Robotica (2026), **44**, pp. 1990–2015
doi:10.1017/S0263574726103865

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Adaptive Dragonfly Algorithm for AI-enabled path planning of humanoid robots in obstacle-constrained environments

Himansu Sekhar Dash[1,2] and Dayal Ramakrushna Parhi[3]

[1]Mechanical Engineering Department, NIT Rourkela, India
[2]Production Engineering Department, Indira Gandhi Institute of Technology, Odisha
[3]Mechanical Engineering, National Institute of Technology Rourkela, India
Corresponding author. Email: 919me5062@nitrkl.ac.in

Received: 6 May 2026 UTC; **Revised:** 23 June 2026 UTC; **Accepted:** 22 July 2026 UTC;
First published online: 4 September 2026 UTC

Keywords: humanoid robot navigation; Adaptive Dragonfly Algorithm (aDA); path planning; cluttered environments; artificial intelligence; obstacle avoidance

Abstract

Reliable path planning is a significant requirement for humanoid robots to move around in an environment filled with obstacles. In this paper, a novel method for humanoid robot navigation is proposed based on the Adaptive Dragonfly Algorithm (aDA), which is an improvement over the traditional Dragonfly Algorithm. The aDA method provides a better trade-off between exploration and exploitation of the search space, thereby avoiding local optima and increasing diversity in the search process. The aDA approach has been implemented and evaluated in a range of Webots-based simulation and experimental environments. Its performance is assessed using both single and multiple humanoid robots to analyze navigation capabilities under static as well as dynamic conditions. In multi-robot scenarios, conflict situations arise since each robot can act as a moving obstacle for the others, to overcome such situation, the dining philosopher controller is integrated with the proposed technique. The results of the aDA method are compared with traditional heuristics and metaheuristics for different scenarios of obstacle densities. The results obtained from the virtual and real environments show a variation within 5%, which proves that the aDA method is a robust and intelligent solution for humanoid robot navigation in a real-time environment. Furthermore, the proposed controller is benchmarked against an existing navigation method, demonstrating its improved performance and effectiveness.

1. Introduction

Humanoids are used in areas where human interaction and adaptability are needed. The humanoid shape helps handle objects and navigate through spaces designed for humans, using sensors, actuators, and control systems. But efficient path planning in a changing real-world environment is a challenge. In this paper, a new Adaptive Dragonfly Algorithm, aDA, is proposed for efficient path planning in a humanoid robot, specifically for the 25-DoF humanoid robot, NAO V4 H25, designed by Aldebaran Robotics [1], which is famous for its rich sensors, accurate motor control, and programable interface with multiple language support.

Traditionally, path planning in robotics has been addressed using various techniques, including graph-based and sampling-based methods. Classical methods such as Dijkstra's algorithm [2] and the A* search algorithm [3] provided the basis for solving shortest-path problems in discrete spaces. However, the applicability of these methods is reduced in high-dimensional spaces. To overcome the limitations of the above methods, sampling-based motion planning techniques such as Probabilistic Road Maps [4] and Rapidly Exploring Random Trees (RRT/RRT*) [5] were introduced as alternatives for motion

© The Author(s), 2026. Published by Cambridge University Press.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

planning in continuous spaces. Recent surveys on motion planning techniques highlight that, although classical methods are computationally efficient, they lack the ability to effectively trade off optimality, smoothness, and real-time adaptability in complex spaces [6, 7].

The application of metaheuristic algorithms, rooted in nature, has been recognized as an efficient alternative for robot path planning. Some of these algorithms include the application of Genetic Algorithm (GA) [8], Particle Swarm Optimization (PSO) [9], and Ant Colony Optimization [10]. Recent developments in metaheuristic algorithms include the application of the Dragonfly Algorithm (DA), proposed by Mirjalili [11], which is based on the swarming and dynamic/static behaviors of dragonflies. Another recent application of metaheuristic algorithms in path planning includes the Butterfly Optimization Algorithm (BOA), proposed by Arora & Singh [12], which is based on the food scent intensity. The application of metaheuristic algorithms in path planning has been confirmed to be efficient, according to the systematic review proposed by Patle et al. [13]. Tu and Yang [14] focused on a 2D static and dynamic environment for the path navigation of a robot by using GA. Wang et al. [15] focus on the automatic adjustment of crossover and mutation probabilities in response to changing environmental conditions, a method they refer to as the adaptive GA for route optimization of mobile robots in unknown environments. The proposed techniques achieve better results than GA in route optimization for a wheeled robot. Yang and Luo [16] focused on the pathfinding of a cleaning robot by using an NN approach. The objective is to reach all the points within the workspace by avoiding obstacles. Wang et al. [17] compared the traditional approach with the neural network-based path planning approach, with a novel technique called NN-driven prediction (NEED) for route navigation of the robot. NEED can anticipate the most promising search region for new route optimization by analyzing the surrounding structure. Patel et al. [18] found that the firefly algorithm (FA), based on concentric spheres, is an effective approach in the route identification of a wheeled manipulator. The approach is ideal for piloting around obstacles in changing environments and offers superior performance compared to other intelligent controllers. Chen et al. [19] suggested a global route optimization model to determine optimal paths under various objective functions, based on the modified FA. The concept and process of identifying the Path Center (PC) are proposed to calculate the distance between any two fireflies. Based on the inputs such as obstacle distance and heading angle from the sensors, the modified cuckoo search algorithm is implemented by the author Kumar et al. [20] in the motion planning of a wheeled robot. Zhang et al. [21] focused on an improved Gray Wolf Algorithm (IGWA) for route finding in a wheeled robot. The investigation found that IGWA performs quicker than multiobjective, random tree, and bee colony approaches when compared to established methods for route finding. Dash et al. [31] emphasized the need to integrate sensing during path planning using the improved butterfly optimization algorithm. Paital et al. [22] proposed a modified GWO for route optimization of a wheeled manipulator. The experimental findings are validated by each corresponding result obtained from the virtual environment using V-REP, and the variation in the results is within an acceptable range for motion planning. Liu et al. [23] used the BFO technique to reduce the path length and the time spent at turns while avoiding obstacles during the navigation of a wheeled robot.

To address the inherent limitations of single metaheuristics, hybridization has become a dominant research direction. Zhou et al. [24] proposed GDEBOA, which incorporates Gaussian estimation with BOA to enhance exploration and avoid premature convergence. Ali et al. [25] proposed a hybrid model that incorporates DA with Hopfield neural networks, while Zhao et al. [26] proposed HIDA, which incorporates local refinement techniques with DA. Fuzzy-based DA hybrids, as presented in ref. [27] by Patel et al., have exhibited increased robustness and flexibility in dynamic environments with obstacles. Reviews presented in refs. [28] and [29] indicate that these hybrid models generally provide a superior trade-off between global and local search, making them more appropriate for real-time robotic navigation scenarios.

Unlike robots with wheels or flying capabilities, humanoid robots need path-planning strategies that take into account whole-body movements, balance, and zero moment point constraints. Kashyap et al. [30] developed a hybrid approach by combining Dynamic Window Approach and the Teaching–Learning-Based Optimization to find out the best turning angle during humanoid navigation.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Furthermore, an intelligent hybrid motion planning approach implemented by Sahu et al. [40] combining regression modeling, and the FA was employed for humanoid navigation, with the regression component providing an initial steering angle that is further optimized to generate the final movement command. An enhanced Gravitational Search Algorithm incorporating the Hill Valley strategy was presented by Vikas et al. [41] to improve humanoid robot navigation by strengthening the search process and determining optimal steering angles while avoiding local optima. A hybrid moth–flame optimization and reinforcement learning controller is proposed by Kashyap et al. [42] to improve humanoid robot navigation through efficient path planning and obstacle avoidance. Simulation tools such as VREP and Webots are commonly used to evaluate humanoid path-planning strategies, allowing the smoothness, stability, and flexibility of trajectories to be tested.

However, despite the developments in various standalone and hybrid approaches of metaheuristics, most of the research in mobile robotics and/or path planning is limited to mobile robots and/or abstract scenarios. In humanoid robotics research, although balance and obstacle avoidance are considered, adaptive swarm-based hybrid approaches of metaheuristics and/or multiobjective optimization are not explored. The reason for this is the lack of a unifying framework for path planning that incorporates adaptive metaheuristic operators and humanoid locomotion constraints and/or realistic simulations. The contribution of this paper is the combination of aDA with a structured decision-making approach, which contributes to improving convergence and path generation. Unlike the previous approaches, the new method has low computational complexity and high accuracy that is appropriate for real-time humanoid robots.

To bridge this gap in the literature, the current research presents an aDA, which enhances DA through adaptive parameter control and crossover-based exploitation. The novelty lies in encoding humanoid path objectives such as path length, smoothness, time, and collision avoidance into aDA's cost function and validating it in Webots under varying obstacle densities. This combination of adaptive swarm optimization and humanoid-specific evaluation establishes aDA as a novel contribution to the literature. An exhaustive comparative study is conducted under similar environment setups, further showing that aDA achieves the best path reduction and faster rate of convergence. The effectiveness and robustness of the proposed strategy are verified by both simulations and experiments, which demonstrates that it can be used in real-time humanoid robot navigation. The proposed method is outlined in detail in the following section.

2. Problem formulation

This section explores the trajectory design, specifically the objective function, for multi-robot navigation. The key idea is to find the optimal path from the start to the destination, balancing safety (to avoid obstacles), path smoothness, shorter path length, and shorter travel time. Designing an efficient objective function is vital in the optimization-based path planning for humanoid robots. The objective function determines the trade-off between efficiency, feasibility, and smoothness of the path. In this work, a general objective function is developed by considering four vital aspects: path length, path safety, path smoothness, and travel time. All these aspects of the path ensure that the path is short, safe, smooth, and takes less time to travel. Moreover, a rule of "feasibility first" is used, where any path that does not satisfy the collision constraint is ranked lower than all feasible paths. For the development of the proposed path-planning framework, the following assumptions are considered. First, the humanoid robot is approximated as a point entity during the global path-planning stage to simplify the navigation problem. Second, environmental obstacles are modeled using geometric representations, and appropriate safety buffers are incorporated to account for the physical dimensions of the robot. Third, the path-planning aim is to determine an optimal collision-free route that optimizes the travel distance from the designated start location to the specified goal position. Finally, low-level locomotion functions, such as gait synthesis, balance regulation, and joint actuation constraints, are assumed to be handled by independent control mechanisms and are therefore excluded from the present investigation.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

2.1. Path optimization

This section describes the mathematical formulation and rationale behind each term in detail. The objective function is designed to balance efficiency, feasibility, and motion quality in humanoid path planning. Four performance indices are considered: path length, path smoothness, path safety, and traversal time. Each measure is normalized to the range [0, 1], and a weighted aggregation is used to form the overall cost. Let the workspace of the humanoid robot be denoted by $W \in R^d$ where d denotes the dimension of the workspace, i.e., 2D or 3D space. The path-planning problem is defined over this workspace with two specific conditions: a start point (S) $\in W$ representing the initial location of the humanoid robot, and a goal point (T) $\in W$ representing the desired destination. In addition, the workspace contains a finite set of obstacles (O), denoted by $O \in W$, with boundary ∂O. The robot's trajectory must strictly avoid these obstacles. A path (P) is expressed as an ordered sequence of waypoints as present in Eq. (1).

$$P = [P_1, P_2, P_3, \ldots\ldots\ldots\ldots P_m] \tag{1}$$

where each waypoint $P_k \in W$ for $k = 1, 2, 3 \ldots\ldots\ldots..m$. The first waypoint $P_1$is fixed at the start point S, and the last P_m is fixed at the goal point T. The intermediate waypoints ($P_2, P_3, P_4 \ldots\ldots\ldots\ldots..P_{m-1}$) are free variables that determine the geometric shape and feasibility of the route. For optimization purposes, these free waypoints are concatenated into a decision vector, as shown in Eq. (2).

$$X = vec(P_2, P_3, P_4, \ldots\ldots\ldots\ldots P_{m-1}) \in R^{(m-2)d} \tag{2}$$

Thus, each search agent in the population encodes one possible path as a position vector X in $(m-2)d$ dimensional space. To mathematically capture safety requirements, we define a signed distance function as mentioned in Eq. (3).

$$D(X) = \min_{O \in \partial O} ||X - O|| \tag{3}$$

where $D(X) > 0$ indicates that X lies outside the obstacle region, and $D(X) < 0$ indicates penetration into an obstacle. This function is critical for measuring clearance and imposing collision penalties in the objective function.

