

# Robotic path planning for NDT using neuroevolution of augmenting topologies

Mengyuan Zhang, M Sutcliffe, QingPing Yang and I Nicholson

*Path planning in ultrasonic non-destructive testing (NDT) must simultaneously optimise coverage integrity and motion efficiency under complex geometric constraints. However, traditional methods primarily focus on shortest-path computation, which struggles to adapt to complex topologies, particularly in inspection scenarios where surface continuity and movement costs vary. These methods often fail to balance path quality and obstacle avoidance. In this study, a neuroevolutionary framework is introduced that combines graph-based modelling with the dynamic topology evolution of the neuroevolution of augmenting topologies (NEAT) algorithm, aiming to improve adaptability in complex inspection environments. The method models the inspection environment as a weighted graph and employs neural networks to optimise the path generation strategy that ensures complete coverage while minimising redundant motion. The experimental results demonstrate that the proposed method achieves 100% coverage while improving path smoothness, reducing sharp turns and enhancing obstacle avoidance. Overall, it outperforms A\*, Dijkstra and a multi-layer perceptron (MLP).*

## 1. Introduction

Ultrasonic non-destructive testing (NDT) is an essential technique for evaluating material properties and detecting defects without causing damage to the inspected object<sup>[1]</sup>. With the rise of industrial automation, NDT has found growing use in robotic inspection systems, especially for complex and large-scale structures such as turbine blades, car parts and pipelines. The completeness of inspection coverage is crucial for ensuring the accuracy of ultrasonic NDT. In automated ultrasonic NDT, robots must follow optimised paths to ensure full coverage while minimising inspection time and operational costs<sup>[2]</sup>.

Many traditional path planning methods, such as those based on shortest path algorithms, are mainly designed to determine the optimal path from the start to the end. The travelling salesman problem (TSP) and its related algorithms are able to achieve full coverage of inspection points and effectively minimise total path lengths in static<sup>[3]</sup>, structured environments, similar to shortest path algorithms such as A\* and Dijkstra<sup>[4]</sup>. However, in practical ultrasonic NDT applications, obstacles (for example welded support structures) and dynamic geometric constraints are often present in the inspection region. For example, in the inspection of aerospace composite components with porous structures, the 'shortest path' generated by the TSP may directly traverse the physical obstacle region, increasing the risk of probe collision. In addition, TSP-based methods primarily optimise overall path length to ensure that all inspection points are visited only once but often fail to accommodate dynamic environmental or inspection constraints. Thus, achieving a balance between complete inspection coverage and adaptability to dynamic constraints remains a challenge in complex inspection tasks.

To address the challenges, this study proposes a novel path planning method that integrates neuroevolution and graph theory. This method establishes a collaborative mechanism that is capable of sensing the dynamic environment and performing multi-objective optimisation (MOO). The inspection region is modelled as a non-uniform graph network with adaptive node density, where edge weights define the cost of traversal between nodes. Building on this, neuroevolution of augmenting topologies (NEAT) is introduced

to enable autonomous adaptation of the neural network topology throughout the detection process. In obstacle-dense regions, the algorithm enhances local path optimisation by increasing the number of hidden nodes, while in flat regions, redundant connections are pruned to improve computational efficiency. This closed-loop 'graph theory representation-evolutionary optimisation' framework allows the algorithm to preserve the full-coverage property of the TSP while incorporating dynamic obstacle avoidance and multi-constraint joint optimisation capabilities.

## 2. Theoretical background

Path planning research for ultrasonic non-destructive testing has undergone a paradigm shift from geometry-driven to physics-informed approaches<sup>[5]</sup>. Early studies focused on the optimisation of detection coverage for regular geometries, where raster scan and spiral path strategies were widely adopted due to their high computational efficiency<sup>[6]</sup>. However, these methods exhibited limitations in free-form surface detection, especially in regions with abrupt curvature changes, often leading to uneven detection coverage or path redundancy<sup>[7]</sup>.

To overcome these limitations, researchers have introduced computer-aided manufacturing (CAM) techniques to optimise the robotic scanning trajectory in ultrasonic NDT systems. For example, the MATLAB trajectory optimisation toolkit developed by Mineo *et al*<sup>[8]</sup> reduces manual intervention based on structured environment modelling and improves the efficiency of

● Submitted 27.03.25 / Accepted 27.08.25

Mengyuan Zhang and Dr QingPing Yang are with the College of Engineering, Design and Physical Sciences, Department of Mechanical and Aerospace Engineering, Brunel University of London, UK

Mengyuan Zhang is also with The Welding Institute, NDT.

