [10] J. Lian, W. Yu, and W. Liu, "A Chaotic Adaptive Particle Swarm Optimization for Robot Path Planning", *Chinese Control Conference (CCC)*, pp. 4751-4756, 2019, doi: 10.23919/ChiCC.2019.8865591
- [11] M. M. AL-Nayar, K. E. Dagher and E. A. Hadi, "A comparative study for wheeled mobile robot path planning based on modified intelligent algorithms", *The Iraqi Journal for Mechanical and Materials Engineering*, vol. 19, no. 1, pp. 60-74, 2019.
- [12] Q. Yuan, Y. Zhang, X. Dai and S. Zhang, "A Modified Reptile Search Algorithm for Numerical Optimization Problems", *Computational Intelligence and Neuroscience*, 2022, doi: 10.1155/2022/9752003
- [13] A. Ait-Saadi, Y. Meraihi, A. Soukane, A. Ramdane-Cherif, and A. Benmessaoud Gabis, "A novel hybrid Chaotic Aquila Optimization algorithm with Simulated Annealing for Unmanned Aerial Vehicles path planning", *Computers and Electrical Engineering*, vol. 104, pp. 108461. doi: 10.1016/j.compeleceng.2022.108461
- [14] X. Cheng, J. Li, C. Zheng, J. Zhang and M. Zhao, "An Improved PSO-GWO Algorithm with Chaos and Adaptive Inertial Weight for Robot Path Planning", *Frontiers in Neurorobotics*, vol. 15, 2021, doi: 10.3389/fnbot.2021.770361
- [15] D. Wu, F. Hao, C. Yuan and Y. Li, "An improved sparrow search algorithm for mobile robot path planning", *2022 41st Chinese Control Conference (CCC)*, pp. 1899-1903. doi: 10.23919/CCC55666.2022.9902116

- [16] C. Ntakolia, S. Moustakidis and A. Siouras, "Autonomous path planning with obstacle avoidance for smart assistive systems", *Expert Systems with Applications*, vol. 213, pp. 119049, 2023, doi: 10.1016/j.eswa.2022.119049
- [17] H. Chu, J. Yi and F. Yang, "Chaos Particle Swarm Optimization Enhancement Algorithm for UAV Safe Path Planning", *Applied Sciences*, vol. 12, no. 18, 2022, doi: 10.3390/app12188977
- [18] Vikas and D. R. Parhi, "Chaos-based optimal path planning of humanoid robot using hybridized regression-gravity search algorithm in static and dynamic terrains", *Applied Soft Computing*, vol. 140, pp. 110236, 2023, doi: 10.1016/j.asoc.2023.110236
- [19] R. Hosseinpourfard, M. M. Javidi, "Chaotic PSO using the Lorenz System: An Efficient Approach for Optimizing Nonlinear Problems", *Cankaya University Journal of Science and Engineering*, vol. 12, no. 1, 2015.
- [20] Cheng, G., & Liu, Y. (2026). Hybrid optimization algorithm for solving path planning problems based on grey wolf optimization algorithm. *Scientific Reports*, 16(1), 8479.
- [21] Ugwoke, K. C., Nnanna, N. A., & Abdullahi, S. E. Y. (2025). Simulation-based review of classical, heuristic, and metaheuristic path planning algorithms. *Scientific Reports*, 15(1), 12643.
- [22] Li, C., Si, Q., Zhao, J., & Qin, P. (2024). A robot path planning method using improved Harris Hawks optimization algorithm. *Measurement and Control*, 57(4), 469-482.
- [23] Zhao, H., Zhang, K., Li, X., & Jin, J. (2025). MYIGWO: A grey wolf optimizer with dual mutation and chaotic adaptive neighborhood for engineering problems and path planning. *Advances in Engineering Software*, 211, 104044.
- [24] Huang, H., & Fu, S. (2025, June). Research on Path Planning Technology for Unmanned Aerial Vehicles by Integrating Chaotic Bat Algorithm with Artificial Potential Field Algorithm. In *2025 9th International Conference on Robotics and Automation Sciences (ICRAS)* (pp. 1-6). IEEE.
- [25] İ. Pehlivan and Y. Uyaroğlu, "A new chaotic attractor from general Lorenz system family and its electronic experimental implementation", *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 18, no. 2, pp. 171-184, 2010, doi:10.3906/elk-0906-67
- [26] A. Akgul, B. Gurevin, I. Pehlivan, M. Yildiz, M. C. Kutlu and E. Guleryuz, "Development of microcomputer based mobile random number generator with an encryption application", *Integration*, vol. 81, pp. 1-16, 2021, doi: 10.1016/j.vlsi.2021.04.010

- [27] Ü. Çavuşoğlu, A. Zengin, I. Pehlivan and S. Kaçar, “A novel approach for strong S-Box generation algorithm design based on chaotic scaled Zhongtang system”, *Nonlinear Dynamics*, vol. 87, no. 2, pp. 1081-1094, 2017, doi: 10.1007/s11071-016-3099-0
- [28] J. Kennedy and R. Eberhart, “Particle swarm optimization”, *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, pp. 1942-1948, 1995, doi: 10.1109/ICNN.1995.488968
- [29] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh and S. Mirjalili, “Particle Swarm Optimization: A Comprehensive Survey”, *IEEE Access*, vol. 10, pp. 10031-10061, 2022, doi: 10.1109/ACCESS.2022.3142859
- [30] J. P. Singh and B. K. Roy, “A more chaotic and easily hardware implementable new 3-D chaotic system in comparison with 50 reported systems”, *Nonlinear Dynamics*, vol. 93, no. 3, pp. 1121-1148, 2018, doi: 10.1007/s11071-018-4249-3
- [31] N. Sene, “Qualitative Analysis of Class of Fractional-Order Chaotic System via Bifurcation and Lyapunov Exponents Notions”, *Journal of Mathematics*, 2021, doi: 10.1155/2021/5548569
- [32] S. Sahoo and B. K. Roy, “Design of multi-wing chaotic systems with higher largest Lyapunov exponent”, *Chaos, Solitons & Fractals*, vol. 157, pp. 111926, 2022, doi: 10.1016/j.chaos.2022.111926
- [33] L. S. Yousuf, “Investigation of chaos in a polydyne cam with flat-faced follower mechanism”, *Journal of King Saud University - Engineering Sciences*, vol. 33, no. 7, pp. 507-516, 2021, doi: 10.1016/j.jksues.2020.07.001