The fitness of a candidate path X_i is defined as a combination of weighted four objectives, such as path length $L(X_i)$, path smoothness $S(X_i)$, path safety $\varnothing_{safe}(X_i)$, and traversal time $T(X_i)$, into a single scalar cost with weights $\alpha_L, \alpha_S, \alpha_{safe}, \alpha_T \geq 0$ as presented in Eq. (4).

$$F(X_i) = \propto_L L(X_i) + \propto_S S(X_i) + \propto_{safe} \varnothing_{safe}(X_i) + \propto_T T(X_i) \tag{4}$$

$$\alpha_L + \alpha_S + \alpha_{safe} + \alpha_T = 1 \tag{5}$$

$$L(X_i) = \sum_{k=1}^{m-1} ||P_{k+1} - P_k|| \tag{6}$$

$$S(X_i) = \sum_{k=2}^{m-1} ||P_{k+1} - 2P_k + P_{K-1}||^2 \tag{7}$$

$$\varnothing_{safe}(X_i) = \lambda_{clr} \sum_{k=1}^{m} \max(0, d_{min} - D(P_k)) + \lambda_{col} \sum_{k=1}^{m-1} C(P_k, P_{K+1}) \tag{8}$$

$$(P_k, P_{K+1}) = \begin{cases} 1, & \textit{if the line segment } (P_k, P_{k+1}) \textit{ intersects any obstacles} \\ 0, & \textit{otherwise} \end{cases} \tag{9}$$

$$T(X_i) = \frac{L(X_i)}{V_o} \tag{10}$$

Where:
$D(P_k)$: The distance from the waypoint P_k to the nearest obstacle boundary.
d_{min} : Required minimum clearance from the obstacle.
λ_{clr} : Weight for clearance penalty.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

λ_{col}: Weight for collision penalty (large value).
$C(P_k, P_{K+1})$: Collision indicator for a segment
V_o: Constant velocity

All raw objective terms (path length $L(X_i)$, path smoothness $S(X_i)$, path safety $\varnothing_{safe}(X_i)$, and traversal time $T(X_i)$) are first normalized to the unit interval so that heterogeneous units and scales do not bias the search. For each quantity, we clamp the raw value to a chosen value within the interval $[X_{min}, X_{max}]$, and set the normalized terms in [0, 1] as per the equations mentioned below.

$$\widehat{L(X_i)} = \frac{clip(L(X), L\,(X_{min})\,, L(X_{max})) - L(X_{min})}{L(X_{max}) - L(X_{min})} \tag{11}$$

$$\widehat{S(X_i)} = \frac{clip(S(X), S\,(X_{min})\,, S(X_{max})) - S(X_{min})}{S(X_{max}) - S(X_{min})} \tag{12}$$

$$\widehat{T(X_i)} = \frac{clip(T(X), T\,(X_{min})\,, T(X_{max})) - T(X_{min})}{T(X_{max}) - T(X_{min})} \tag{13}$$

$$\widehat{\varnothing_{safe}(X_i)} = \frac{clip\big(\varnothing_{safe}(X), \varnothing_{safe}\,(X_{min})\,, \varnothing_{safe}(X_{max})\big) - T(X_{min})}{\varnothing_{safe}(X_{max}) - \varnothing_{safe}(X_{min})} \tag{14}$$

The final cost is a weighted sum with nonnegative weights $\alpha_L, \alpha_S, \alpha_T, and\ \alpha_{safe}$ chosen so that $\sum \alpha = 1$ is shown in Eq. (15), which is used as a fitness function in the aDA and discussed in the subsequent section.

$$F(X) = \alpha_L \hat{L} + \alpha_S \hat{S} + \alpha_T \hat{T} + \alpha_{safe} \widehat{\varnothing_{safe}} \tag{15}$$

2.2. Architecture of Adaptive Dragonfly Algorithm (aDA)

The DA, introduced by Mirjalili [11], is a swarm intelligence-driven metaheuristic inspired by the swarming behavior of dragonflies in nature. Dragonflies exhibit two characteristic modes: static grouping, which represents exploration of new regions (food searching), and dynamic grouping, which corresponds to exploitation of promising areas (migration). DA mathematically encodes these behaviors to balance the global and local search capabilities of the algorithm. The interaction among agents is defined by five behavioral operators: separation (S_p), alignment (A_p), attraction toward the swarm center (i.e., cohesion, C_p), movement toward food sources (F_p), and avoidance *of enemies* (E_p), which are expressed in Table I.

These components are integrated into the step vector update equation:

$$\Delta X_{t+1} = sS_p + aA_p + cC_p + fF_p + eE_p + w\Delta X_t \tag{16}$$

where s, a, c, f, e are the influence coefficients of each behavioral component, and w is an inertia weight controlling the memory of previous movements. The position vector governs the relocation of the search agent within the search space as shown in Eq. (17). An agent is considered isolated when it has no neighboring solutions within the interaction radius. Isolation indicates that the agent is in an exploratory region of the search space, so it switches to a stochastic operator instead of following swarm forces. In this mode, the agent performs a Lévy-based random walk, adding a Lévy-distributed step to its current position, which is present in Eq. (18). This heavy-tailed jump allows the agent to escape poor regions and discover new, potentially better areas of the path-planning environment.

$$X_{t+1} = X_t + \Delta X_{t+1} \tag{17}$$

$$X_{t+1} = X_t + Lev\,(d) \times X_t \tag{18}$$

The conventional DA relies solely on neighbor-based social interaction rules, which often causes premature convergence or stagnation when agents become isolated or trapped near obstacles in complex

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574724103865

Table I. *Representation of DA parameters used in the proposed approach.*

Operator	Expression	Remark
Separation (S_p)	$S_p = -\sum_{j=1}^{N} X - X_j$	X: Individual current position X_j: j-th neighboring individual location N: Neighboring individual number
Alignment (A_p)	$A_p = \frac{\sum_{j=1}^{N} V_j}{N}$	V_j: j-th neighboring individual velocity.
Cohesion (C_p)	$C_p = \frac{\sum_{j=1}^{N} X_j}{N} - X$	X: Individual current position N: Neighboring individual number X_j: j-th neighboring individual location
Attraction to food (F_p)	$F_p = X^+ - X$	X: Individual current position X^+: food source location
Distraction to enemies (E_p)	$E_p = X^- + X$	X: Individual current position X^-: Enemy location.

environments. By integrating hybrid Lévy [32], Gaussian [33], and OU [34] random-walk operators, the proposed adaptive DA approach significantly strengthens DA's search dynamics, which is shown in Eqs. (19) and (20). The long-range jumps of Lévy flights are beneficial in evading local minima and finding new unexplored safe paths, while Gaussian noise helps in refining paths in promising regions. This dual-scale exploration-exploitation strategy helps in preserving diversification in the search process and prevents premature convergence, which is common in standard DA algorithms. The incorporation of the OU process helps in stabilizing the drift toward the global best path, which prevents individual agents from drifting toward infeasible regions, a problem common in humanoid path planning, where safe paths are those, which follow smooth and collision-free paths.

$$X_{t+1}^i = X_t^i + RW(t) \tag{19}$$

where $RW(t)$ is the hybrid Lévy, Gaussian, and OU random-walk operators?

$$RW(t) = \begin{cases} \alpha_t \cdot Levy(\beta), & t < T_{switch} \\ N\left(0, \sigma^2\right), & t \geq T_{switch} \end{cases} \tag{20}$$

$$\alpha_t = \alpha_0 \left(1 - \frac{t}{T_{Max}}\right) \tag{21}$$

where T_{switch} is the switching iteration (e.g., 50% of the max iterations), α_t is the adaptive scaling operator that guarantees a smooth transition from exploration to refinement, avoiding large jumps at the end, and $Levy(\beta)$ is a heavy-tailed step. In aDA, the Ornstein–Uhlenbeck (OU) random walk is incorporated as a stabilizing exploration operator for isolated agents, ensuring that their movement remains directed and feasible in complex humanoid navigation tasks. Unlike unconstrained random steps, the OU process adds a drift term that continuously pulls the agent toward the current global best path while allowing controlled stochastic variation. The update is expressed as shown in Eq. (22).

$$X_{t+1}^i = X_t^i + \theta\left(X_{best} - X_t^i\right) + \sigma_{OU} N(0, 1) \tag{22}$$

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Pseudocode 1: aDA

Initialize the fitness function
Initialize population of dragonflies (candidate humanoid paths) X^i
Initialize control parameters (s, a, c, f, e) for aDA
Set α_0 (initial step size), (max iterations), T_{max} (max iteration), T_{switch} (explore → exploit switch), ρ (Probability)
for each iteration t = 1 to T_{max}
  for each agent i in population
    if agent i has neighbors **then**
      compute separation S_p, alignment A_p, cohesion C_p
      compute attraction to food F_p and distraction from enemy E_p
      update velocity using Eq. 16
      update position using Eq. 17
    else
      if $t/T_{max} < \rho$
      update the position using Eq. 19-21 "Hybrid Levy, Gaussian strategy"
      else
      update the position using Eq. 22 "Ornstein-Uhlenbeck strategy"
    end if
  end for
  project X^i into humanoid joint limits
  repair X^i if collision is detected
  evaluate the fitness
  update global best (food source) and worst (enemy) positions
end for
return best path

where the value of θ determines the strength of the attract, and σ_{OU} determines the amplitude of the noise. This prevents isolated nodes from moving into unproductive regions, yet still provides sufficient randomness to search for alternative productive paths. By including this mean-reverting behavior, the OU operator provides improved stability, convergence reliability, and guarantees the random search is still biased toward productive path structures, making it superior to the random walk used in the standard DA.

This way, a better balance is achieved in exploration and precision compared to the conventional approach of DA. This is due to the adaptive step-size reduction coupled with condition-based switching of the random walk operators in aDA. This results in safer, smoother, and shorter humanoid path planning compared to conventional DA. The subsequent section discusses the simulation results and the experimental validation of the aDA approach on a humanoid robot. Figures 1 and 2 show the flowchart of the proposed aDA and the flowchart detailing how the proposed aDA framework is implemented in a robotic environment, respectively. Figure 3 depicts the activation of the proposed aDA approach in a robot simulator, along with the resulting path generation. Using the traditional approach, the robot can escape the barriers without getting too close. However, when the aDA approach is activated in the robot's path planning system, the robot can move toward the obstacle up to a safe distance before moving away from it.

2.3. Dining philosopher controller for collision avoidance

The Dining Philosophers Problem, first introduced by Dijkstra in 1971 [35], describes a resource-sharing scenario in which multiple philosophers seated around a circular table compete for limited shared forks. Each philosopher must acquire two forks to eat, with mutual exclusion enforced between them.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

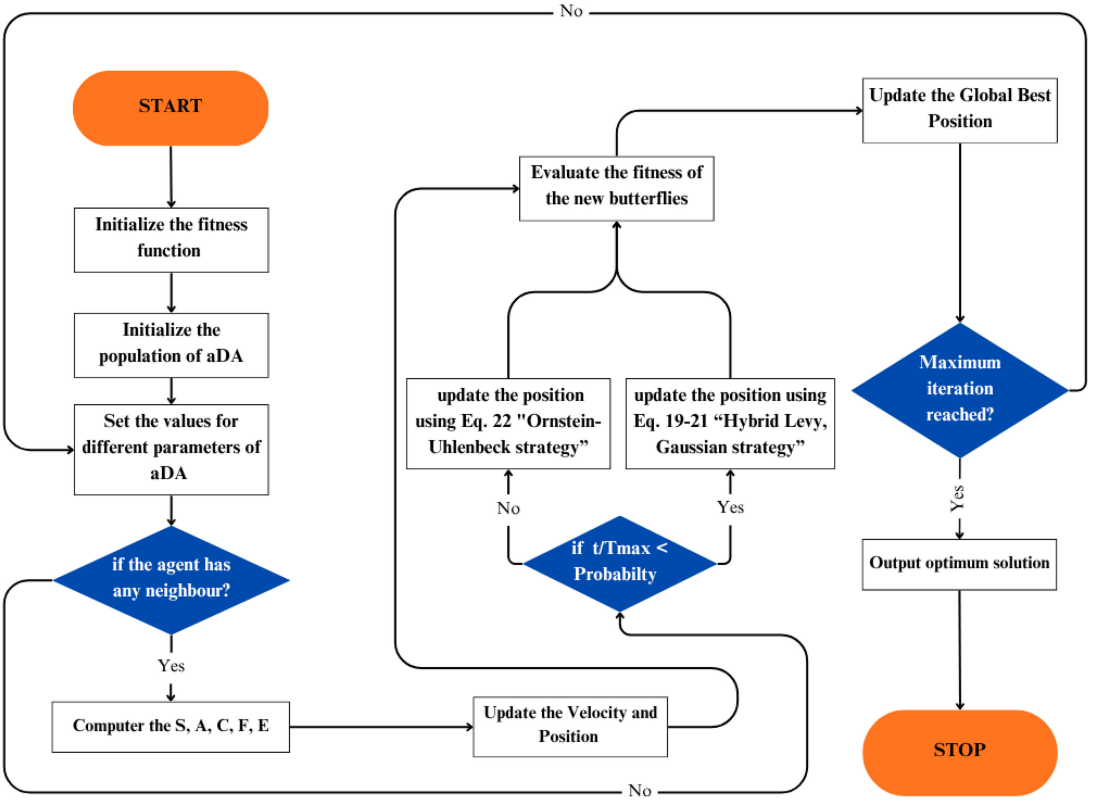


Figure 1. *The flowchart of the proposed adaptive DA (aDA) technique for humanoid navigation.*

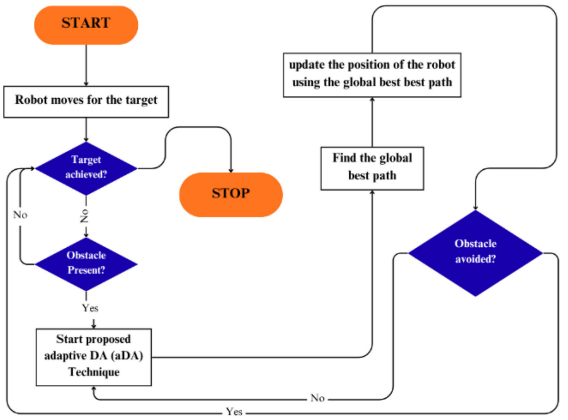


Figure 2. *The flowchart of the implementation of the proposed aDA framework within the robotic environment.*

Semaphore-based synchronization is commonly used to manage resource allocation, and priority-based strategies are used to prevent deadlock. Figure 4 depicts the three philosophers (P_1, P_2, P_3) seated around a circular table, with forks (F_1, F_2, F_3) placed between adjacent pairs and a common food source at the center, accessible to all.