Dr Mark Sutcliffe is with TWI, VAL.

Dr Ian Nicholson is with TWI Technology Centre, NDT.

inspection paths. Hu *et al*<sup>[9]</sup> further utilised CATIA to compute the inspection points for complex geometries, which is particularly suitable for scenarios where robots hold components or ultrasonic transducers. The method relies on computer-aided design (CAD) models to generate the detection path, effectively reducing the need for manual adjustment and demonstrating strong applicability. With the increasing complexity of inspection scenarios, free-form surface path planning has gradually become a core direction of research. A hybrid graph-theoretic and K-dimensional tree algorithm proposed in previous work<sup>[4]</sup> enhances computational efficiency by optimising spatial indexing to automatically generate inspection paths for complex surfaces. However, the dependence of the method on high-precision point cloud data limits its applicability to field inspection. Similarly, Li *et al*<sup>[10]</sup> optimised the detection point layout based on directional clustering and used algorithms such as Lin-Kernighan and greedy search to optimise the global detection path. Although the method has demonstrated improvements in path efficiency, it remains limited in highly variable or dynamic environments.

For free-form surfaces with sharp curvature variations and discontinuities, researchers have turned to more adaptive optimisation strategies. Chen *et al*<sup>[11]</sup> introduced an enhanced cuckoo search algorithm that uses adaptive parameter tuning and dynamic neighbourhood search to improve computational efficiency. Their experiments on engine impeller inspection show that the method outperforms traditional optimisation algorithms (for example the genetic algorithm (GA) and ant colony optimisation (ACO)) in terms of path length and computational speed. On the other hand, Poole *et al*<sup>[12]</sup> proposed a multi-scale, collision-free path planning method that combines low-cost sensors to deal with noise and incomplete point clouds, which enables it to show more flexibility and adaptability in real-time NDT environments, especially for working conditions that lack complete surface data. Mineo *et al*<sup>[13]</sup> achieved simultaneous geometric and volumetric inspection for free-form parts using a CAD-driven robotic inspection system. Although efficient in structured environments with reliable CAD models, their framework requires exact CAD specifications and extensive offline calibration and alignment.

In addition to NDT-specific path planning methods, the development of general-purpose robot path planning has also provided new ideas for the field. For example, the bidirectional rapidly-exploring random tree (B-RRT) algorithm proposed by Akgun and Stilman<sup>[14]</sup> can quickly find an initial feasible solution in complex environments and improve the path quality through subsequent optimisations, making it suitable for inspection path planning for complex geometries. Han *et al*<sup>[15]</sup> further developed this approach by proposing an improved Theta\* algorithm that optimises 3D paths through obstacle buffering and trajectory smoothing to make the paths more compatible with detection requirements.

Evolutionary algorithms (for example the GA and ACO) have also been gradually applied in robotic path planning in recent years. Lim *et al*<sup>[16]</sup> used genetic algorithms to optimise bridge crack detection paths, which demonstrated superior performance in detection efficiency. Rajendran and Othman<sup>[17]</sup> compared the path planning performance of ACO and a GA in automated warehouse robots and found that ACO was more advantageous in obstacle avoidance and path optimisation. This finding suggests the potential application of ACO in NDT tasks with complex geometry and dynamic environments. In addition, the hybrid genetic-ant colony optimisation (HGACO) algorithm proposed by Xie *et al*<sup>[18]</sup> further improves the coverage of path planning when applied to large-scale, high-complexity NDT tasks through grid partitioning and subregion optimisation strategies.

Despite improvements in efficiency and coverage<sup>[4,8]</sup>, NDT-specific path planning methods still face challenges in adapting to rapidly changing and geometrically complex environments. In contrast, general-purpose robot path planning algorithms (for example RRT\*, HGACO) provide valuable insights in terms of path optimisation, computational efficiency and environmental adaptability. However, current research still faces the fundamental contradiction: geometry-driven methods struggle to incorporate multi-physics constraints, while data-driven approaches are limited by their generalisation capabilities across different inspection scenarios. Therefore, the coupled NEAT-graph theory framework proposed in this paper combines dynamic topology evolution with adaptive graph network modelling to achieve multi-objective co-optimisation while maintaining the integrity of the detection coverage, which is particularly suitable for NDT tasks involving irregular and complex geometries.