The dining philosopher-based controller is derived from a classical problem in computer science known as the "Dining Philosophers Problem," which is commonly used in multi-robot path planning problems. In this controller, each robot is considered a philosopher, and the available paths or

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

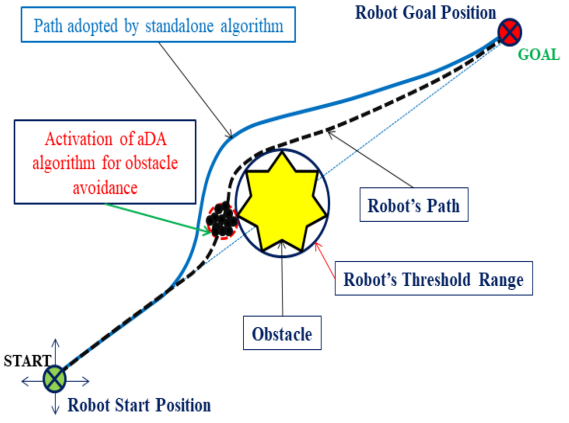


Figure 3. *The activation of the aDA and path generation using aDA approach to achieve the Goal.*

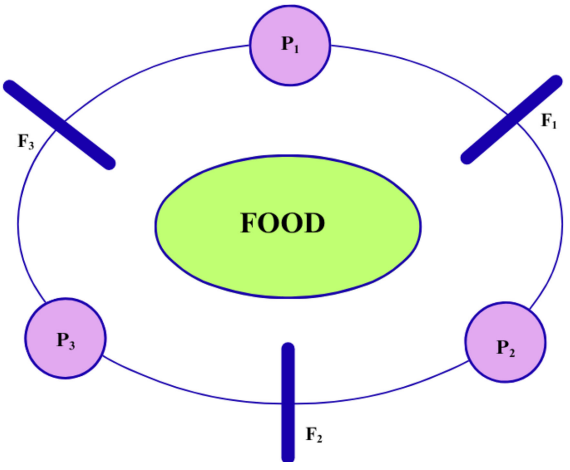


Figure 4. *The fundamentals of Dining Philosopher [35].*

intersections in a given environment are considered limited resources that can be accessed by a robot in an exclusive manner. A robot can move when all required resources are available, thereby ensuring collision-free robot movement in a given environment. The controller regulates the access and release of shared resources according to predefined coordination rules, ensuring that multiple robots do not simultaneously occupy conflicting regions of the environment. This mechanism helps to avoid collision, deadlock, and starvation conditions during multi-robot navigation. This controller can effectively control robots in a given environment by limiting access to certain regions in the environment. Figure 5 depicts a step-by-step explanation of the dining philosopher-based controller. For the problem considered in this manuscript, the controller follows these steps [36]:

Stage 1: The robots are placed at random positions on the field and wait for the start signal to move toward the objectives.

Stage 2: The robots start moving toward the objectives, possibly colliding with static obstacles.

Stage 3: The robots detect the obstacles during the movement.

Stage 4: The robot that is nearest to the goal is given priority.

Stage 5: The priority robot moves toward the goal, while the rest are stopped and act as obstacles.

Stage 6: The stopped robots are cleared, and the process is repeated for the rest of the robots.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Table II. *Technical specifications of the humanoid robot NAO (V4, H25) used in the current study [1].*

NAO Specification	
Version	NAO H25 V4.0
Developed by	Aldebaran Robotics
Dimension, material	574 H x 275W x 311D (mm), ABS-PC/PA-66/XCF-30
Weight	5.4 KG
Language	Understand and revert seven different languages
Cameras	MT9M114, SOC Image Sensor, 1,280×960 resolution at 30 FPS.
Motherboard	ATOM Z530 1.6 GHz CPU 1 GB RAM, 2 GB flash memory, 8 GB Micro SDHC
Input	100 to 240 V AC, 50–60 Hz and maximum 1.2A
Battery	Lithium-ion, 21.6 V, 2014AH, 27.6 WH. Working: up to 2 hours.
DOF	25 DOF, head: 2 DOF, arm: 6 DOF, leg: 5 DOF, hip Joint: 1 DOF.

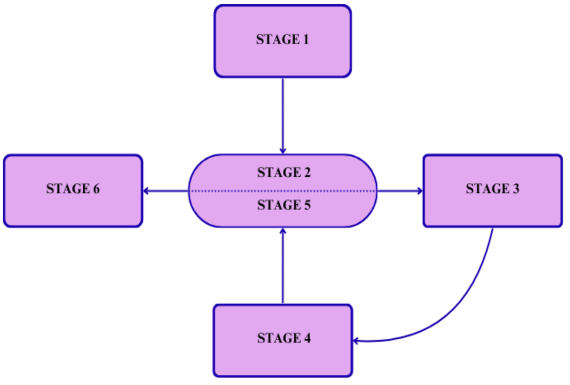


Figure 5. *The step-by-step functioning of the dining philosopher approach [36].*

3. Performance evaluation and experimental demonstration

To verify the efficiency and efficacy of the proposed adaptive DA-based path planning algorithm, extensive simulation and experimentation were carried out with a humanoid robot moving in both structured and unstructured environments. This section is intended to show the efficacy of the improved optimization algorithm in terms of optimality, stability, and safety. The efficacy of the algorithm was tested in terms of path length, smoothness, safety, and time to traverse the path to ensure a comprehensive qualitative and quantitative evaluation of the algorithm. The following sub-paragraphs will discuss the simulation and experimentation conducted on various scenarios.

3.1. Simulation-based environment modeling

The current work utilizes a humanoid robot, namely, the NAO H25 V4, which is a highly versatile platform for both simulation and physical experiments. The robot has 25 degrees of freedom, which provide high mobility for legged robots, while its sensors, cameras, and microphones provide advanced capabilities in terms of object detection, speech, and autonomous movement. The specification of the NAO robot is provided in Table II and Figure 6. The proposed framework for path optimization was first implemented in a simulated environment using MATLAB R2023b and a Webots platform, which are officially supported by Aldebaran Robotics [37], as both are widely used in robotics for simulation-based analysis of robotic path optimization algorithms, etc. All simulations were performed on a laptop with an Intel i5-1235U processor. To validate the performance of the proposed framework, the experiments

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

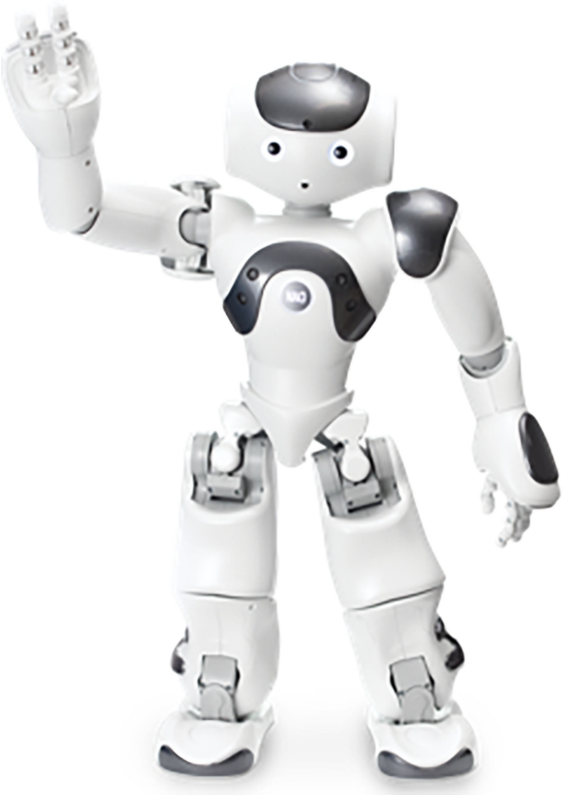


Figure 6. *Illustration of the structural layout of the NAO humanoid robot [1].*

were conducted in a laboratory environment using a humanoid robot, namely, the NAO robot, version H25, and V4.0. The results of the proposed controller for both single-robot and multi-robot scenarios are discussed in the following section.

3.2. Performance comparison across techniques

The aDA approach was developed according to the mechanism outlined in the previous section. Before deploying it on the NAO robot, its performance must be verified against the benchmark standalone methods used for the navigation of humanoid robot. The primary goal of this evaluation is to achieve an optimal path, one that is as short as possible while ensuring safe obstacle avoidance. To assess the effectiveness of all methods, a simulation environment is created in MATLAB containing a start point, a goal, a robot, and different barriers, as illustrated in Figure 7 (a). In this setup, the robot detects the obstacle once it enters the predefined threshold distance and initiates a turn to steer away from it.

To verify the efficiency of the proposed controller relative to individual optimization methods, the robot is directed toward a predefined target within an environment populated with randomly distributed obstacles. To evaluate solution stability, the proposed aDA was executed over multiple independent runs using identical parameter settings and environment configurations. The obtained convergence curves and final path lengths exhibit consistent behavior with minimal variance, indicating stable optimization performance despite the stochastic nature of the algorithm. The navigation task is evaluated using the benchmark PSO, GA, and DA approaches, with the primary objective of identifying the shortest feasible path while ensuring safe maneuvering around obstacles. A similar assessment is then carried out using the proposed aDA approach. The simulation results obtained from all four approaches are

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

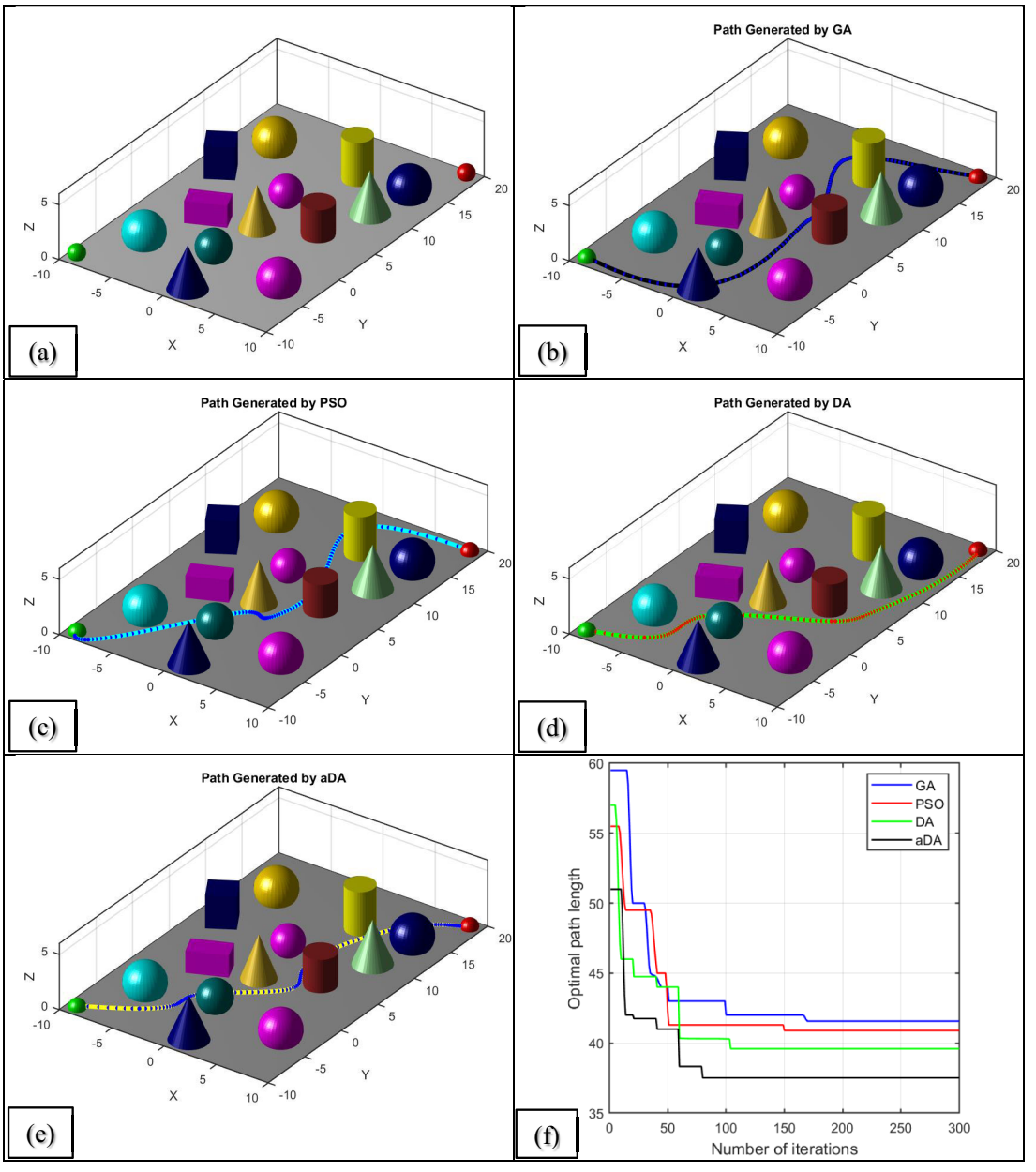


Figure 7. *The simulated path achieved by (b) GA, (c) PSO, (d) DA, (e) proposed aDA, and (f) convergence curve during virtual navigation.*

shown in Figure 7 (b)–(e) and the obtained optimal path lengths are shown in Table III with comparative analysis. Computational efficiency was assessed by measuring the number of iterations required to reach convergence. Compared with existing benchmark approaches, i.e., GA, PSO, and DA, the proposed aDA demonstrates faster convergence while maintaining comparable computational complexity.

Figure 7(f) demonstrates the convergence behavior of all four controllers in comparison to the optimal path length across iterations. As evidenced by the figure, the proposed controller exhibits notably faster convergence, reaching the optimal travel distance in substantially fewer iterations than the conventional, standalone methods such as PSO, GA, and DA. This efficient convergence translates directly to a lower

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Table III. *Demonstration of the improvement in path length using the proposed aDA method over other techniques in navigating a virtual cluttered environment.*

Sl. No.	Algorithm	Path length (cm)	Improvement (%)
1	GA	41.57	11.39
2	PSO	40.93	9.67
3	DA	39.61	6.14
4	aDA	37.32	–
	Average		**9.07**

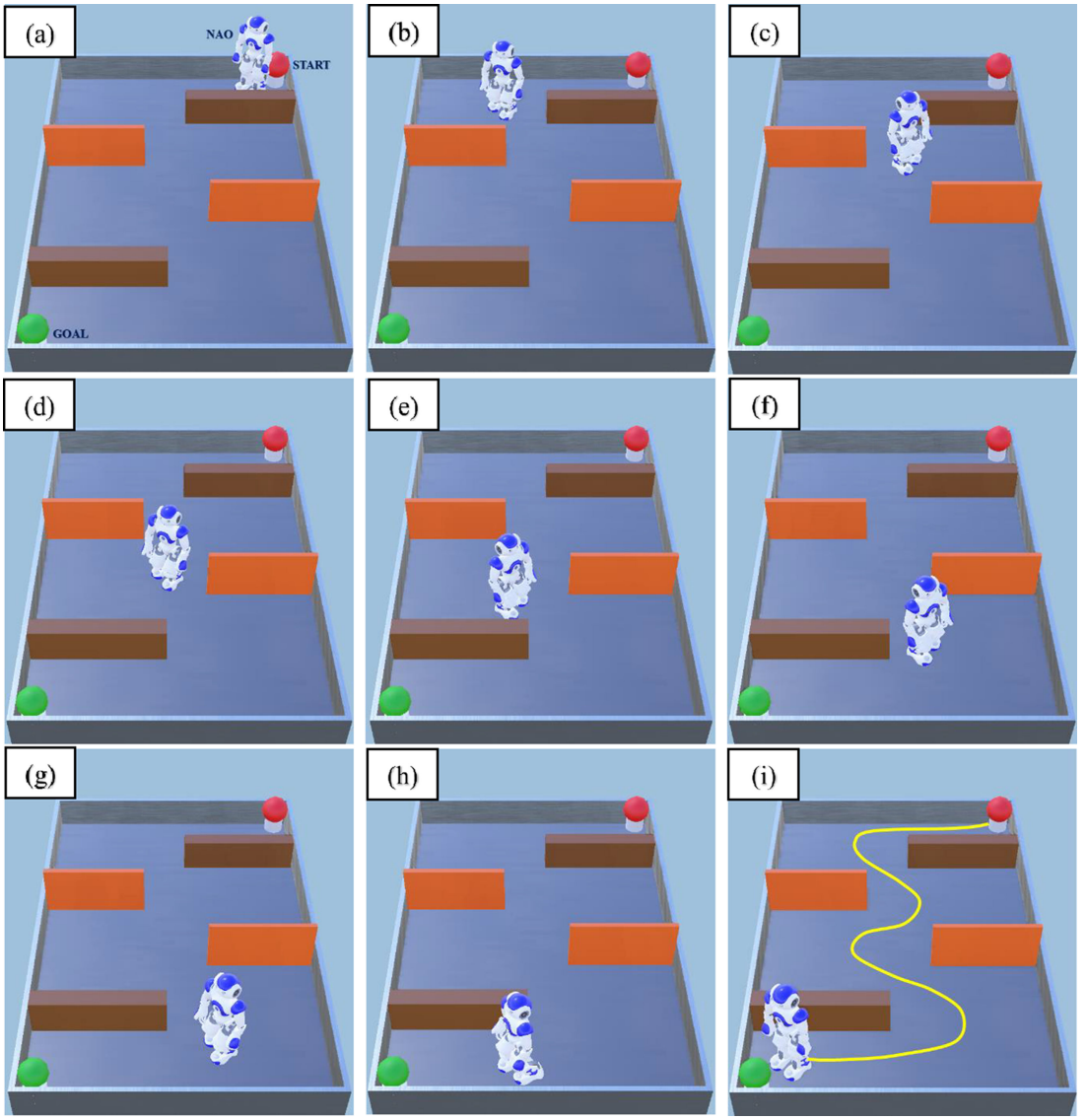


Figure 8. *Simulation result showing the path of a single NAO robot with the proposed aDA algorithm.*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use , available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Table IV. *Performance comparison of simulation and real-world tests for a single NAO humanoid with the aDA approach, measured by route length (cm) and completion time (s).*

	Simulation		Experiment		Deviation (%)	
Sl. No.	**RL (in cm)**	**CT (in sec)**	**RL (in cm)**	**CT (in sec)**	**PL**	**CT**
1	235.21	65.21	243.25	68.32	3.31	4.55
2	232.42	66.15	242.35	69.15	4.10	4.34
3	231.56	61.74	250.11	64.41	7.42	4.15
4	238.78	68.52	246.71	70.32	3.21	2.56
5	240.01	63.49	245.31	66.45	2.16	4.45
6	237.25	69.12	249.15	71.35	4.78	3.13
7	231.76	64.48	247.39	66.87	6.32	3.57
8	234.86	67.39	247.01	70.34	4.92	4.19
9	239.74	62.41	251.12	65.81	4.53	5.17
10	238.47	63.24	246.45	66.71	3.24	5.20
11	236.84	70.23	248.36	71.47	4.64	1.73
12	238.78	68.25	246.94	69.53	3.30	1.84
13	239.44	61.11	248.15	64.91	3.51	5.85
14	231.09	62.36	245.12	66.28	5.72	5.91
15	230.48	66.52	248.28	70.12	7.17	5.13
16	237.06	67.74	247.51	69.46	4.22	2.48
17	233.65	63.28	245.61	65.26	4.87	3.03
18	235.23	65.42	250.21	66.18	5.99	1.15
19	240.82	68.27	250.92	70.53	4.03	3.20
20	236.4	69.81	251.32	70.95	5.94	1.61
Avg.	235.99	65.74	247.56	68.22	4.67	3.66

computational burden and decreased operational cost for the navigation task, due to the reduced time required to determine an effective path.

3.3. Testing of the proposed aDA on a single NAO system

The effectiveness of the proposed aDA method is also validated in a virtual robotic environment and a real-world robotic setup. A customized test environment is created by introducing static obstacles of different geometrical shapes to simulate a partially concealed operational space. In the case of the navigation problem, a single NAO robot (N) is assigned a predefined start position (S) and destination (T). The primary objective is to reach the destination from the start position without any collision with the obstacles and by the shortest possible route. The shortest route is determined by selecting the route that provides the minimum possible Euclidean distance between the start and destination points. The simulation is carried out in the WEBOT platform [37], and the aDA algorithm is developed in MATLAB R2023b. The trajectory traced by the aDA method is shown in Figure 8.

The equivalent setup is also created in a real-world robotic environment to validate the proposed method by performing experiments on a single NAO robot. The experimental results are shown in Table IV, and the path traced by aDA is shown in Figure 9. RL stands for route length in cm, and CT stands for completion time in sec.

To ensure the reliability of the simulation and experimental results, each experiment is conducted multiple times for the same parameter settings. The performance metrics obtained are RL and completion time (CT), which are shown in normality test and histogram plots along with the mean and standard

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

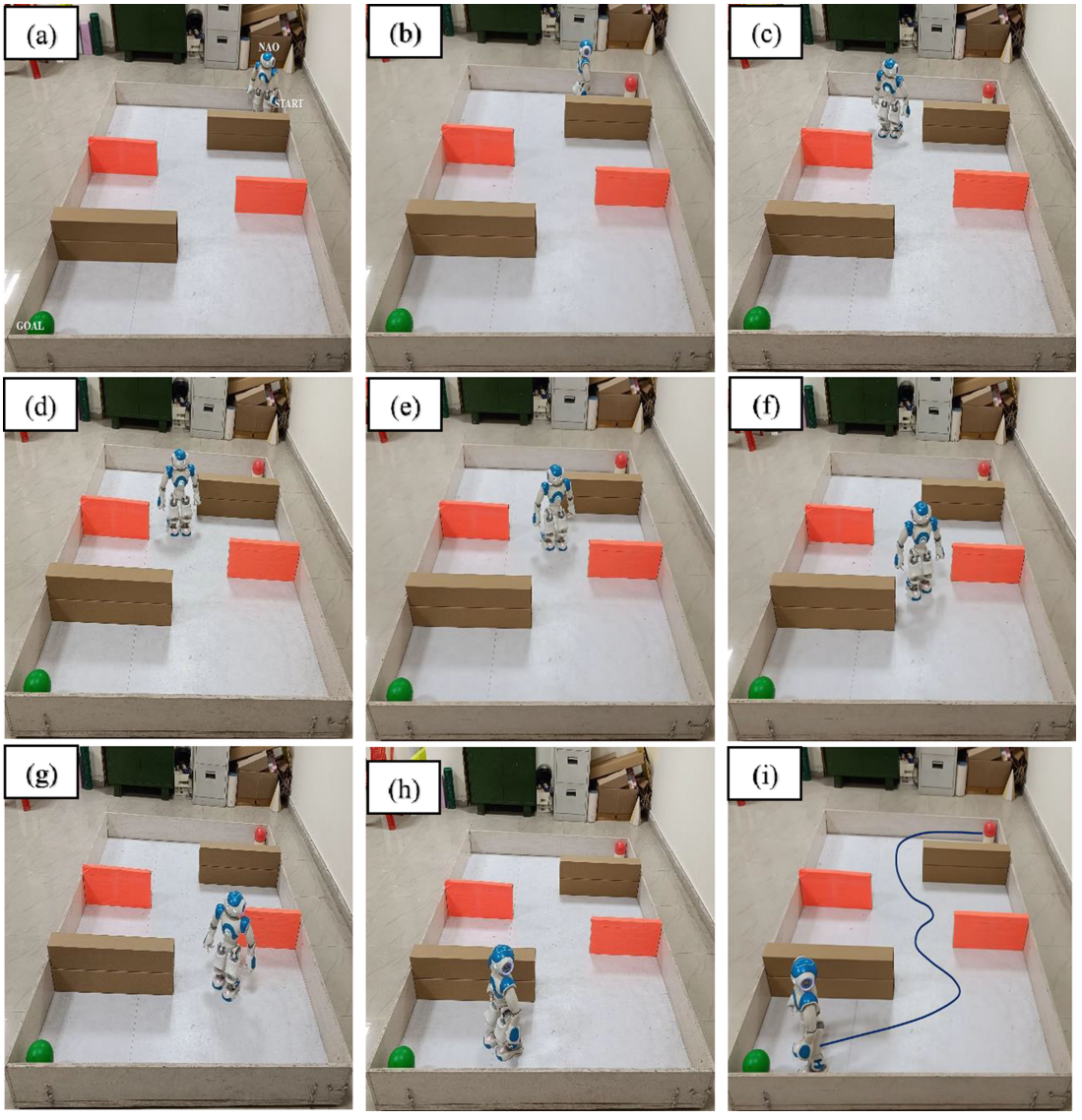


Figure 9. *Experimental result showing the path of a single NAO robot with the proposed aDA algorithm.*

deviation values. The results have shown that the gain obtained by the aDA method is statistically significant, which confirms the reliability and consistency of the method. The statistical analysis of the simulation and experimental results, conducted by aDA for a single NAO, is shown in Figures 10–13.