## 3. Methodology

### 3.1 Neuroevolution of augmenting topologies algorithm

The NEAT algorithm was proposed by Stanley and Miikkulainen in<sup>[19]</sup> and its core contribution lies in simultaneously optimising the topology and parameters of a neural network. Unlike traditional evolutionary algorithms that only optimise connection weights, NEAT employs a minimalist initial structure consisting of only input and output layers and gradually introduces hidden nodes and cross-layer connections via mutation operators, following an incremental process from simple to complex. Through progressive complexity increases, NEAT enables the network to self-adjust to changing task demands<sup>[20]</sup>. The key concepts of NEAT are as follows:

- **Co-evolutionary mechanism:** NEAT dynamically adjusts its network topology rather than relying on a predefined structure to effectively avoid overfitting or underfitting issues that may result from a fixed topology. In automated ultrasound inspection path planning, this feature enables the algorithm to autonomously adjust the hidden layer node density according to the curvature gradient of the surface by generating dense subnetworks in high-curvature regions to refine the path representation, while maintaining a simplified structure in flat regions to improve computational efficiency.
- **Genome encoding and innovation numbering:** NEAT tracks the evolution of network topology through an innovation ID system, where each connection gene is assigned a unique identifier to track its mutation type (for example node insertion or cross-layer connection). This design helps preserve topological compatibility during crossover and also offers a basis for species differentiation. For example, in a path planning task, obstacle avoidance connectivity genes generated for the edges of holes can be traced back to specific mutation events, ensuring stable inheritance of dominant structures.
- **Speciation preservation mechanism:** by calculating the topological dissimilarity  $\delta = \frac{c_1 E + c_2 D}{N}$  (where E is the number of redundant genes, D is the number of high-dispersity genes and N is the genome length), NEAT classifies the populations using a species classification strategy. Selection and crossover occur independently within each species to prevent well-adapted simple structures from prematurely suppressing the emergence of complex ones in early evolution. This diversity-preserving mechanism is particularly relevant for complex weld detection; different species separately explore global coverage optimisation

and local avoidance optimisation and eventually converge on the optimal solution through elite migration.

- **Multi-objective adaptive evolution:** in the MOO task, each objective corresponds to an independent fitness function and different objectives may conflict with each other, for example the shortest path and the safety of obstacle avoidance in path planning. NEAT assesses individual performance on each objective task during evolution and evaluates the performance of an individual on each objective task based on fitness sharing and Pareto optimality. It iteratively selects and propagates individuals according to these principles to achieve multi-objective optimisation.

### 3.1.1 Genome encoding

NEAT enables the co-evolution of neural network topology and parameters through a linked-list encoding system. The genome consists of node genes and connection genes: node genes define the functional attributes of neurons in the input, hidden and output layers, and connection genes define information transfer between neurons through weight values and the indices of start and target nodes, as shown in Figure 1. Each connectivity gene carries a unique innovation number, which precisely records the evolutionary history of the mutation event (for example the connectivity mutation of the  $n$ th generation of the  $k$ th individual), providing topological compatibility guarantees for gene alignment in crossover operations. This encoding mechanism allows the network structure to evolve with the complexity of the task, initially containing only the necessary connections to reduce computational overhead and gradually introducing hidden nodes and cross-layer connections as the evolution progresses to form a multi-scale topology adapted to the characteristics of the detection environment.