3.4. Testing of the proposed aDA on double NAO system

Following the successful implementation of the aDA method in single-robot navigation, the next area of investigation was to test the method's usability in a multi-humanoid environment. For this purpose, two humanoids, namely NAO robots N1 and N2, were added to the same environment, and their ability to perform simultaneous and independent tasks was tested. Each of these robots was given a unique start point, S1a and S2, and destination points, T1 and T2, to navigate through the environment, avoiding all obstacles.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

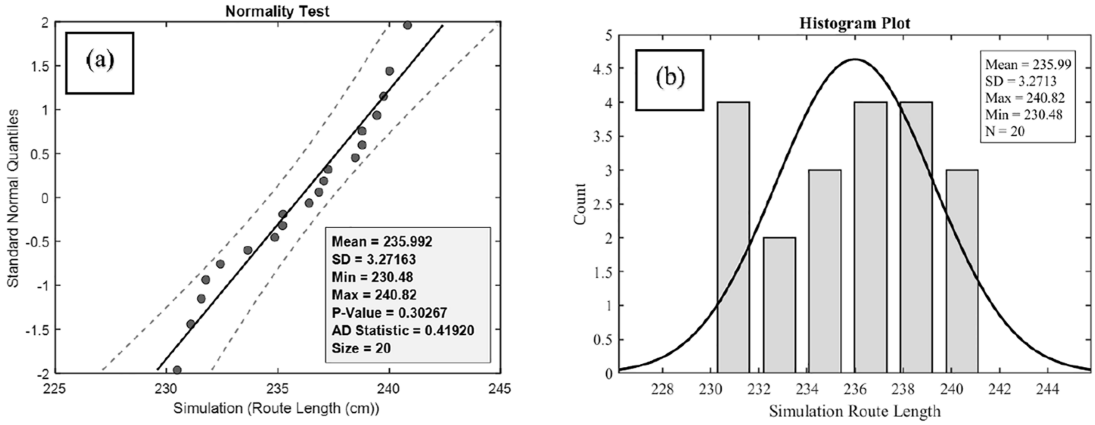


Figure 10. *The (a) normality test and (b) histogram analysis of single NAO movement in a simulation environment based on route length.*

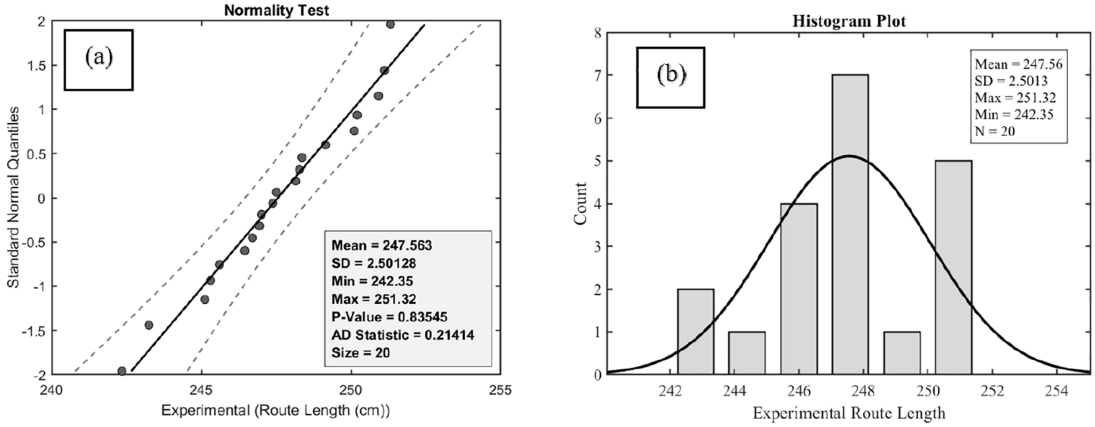


Figure 11. *(a) The normality test and (b) the histogram analysis of a single NAO robot's movement in an experimental setup, evaluated in terms of route length.*

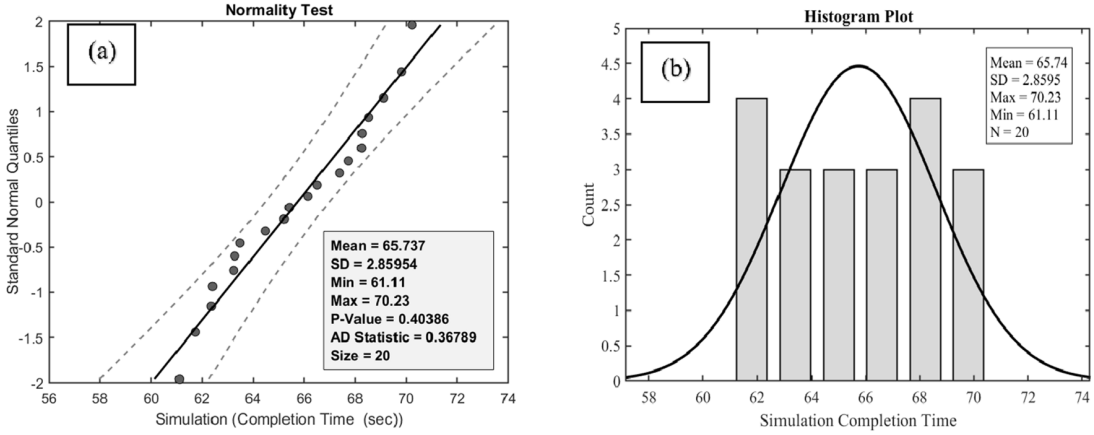


Figure 12. *(a) The normality test and (b) the histogram analysis of a single NAO robot's movement in a simulation setup, evaluated in terms of completion time.*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574724103865

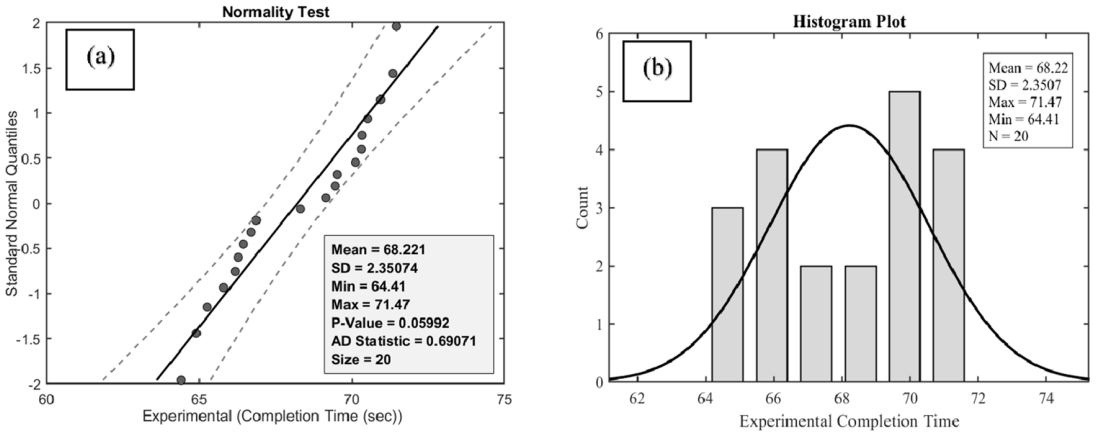


Figure 13. *(a) The normality test and (b) the histogram analysis of a single NAO robot's movement in an experimental setup, evaluated in terms of completion time.*

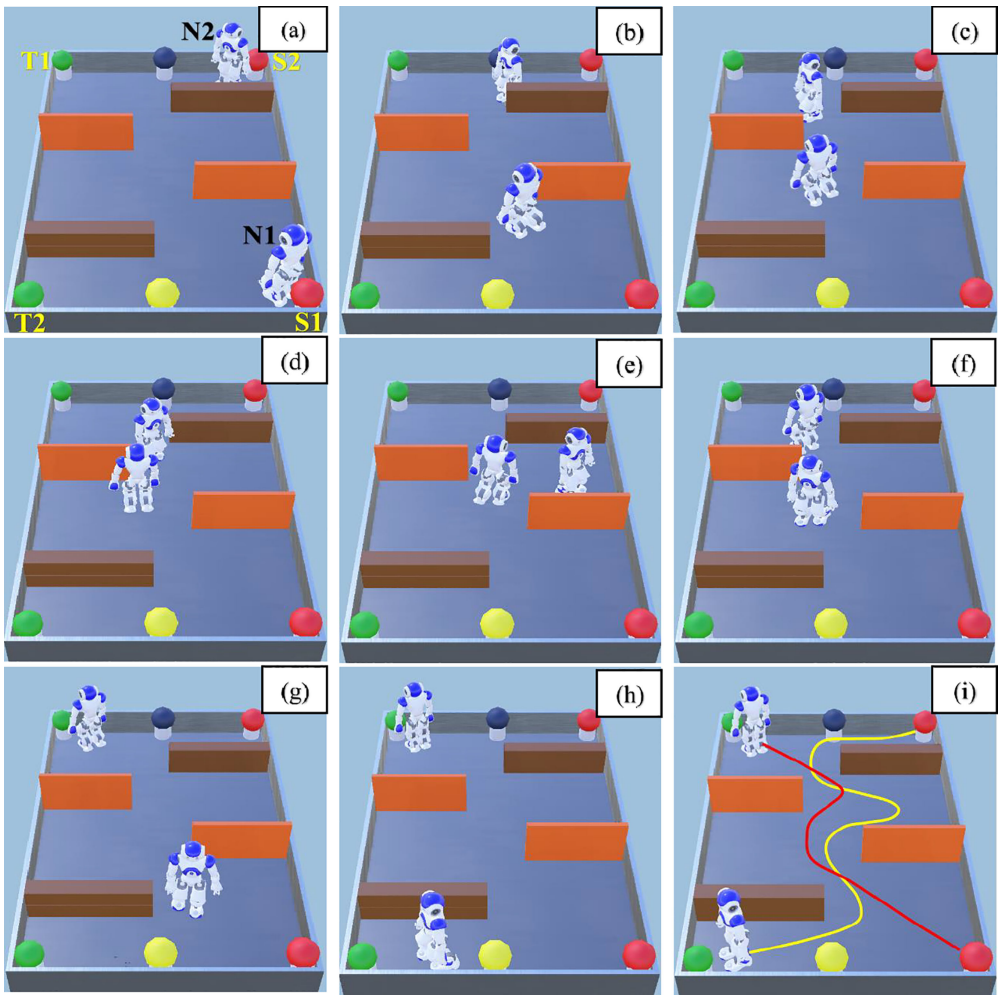


Figure 14. *Simulation result showing the path of a double NAO robot with the proposed aDA algorithm.*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

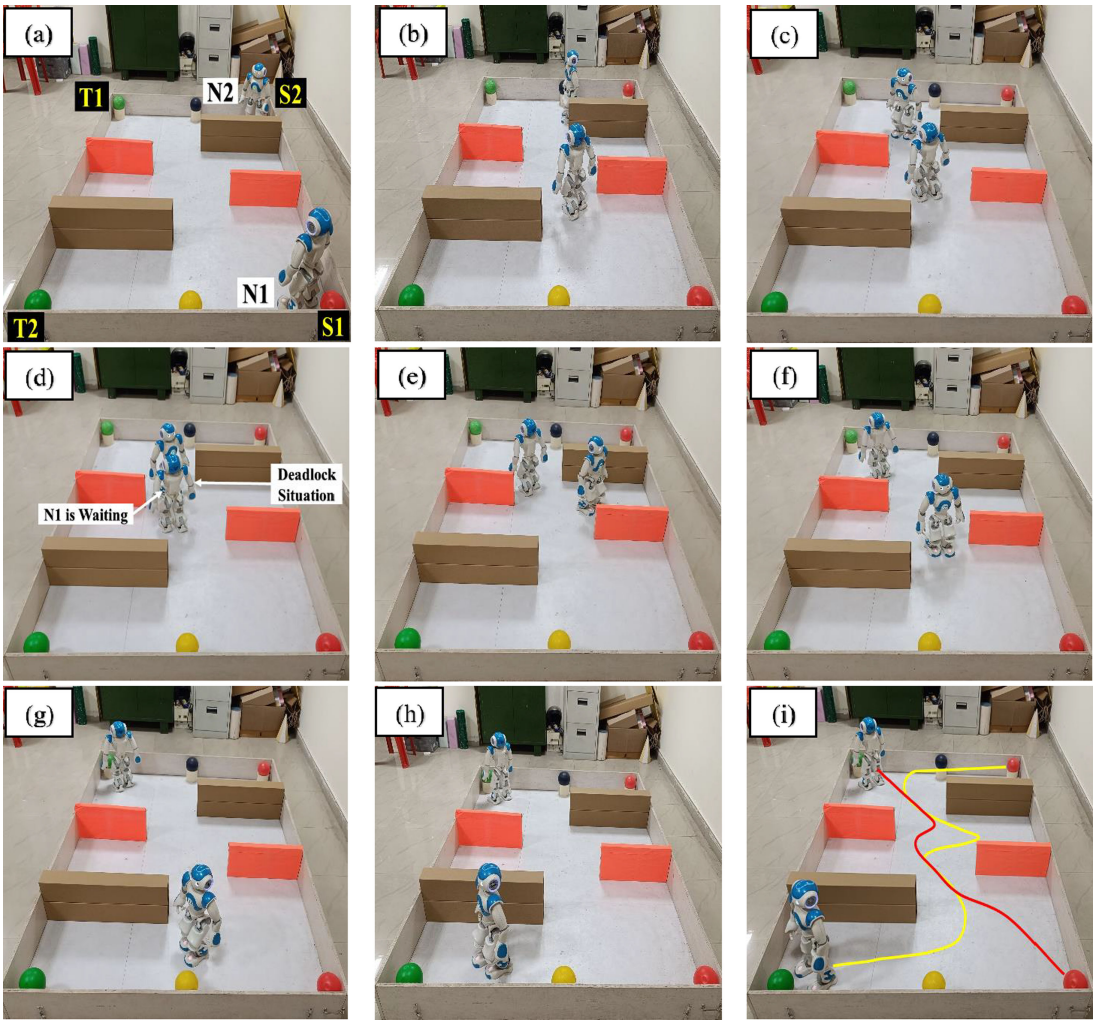


Figure 15. *Experimental result showing the path of a double NAO robot with the proposed aDA algorithm.*

To maintain a level of consistency between the simulation and the actual test scenarios, the complex terrain with its irregularly shaped obstacles and points of navigation was created both in the WEBOT simulation tool and inside the lab test environment. The simulated trajectories of N1 and N2 are depicted in Figure 14, and the corresponding experimental motion paths observed during physical trials are shown in Figure 15.

The normality test and histogram curve for the simulation and experimental results obtained by using the proposed aDA on double NAO both in simulation and experimental environments are shown in Figures 16–19.

Both robots were able to move through the complex terrain without difficulty, and the corresponding path lengths and travel times from the simulation and physical trials are summarized in Tables V and VI. It is expected that there will always be a certain level of discrepancy between results obtained in simulation and real environments due to factors such as environmental noise, surface friction, floor irregularities, and communication delays. However, as can be observed in Tables V and VI, in spite of

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

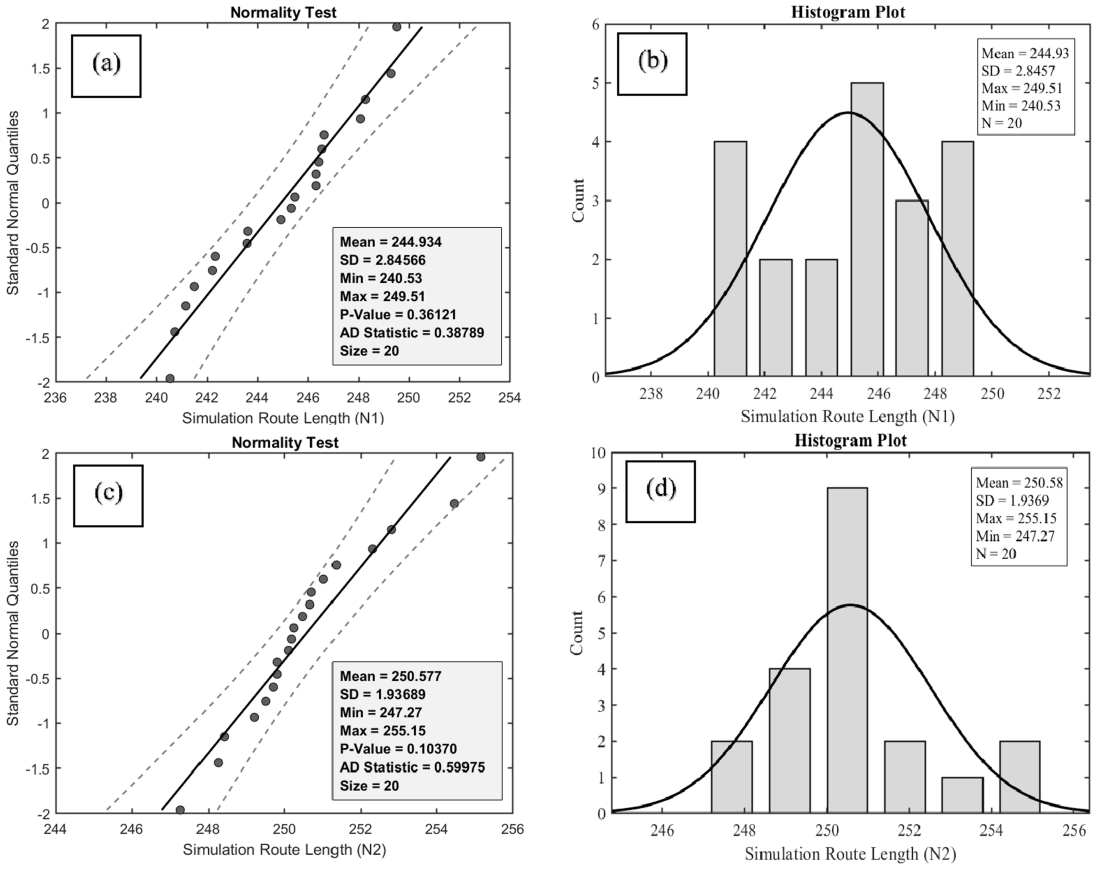


Figure 16. *Normality test and histogram analysis of multiple NAO (N1 & N2) movements in a simulation environment based on route length.*

these effects, there is a strong similarity in the results obtained by simulation and experiments, as the difference is always less than 5%. This is a strong validation of the reliability of the proposed method.

4. Comparative review of existing approaches

The efficacy of the proposed method in terms of minimum distance traveled was first tested by comparing it with an already existing method under similar testing conditions. To strengthen the validation of the proposed method, a comparison was also made with another reliable method, namely GA-PSO [38] by Huang et al., under similar testing conditions as shown in [38]. The testing environment was set up with static obstacles and specific start and goal positions, as shown in Figure 20(a) and (b) illustrates the path generated using the proposed aDA technique. It is evident that the path generated using aDA is shorter than that generated using the GA-PSO technique. The convergence curves for both methods are shown in Figure 21. The aDA algorithm is quite efficient at finding the optimal cost, reducing the number of iterations. It achieves faster convergence by reducing redundant computations while maintaining solution accuracy.

To facilitate a fair and unbiased comparison, the experimental setup was again replicated in a new scenario as presented by He et al. [39] with the same configuration of obstacles along with identical start and goal coordinates. The robot starts at (14, 8) cm and is channeled toward the goal point (14, 18) cm as shown in Figure 22. The robot was able to navigate the environment with all obstacles avoided

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

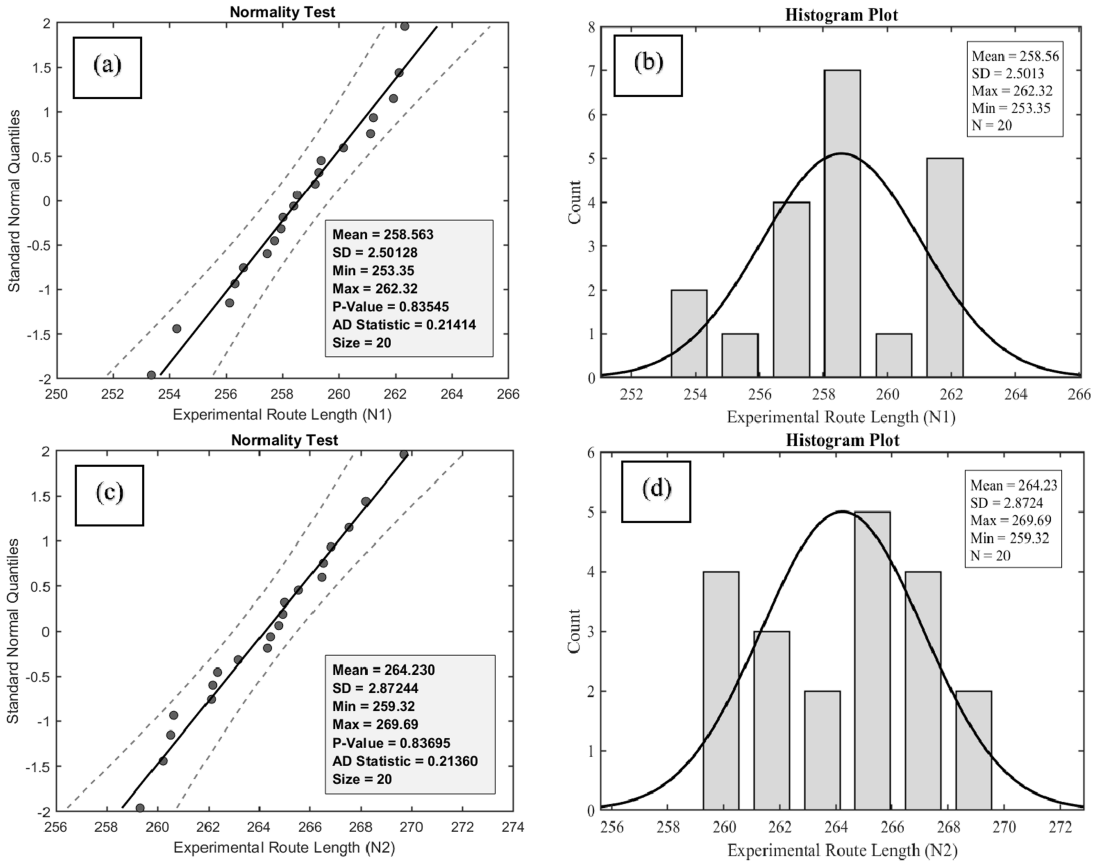


Figure 17. *Normality test and histogram analysis of multiple NAO (N1 & N2) movements in an experimental environment based on route length.*

using the proposed aDA method. The route obtained using the proposed aDA is shown in Figure 22. Examining the path lengths of both techniques, as shown in Table VII, indicates that the proposed aDA technique reduces the route distance by roughly 10.24% under the same environmental setup, which highlights the enhanced performance and effectiveness of the proposed aDA controller.

When the above environments are used to test the performance of the aDA approach, the paths turn out to be significantly shorter compared to the performance of the benchmark methods, such as FNN and GA-PSO. The above performance gains further confirm the optimized performance of the proposed aDA approach while ensuring reduced computational complexity.