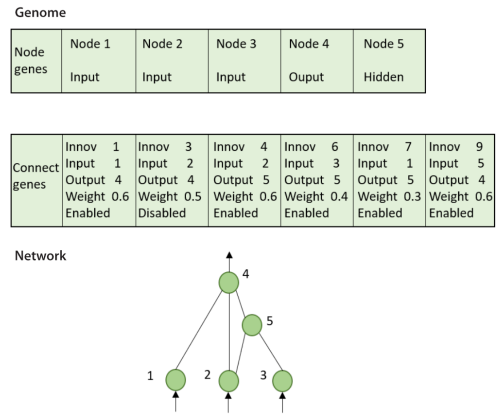


Figure 1. Genome encoding

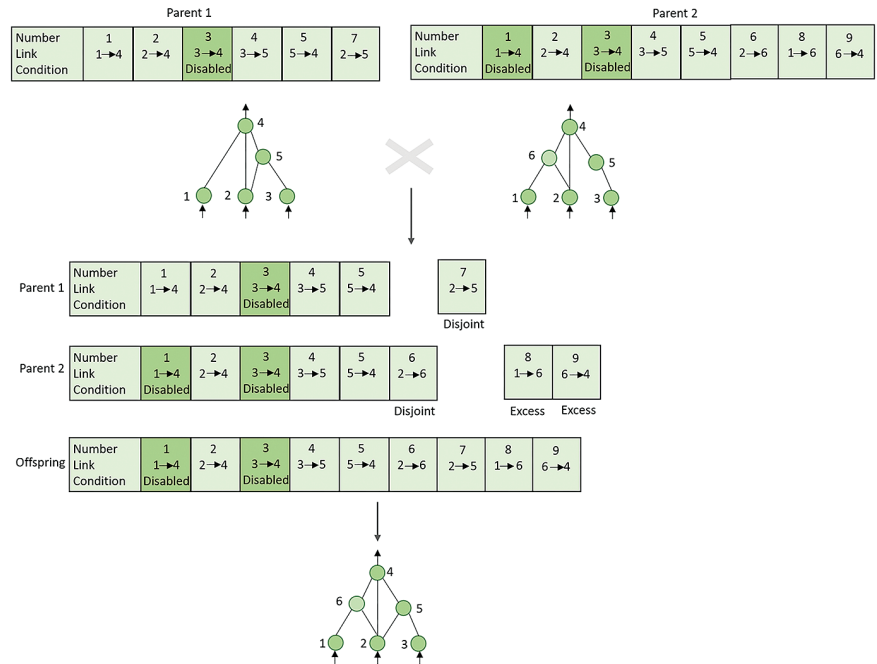


Figure 2. NEAT crossover operations

### 3.1.2 Crossover operations

In the crossover operation, NEAT achieves precise alignment of parental genomes based on innovation numbers: for genes sharing innovation numbers, the connection weights of the dominant parent are selected by a roulette wheel strategy; non-matching genes are retained or discarded based on the fitness priority of the parental individuals, as shown in Figure 2. This mechanism both preserves dominant genes (for example optimal avoidance paths in regions of high curvature) and maintains population diversity through topological innovations (for example new inter-detection point connections). In particular, disabled genes remain dormant in offspring but may be reactivated in subsequent evolution in response to environmental changes, providing a potential adaptive reserve for dynamic detection tasks.

### 3.1.3 Mutation operations

Mutation operations increase network complexity through topological reconfiguration. Connection mutation creates new connections between randomly selected node pairs with a probability of  $p_c = 0.5$ , extending the information transfer path; node mutation inserts hidden nodes into existing connections with probability  $p_n = 0.2$ , splitting an existing connection into two weighted paths, as shown in Figures 3 and 4. In addition, a node

is deleted with a probability of 0.5 to prevent over-complexity of the network; while connection deletion is not explicitly listed in the table, it can typically be assigned a similar probability. For example, a connection between nodes (1,4) can be split into (1,6) and (6,4), where the original connection is disabled but still stored in the genome to keep topological evolution reversible. This gradual complexification strategy ensures that the network structure always evolves in tandem with the physical constraints of the detection task (for example curvature gradient of the surface, density of obstacle distributions), which provides the theoretical underpinning of the present study of the NEAT-graph hybrid algorithm. The graph network ensures the global coverage, while the NEAT variant optimises the local path details, and the two work together to achieve multi-objective optimality.