5. Conclusion

The proposed adaptive aDA algorithm effectively achieves optimal solutions within a reduced computation time while preventing the humanoid robot from getting stuck in local minima. This contributes to lowering computational complexity and improving overall optimization performance. The aDA-based controller was successfully designed and validated for both single and multiple NAO humanoid robots operating in cluttered environments, demonstrating reliable and satisfactory performance. A total of 100 experimental runs were conducted to train the adaptive controller, enabling it to generate optimal

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

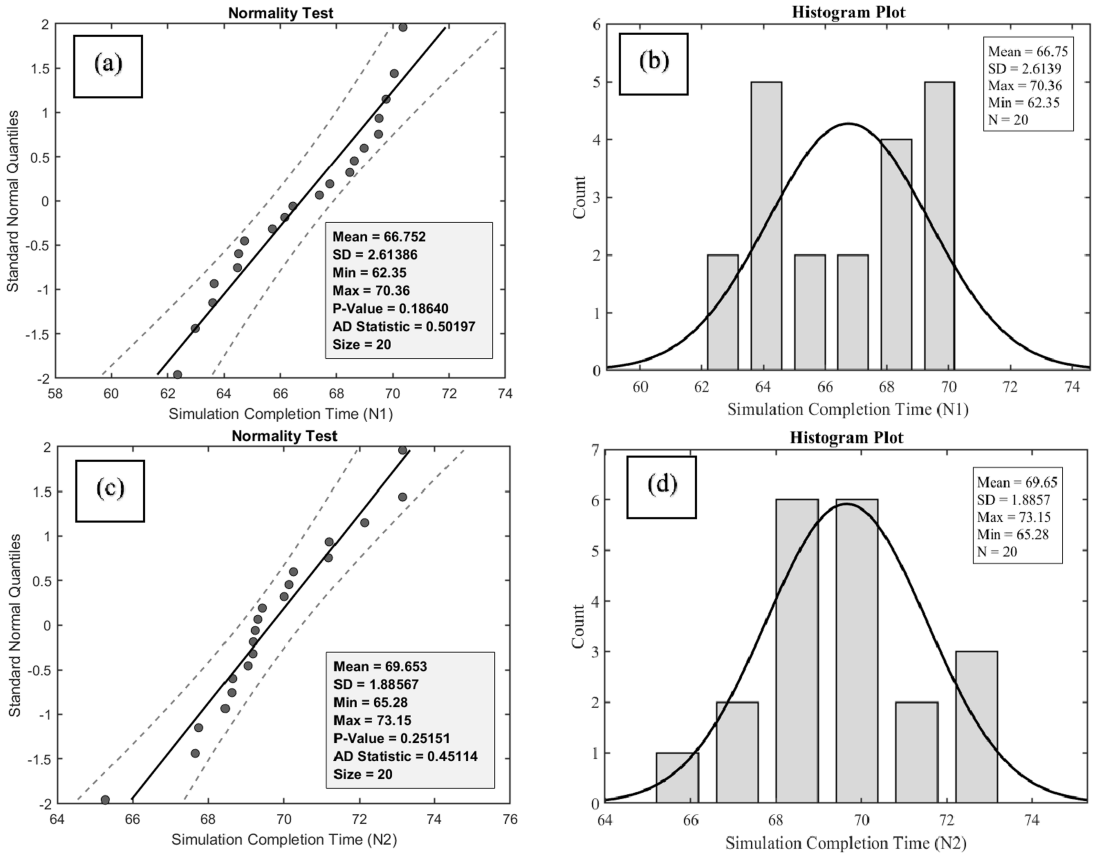


Figure 18. *Normality test and histogram analysis of multiple NAO (N1 & N2) movements in a simulation environment based on completion time.*

paths while avoiding static obstacles. In addition, the system is capable of handling dynamic obstacles. To address potential deadlock situations and avoid collisions among multiple robots, a dining philosopher-based controller was integrated with the aDA approach.

The evaluation of the proposed method was carried out using essential performance indicators such as obstacle avoidance capability, CT, and RL. Statistical analyses, including normality testing and histogram evaluation, were performed on both simulation and experimental datasets to verify the consistency and robustness of the results. The observed variation between simulation and experimental outcomes remained within an acceptable range of approximately 5%. Comparative analysis with conventional navigation methods highlights the superiority of the proposed controller, achieving an improvement of 9.07% in path length. Furthermore, when tested across different environments and benchmarked against existing approaches, the method demonstrated an average enhancement of 10.24% in terms of RL between the start and goal positions.

Overall, the results confirm that the proposed navigation strategy provides a more accurate, efficient, and robust solution for path planning in both single- and multi-humanoid robotic systems operating in complex, cluttered terrains. Future work may focus on integrating this approach with other optimization techniques and extending its application to uneven and more challenging environments for further performance enhancement.

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

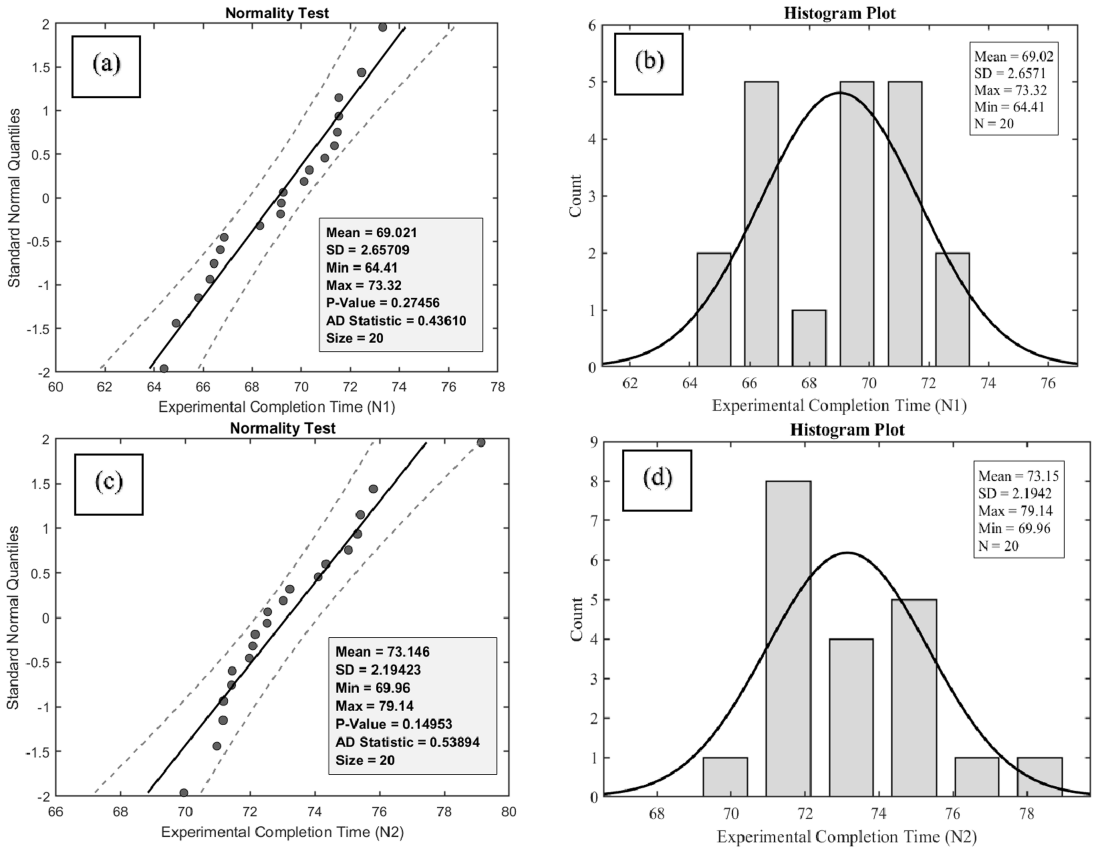


Figure 19. *Normality test and histogram analysis of multiple NAO (N1 & N2) movements in an experimental environment based on completion time.*

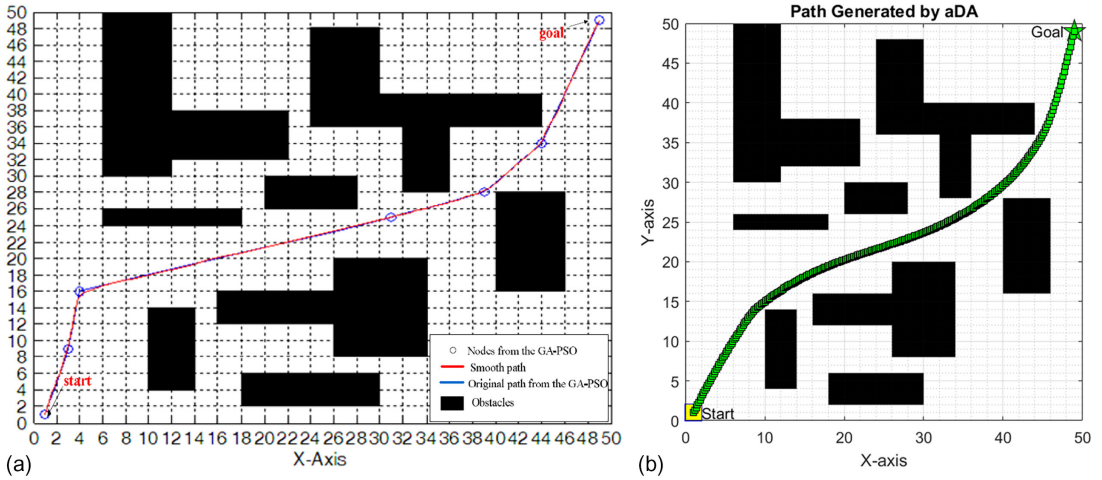


Figure 20. *Trajectory comparison between the (a) GA-PSO technique used in Ref. [38] and (b) proposed aDA algorithm.*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Table V. *Performance comparison of simulation and real-world tests for a multiple NAO humanoid with the aDA approach, measured by route length (cm).*

	Simulation route length (in cm)		Experimental route length (in cm)		Deviation (in %)	
Sl. No.	**N1**	**N2**	**N1**	**N2**	**N1**	**N2**
1	241.15	248.26	254.25	262.11	5.15	5.28
2	242.32	249.21	253.35	259.32	4.35	3.90
3	246.54	251.36	261.11	264.99	5.58	5.14
4	249.28	250.71	257.71	260.21	3.27	3.65
5	242.21	249.51	256.31	262.16	5.50	4.83
6	246.63	252.31	260.15	264.31	5.20	4.54
7	243.58	251.03	258.39	265.54	5.73	5.46
8	244.92	250.48	258.01	264.92	5.07	5.45
9	240.53	255.15	262.12	267.53	8.24	4.63
10	248.27	250.25	257.45	260.63	3.57	3.98
11	245.47	249.71	259.36	264.76	5.36	5.68
12	249.51	250.67	257.94	260.51	3.27	3.78
13	240.72	254.47	259.15	269.69	7.11	5.64
14	241.49	248.42	256.12	266.45	5.71	6.77
15	246.31	250.19	259.28	262.35	5.00	4.64
16	248.07	252.81	258.51	266.81	4.04	5.25
17	245.33	250.11	256.61	268.19	4.40	6.74
18	246.31	249.81	261.21	263.17	5.70	5.08
19	243.61	247.27	261.92	264.43	6.99	6.49
20	246.42	249.81	262.32	266.51	6.06	6.27
Avg.	**244.93**	**250.58**	**258.56**	**264.23**	**5.27**	**5.16**

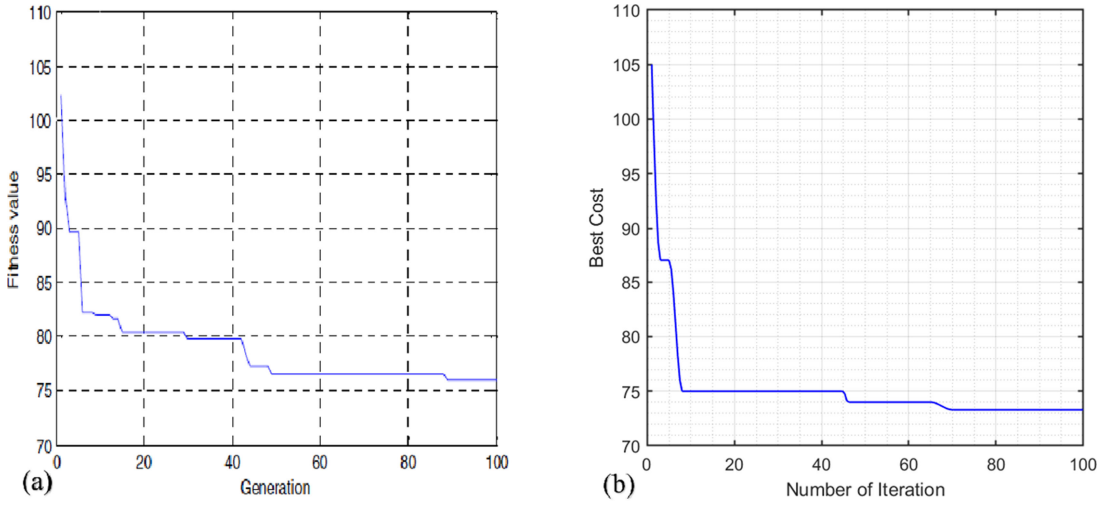


Figure 21. *Convergence curve comparison between the (a) GA-PSO technique used in [38] and (b) proposed aDA algorithm.*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Table VI. *Performance comparison of simulation and real-world tests for a multiple NAO humanoid with the aDA approach, measured by completion time (sec).*

	Simulation completion time (in sec)		Experimental completion time (in sec)		Deviation (in %)	
Sl. No.	**N1**	**N2**	**N1**	**N2**	**N1**	**N2**
1	66.45	69.31	68.32	74.11	2.74	6.48
2	67.39	71.21	69.15	75.32	2.55	5.46
3	62.98	68.64	64.41	71.17	2.22	3.55
4	69.76	73.15	73.32	75.81	4.86	3.51
5	64.73	70.26	66.45	72.09	2.59	2.54
6	70.36	73.15	71.35	79.14	1.39	7.57
7	65.72	69.18	66.87	74.35	1.72	6.95
8	68.63	69.24	70.34	71.98	2.43	3.81
9	63.65	68.45	65.81	72.16	3.28	5.14
10	64.48	67.75	66.71	70.98	3.34	4.55
11	68.47	70.14	71.47	75.41	4.20	6.99
12	69.49	71.19	71.53	73.23	2.85	2.79
13	62.35	68.62	64.91	71.18	3.94	3.60
14	63.6	65.28	66.28	69.96	4.04	6.69
15	67.76	69.19	70.12	72.55	3.37	4.63
16	68.98	69.05	72.46	72.53	4.80	4.80
17	64.52	67.66	69.26	71.43	6.84	5.28
18	66.16	69.43	69.18	71.45	4.37	2.83
19	69.51	70.01	71.53	73.03	2.82	4.14
20	70.05	72.14	70.95	75.04	1.27	3.86
Avg.	**66.75**	**69.65**	**69.02**	**73.15**	**3.28**	**4.76**

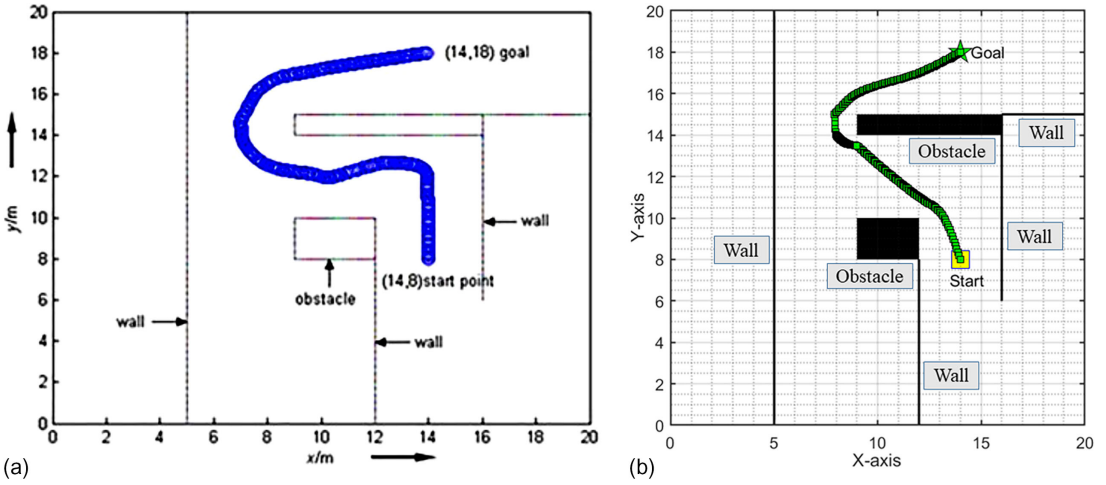


Figure 22. *Route length comparison between the (a) benchmark feed-forward neural network (FNN) technique used in ref. [39] and (b) the proposed aDA algorithm.*

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

Table VII. *Performance evaluation of the proposed aDA approach versus the FNN technique used by He et al. [39].*

Technique Used	Route Length (in cm)	Deviation (%)
FNN by He et al. [39]	16.5	10.24
By proposed aDA technique	14.81	

Author contributions. **Himansu Sekhar Dash:** Conceptualization, Methodology, MATLAB simulation, Webots Simulation, Experimental analysis, Statistical analysis, Paper original draft, and Editing, Literature review, Editing, Comparative analysis. Validation, Writing – Review and Editing. **Dayal R. Parhi:** Supervision, Conceptualization, Methodology, MATLAB simulation, Experimental analysis, Statistical analysis, Paper drafting, Literature review, Review and editing, comparative analysis.

Financial support. This research received no specific grant from any funding agency, commercial, or not-for-profit sectors.

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

Ethical standards. Not applicable.

References

[1] A. Robotics, "Nao documentation," (2012). http://www.aldebaran-robotics.com/documentation.
[2] E. W. Dijkstra, "A note on two problems in connexion with graphs," **In:** *Edsger Wybe Dijkstra: His Life, Work, and Legacy* (2022) pp. 287–290.
[3] P. E. Hart, N. J. Nilsson and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Ttans. Syst. Sci. Cybern.* **4**(2), 100–107 (1968).
[4] T. Bansal and S. Anand, "Probabilistic Roadmap Generation for Autonomous Robot Path Planning in Dynamic Environments," **In:** *International Conference on Mechanical and Energy Technologies 2023 Dec 7* (Singapore: Springer Nature Singapore) pp. 467–476.
[5] D. Szabó and E. G. Szádeczky-Kardoss, "Robotic manipulator path-planning: Cost-function approximation with fuzzy inference system," **In:** *2019 24th International Conference on Methods and Models in Automation and Robotics (MMAR) 2019 Aug 26*, pp. 259–264. IEEE.
[6] J. R. Sánchez-Ibáñez, C. J. Pérez-del-Pulgar and A. García-Cerezo, "Path planning for autonomous mobile robots: A review," *Sensors*. **21**(23), 7898 (2021).
[7] K. C. Ugwoke, N. A. Nnanna and S. E. Abdullahi, "Simulation-based review of classical, heuristic, and metaheuristic path planning algorithms," *Sci. Rep.* **15**(1), 12643 (2025).
[8] P. B. Kumar and D. R. Parhi, "Navigational analysis of a humanoid using genetic algorithm with vision assistance," *Multimed. Tools Appl.* **79**(11), 8125–8144 (2020).
[9] E. Masehian and D. Sedighizadeh, "A multi-objective PSO-based algorithm for robot path planning," **In:** *2010 IEEE International Conference on Industrial Technology 2010 Mar 14*, pp. 465–470. IEEE.
[10] J. Liu, J. Yang, H. Liu, X. Tian and M. Gao, "An improved ant colony algorithm for robot path planning," *Soft Comput.* **21**(19), 5829–5839 (2017).
[11] S. Mirjalili, "Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems," *Neural Comput. Appl.* **27**(4), 1053–1073 (2016).
[12] S. Arora and S. Singh, "Butterfly optimization algorithm: A novel approach for global optimization," *Soft Comput.* **23**(3), 715–734 (2019).
[13] B. K. Patle, A. Pandey, D. R. Parhi and A. J. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Defence Technol.* **15**(4), 582–606 (2019).
[14] J. Tu and S. X. Yang, "Genetic Algorithm Based Path Planning for a Mobile Robot," **In:** *2003 IEEE International Conference on Robotics and Automation (Cat. No. 03CH37422)*. vol. 14 (2003 Sep 14) pp. 1221–1226. IEEE.
[15] J. Wang, Y. Zhang and L. Xia, "Adaptive genetic algorithm enhancements for path planning of mobile robots," **In:** *2010 International Conference on Measuring Technology and Mechatronics Automation 2010 Mar*, vol. 13, pp. 416–419, IEEE.
[16] S. X. Yang and C. Luo, "A neural network approach to complete coverage path planning," *IEEE Trans. Syst.* **34**(1), 718–724 (2004).
[17] J. Wang, J. Liu, W. Chen, W. Chi and M. Q. Meng, "Robot path planning via neural-network-driven prediction," *IEEE Trans. Artif. Intell.* **3**(3), 451–460 (2021).
[18] B. K. Patle, D. R. Parhi, A. Jagadeesh and S. K. Kashyap, "On firefly algorithm: Optimization and application in mobile robot navigation," *World J. Eng.* **14**(1), 65–76 (2017).

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865

[19] X. Chen, M. Zhou, J. Huang and Z. Luo, "Global path planning using modified firefly algorithm," **In:** *2017 International Symposium on Micro-NanoMechatronics and Human Science (MHS) 2017 Dec 3*, pp. 1–7.IEEE.
[20] S. Kumar, D. R. Parhi, A. K. Kashyap and Vikas, "Path optimization and control of mobile robot using modified cuckoo search algorithm," **In:** *Applications of Computational Methods in Manufacturing and Product Design: Select Proceedings of IPDIMS. 2020 2022 May 4* (Singapore: Springer Nature Singapore), pp. 125–133.
[21] Q. Zhang, J. Zhao, L. Pan, X. Wu, Y. Hou and X. Qi, "Optimal path planning for mobile robots in complex environments based on the gray wolf algorithm and self-powered sensors," *IEEE Sens. J.* **23**(18), 20756–20765 (2023).
[22] C. Paital, S. Kumar, M. K. Muni, D. R. Parhi and P. R. Dhal, "Navigation of a wheeled mobile robotic agent using modified grey wolf optimization controller," *Int. J. Intell. Unmann. Syst.* **11**(2), 197–213 (2023).
[23] W. Liu, B. Niu, H. Chen and Y. Zhu, "Robot path planning using bacterial foraging algorithm," *J. Comput. Theor. Nanosci.* **10**(12), 2890–2896 (2013).
[24] H. Zhou, H. Y. Cheng, Z. L. Wei, X. Zhao, A. D. Tang and L. Xie, "A hybrid butterfly optimization algorithm for numerical optimization problems," *Comput. Intell. Neurosci.* **2021**(1), 7981670 (2021).
[25] G. A. Ali, H. Abubakar, S. A. Alzaeemi, A. H. Almawgani, A. Sulaiman and K. G. Tay, "Artificial dragonfly algorithm in the Hopfield neural network for optimal Exact Boolean k satisfiability representation," *PLoS One*. **18**(9), e0286874 (2023).
[26] J. Zhao, D. Zhang, Q. He and L. Li, "A hybrid-strategy-improved dragonfly algorithm for the parameter identification of an SDM," *Sustainability-BASEL*. **15**(15), 11791 (2023).
[27] B. Patel, V. Dubey, S. Barde and N. Sharma, "Optimum path planning using dragonfly-fuzzy hybrid controller for autonomous vehicle," *Eng*. **5**(1), 246–265 (2024).
[28] K. Katona, H. A. Neamah and P. Korondi, "Obstacle avoidance and path planning methods for autonomous navigation of mobile robot," *Sensors*. **24**(11), 3573 (2024).
[29] S. Agrawal, B. K. Patle and S. Sanap, "A systematic review on metaheuristic approaches for autonomous path planning of unmanned aerial vehicles," *Drone Syst. Appl.* **6**(12), 1–28 (2024).
[30] A. K. Kashyap, D. R. Parhi, M. K. Muni and K. K. Pandey, "A hybrid technique for path planning of humanoid robot NAO in static and dynamic terrains," *Appl. Soft Comput.* **96**, 106581 (2020).
[31] H. S. Dash, D. R. Parhi, M. K. Muni and P. Das, "An improved butterfly optimization algorithm-based path navigation of humanoid robots in an unfamiliar setting," *Robotica*. **43**(6), 2273–2303 (2025).
[32] A. V. Chechkin, R. Metzler, J. Klafter and V. Y. Gonchar, "Introduction to the theory of Lévy flights," **In:** *Anomalous Transport: Foundations and Applications*, pp.129–162 (2008).
[33] Y. P. Liu, A. Sah and M. Sawhney, "A Gaussian fixed point random walk. arXiv preprint arXiv:2104.07009 (2021).
[34] J. Nauta, Y. Khaluf and P. Simoens, "Using the Ornstein-Uhlenbeck process for random exploration," **In:** *COMPLEXIS*, pp. 59–66 (2019),
[35] E. W. Dijkstra, "Hierarchical ordering of sequential processes," *Acta Inform.* **1**(2), 115–138 (1971).
[36] A. K. Kashyap, D. R. Parhi and P. B. Kumar, "Route outlining of humanoid robot on flat surface using MFO aided artificial potential field approach," *Proc. Inst. Mech. Eng. Part B: J. Eng. Manuf.* **236**(6–7), 758–769 (2022).
[37] O. Michel, "Symbiosis between virtual and real mobile robots." **In:** *International Conference on Virtual Worlds* (Berlin, Heidelberg, Berlin Heidelberg: Springer) pp. 254–263 (1998).
[38] H. C. Huang and C. C. Tsai, "Global path planning for autonomous robot navigation using hybrid metaheuristic GA-PSO algorithm," **In:** *InSICE Annual Conference*, vol. 13, pp. 1338–1343 (2011). IEEE.
[39] K. He, Y. Gao and H. Sun, "A fuzzy neural network based on TS model for mobile robots to avoid obstacles", **In:** *International Conference on Intelligent Robotics and Applications 2008 Oct 15* (Berlin, Heidelberg, Berlin Heidelberg: Springer) pp. 1127–1134.
[40] C. Sahu, D. R. Parhi, M. K. Muni and S. S. Kamat, "Tailoring the motion planning of humanoids in complex arena: A regression-firefly-based approach," *Robotica*. **42**(3), 757–774 (2024).
[41] D. R. Parhi, "Single and multiple humanoid path planning using Hill valley approach applied to gravitational drift in Gravitational search algorithm," *Robotica*. **41**(12), 3627–3648 (2023).
[42] A. K. Kashyap, D. R. Parhi and V. Kumar, "Navigation for multi-humanoid using MFO-aided reinforcement learning approach," *Robotica*. **41**(1), 346–369 (2023).

Cite this article: H. S. Dash and D. R. Parhi (2026). "Adaptive Dragonfly Algorithm for AI-enabled path planning of humanoid robots in obstacle-constrained environments", Robotica **44**, 1990–2015. https://doi.org/10.1017/S0263574726103865

Downloaded from https://www.cambridge.org/core. UNIVERSIDAD AUTONOMA METROPOLITANA, on 16 Nov 2026 at 22:57:22, subject to the Cambridge Core terms of use, available at https://www.cambridge.org/core/terms. https://doi.org/10.1017/S0263574726103865