## 3.2 Graph theory

The path planning problem in this study is addressed by modelling a non-uniform weighted graph (NWG), which is mathematically represented as:

$$G = (V, E) \dots \dots \dots (1)$$

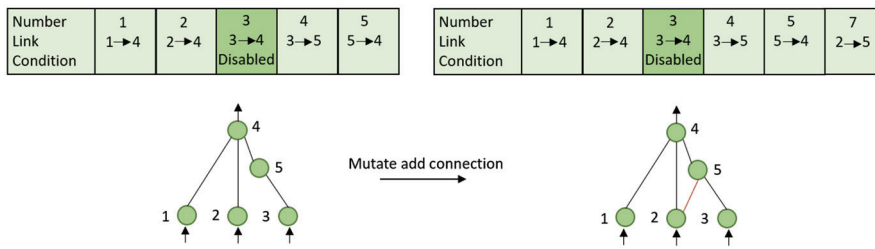


Figure 3. Connection mutation

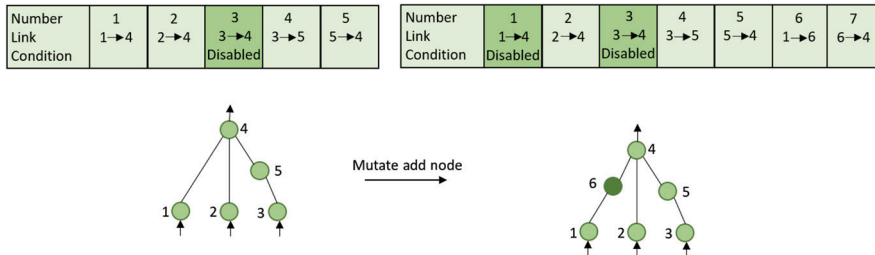


Figure 4. Node mutation

where  $V$  is the set of detection points and  $E$  represents the set of feasible paths between them. Each edge  $(v_i, v_j) \in E$  is assigned a weight  $w_{ij}$ , which quantifies the movement cost between two detection points.

The weight  $w_{ij}$  is determined by both the Euclidean distance between detection points and additional environmental factors such as surface curvature and obstacle distribution. It is expressed as:

$$w_{ij} = d(v_i, v_j) + C_{ij} \dots \dots \dots (2)$$

### 3.2.1 Neighbour identification and graph construction

When constructing a graph network topology, it is necessary to define the neighbourhood relationships between nodes. In this study, a fixed-radius neighbourhood selection method is employed, where the neighbour distance threshold  $s$  is defined as:

$$s = \alpha \cdot d, \quad \alpha \in [1.2, 1.5] \dots \dots \dots (3)$$

where  $d$  is the predefined base distance for adjacency determination and  $\alpha$  is an expansion factor used to adjust the adjacency range, typically set within the range of 1.2-1.5.

Algorithm 1 performs neighbourhood node selection as follows:

#### Algorithm 1. Neighbourhood node selection algorithm

```

Function GET_NEIGHBOURS (current_node, nodes, spacing)
1. Nb ← ∅ // Initialise set of neighbouring nodes
2. For each node ∈ nodes do
3.   Δx ← node.x - current_node.x
4.   Δy ← node.y - current_node.y
5.   dist ← √(Δx² + Δy²) // Calculate Euclidean distance
6.   If dist ≤ spacing and dist > 0 then
7.     Nb ← Nb ∪ {node} // Add neighbouring nodes while
       excluding self-connections
8.   End If
9. End For
10. Return Nb
End Function

```

### 3.2.2 Integration of NEAT and graph theory in path planning

This study proposes an autonomous path planning framework based on the integration of the NEAT algorithm and graph theory to achieve topology-adaptive decision-making in dynamic environments through a neural evolution mechanism. The system models the detection region as a weighted graph  $G = (V, E, W)$ , where the vertex set  $V$  corresponds to the environmental detection points, the edge set  $E$  characterises the feasible paths and the weight function  $W: E \rightarrow R^+$  quantifies the movement cost. NEAT optimises path weights while dynamically selecting movement paths based on neural network decisions.

As shown in Figure 5, the original environment data (coordinates, features) are transformed into a discrete vertex-edge topology. NEAT adopts a feed-forward neural network (FNN) as the decision-making model and optimises the network structure during evolution. It dynamically adjusts hidden nodes and connections based on the current state of detected points, generating optimal path decisions while satisfying two constraints: to ensure full coverage of all target detection points; and to minimise the total movement cost by optimising accumulated edge weights.

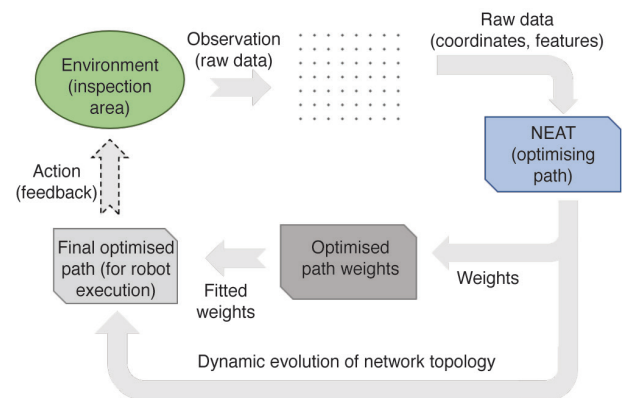


Figure 5. Framework of NEAT algorithm application in path planning

NEAT optimises path parameters through adaptive evolutionary mechanisms and incorporates species formation strategies to maintain population diversity and prevent evolution from falling into local optima. When the environment changes, the system performs dynamic closed-loop optimisation by updating path weights in real time and feeding the optimisation results back to the path execution unit<sup>[21]</sup>.

To illustrate the evolution of the NEAT algorithm, Figure 6 demonstrates the progressive development of the network structure:

1. The algorithm starts with a basic network configuration.
2. New connections are added, increasing network complexity.
3. Weights are optimised to improve path selection.
4. The network becomes fully connected, enabling efficient handling of complex environments.

The state space of the agent is defined as  $S_t = (v_{curr}, H_t)$ , where  $v_{curr} \in V$  represents the current vertex and  $H_t \subseteq V$  is the set of

previously visited vertices. The state encoder maps  $S_t$  to a feature vector  $\phi_t$ , which is then processed by the NEAT-evolved policy function  $\pi_\theta: \phi_t \rightarrow \Delta^{|V|}$  to generate the probability distribution over the target vertices. The agent selects the next action as  $a_t = \arg\max P_t$  or performs stochastic exploration based on the  $\epsilon$ -greedy policy to optimise path selection.

After the agent completes the full traversal, the total path cost derived from edge weights  $W(v_i, v_j)$  serves as feedback for the fitness evaluation module of NEAT, guiding the neural network evolution to improve the path planning performance. The specific objectives of path optimisation are detailed in the fitness function section.

#### Algorithm 2. Autonomous agent path planning with NEAT

**Function:** run\_agent(graph G, start\_node vs, NEAT\_model)

```

1. Set v_curr ← vs    // Initialise agent at start node
2. Set H_t ← {v_curr} // Initialise visited node set
3. While not all targets visited do
    a.  $\phi_t \leftarrow \text{EncodeState}(v_{\text{curr}}, H_t)$  // State encoding
    b.  $P_t \leftarrow \text{NEAT\_model.forward}(\phi_t)$  // Neural network inference
    c.  $v_{\text{next}} \leftarrow \arg\max P_t$  // Select next node following  $\epsilon$ -greedy strategy
    d. If  $v_{\text{next}} \notin H_t$  then
        i.  $\text{MoveAgent}(v_{\text{curr}} \rightarrow v_{\text{next}})$  // Execute movement
        ii.  $H_t \leftarrow H_t \cup \{v_{\text{next}}\}$  // Update visited nodes
    End If
    e.  $v_{\text{curr}} \leftarrow v_{\text{next}}$  // Update current node
End While

4. EvaluateFitness(NEAT_model, agent, points) // Compute fitness after full traversal

```

### 3.3 Fitness function for 2D and 3D path optimisation

The fitness function  $F_{\text{total}}$  is formulated to assess the effectiveness of the planned paths based on multiple criteria: ensuring that all target

points are visited; penalising redundant visits, excessive path length and abrupt directional changes; and minimising sharp turns in 2D and sudden altitude variations in 3D.

#### 3.3.1 Fitness function in 2D path planning

In the 2D scenario, smoothness is primarily evaluated based on directional continuity to prevent abrupt turns:

$$F_{2D} = R_{\text{coverage}} - P_{\text{redundancy}} - P_{\text{length}} - w_{\text{smooth}} \cdot P_{\text{turn}} + B_{\text{full}} \dots (4)$$

where the components are defined as follows:

- Coverage reward:

$$R_{\text{coverage}} = w_{\text{coverage}} \times N_{\text{visited}} \dots (5)$$

- Redundancy penalty:

$$P_{\text{redundancy}} = w_{\text{redundancy}} \times N_{\text{redundant}} \dots (6)$$

- Path length penalty:

$$P_{\text{length}} = w_{\text{length}} \times \sum_{i=1}^{n-1} d(i, i+1) \dots (7)$$

- Turning penalty (to discourage sharp turns):

$$P_{\text{turn}} = w_{\text{turn}} \times \sum_{i=2}^{n-1} \left( 1 - \frac{\theta(i, i+1)}{\theta_{\text{max}}} \right) \dots (8)$$

where  $\theta(i, i+1)$  is the turning angle between two consecutive steps and  $\theta_{\text{max}}$  is the maximum allowable turning angle.

#### 3.3.2 Fitness function in 3D path planning

In 3D environments, additional constraints are introduced to minimise elevation changes and ensure the path remains smooth and aligned with the inspection surface.

$$F_{3D} = R_{\text{coverage}} - P_{\text{redundancy}} - P_{\text{length}} - w_{\text{smooth}} \cdot P_{z\text{-axis}} + B_{\text{full}} \dots (9)$$

where the smoothness penalty  $P_{z\text{-axis}}$  is calculated as follows:

$$P_{z\text{-axis}} = w_z \times \sum_{i=1}^{n-1} |z_{i+1} - z_i| \dots (10)$$

In this equation:

- $|z_{i+1} - z_i|$  represents the vertical displacement between consecutive detection points; and
- $w_z$  is a weighting factor that adjusts the severity of the penalty for large height changes between points.

In both 2D and 3D environments, weight parameters  $w_{\text{coverage}}$ ,  $w_{\text{redundancy}}$ ,  $w_{\text{length}}$ ,  $w_{\text{turn}}$  and  $w_z$  are tuned experimentally.

## 4. Experimental configuration

To evaluate the performance of the NEAT-based path planning algorithm, simulations are conducted in both 2D and 3D environments. The experiments focus on assessing the ability of the algorithm to optimise coverage, path length and smoothness while adapting to different geometric constraints. Additionally, this experiment compares NEAT with traditional

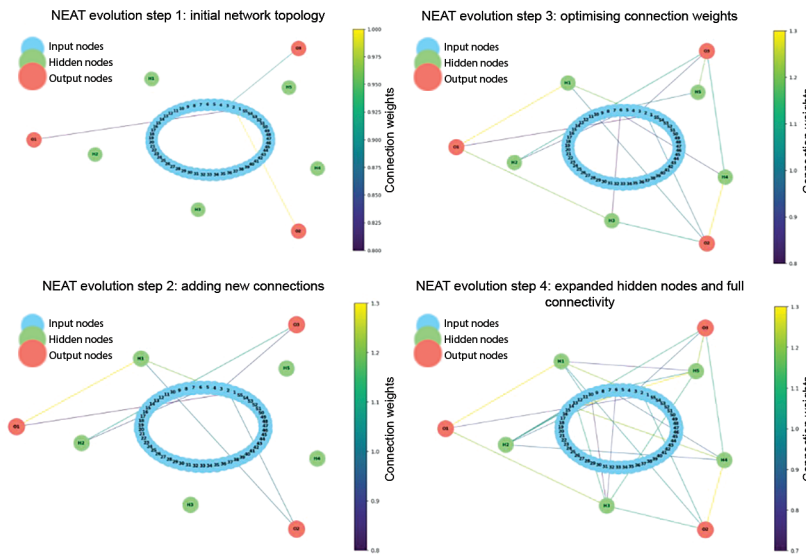


Figure 6. NEAT algorithm: progressive evolution of network structure

path planning methods, including A\*, Dijkstra, a multi-layer perceptron (MLP) and a GA, to highlight its advantages in dynamic path optimisation.

## 4.1 Experiment set-up

The 2D workspace is modelled as a discrete grid, where each node represents an inspection point. The agent moves between adjacent nodes, with horizontal and vertical moves assigned a unit cost and diagonal moves assigned a higher cost.

In the 3D workspace, the environment is represented as a set of discrete points sampled from a curved surface, forming a point cloud. The agent plans its path in three-dimensional space while minimising height variations to ensure surface alignment.

For decision-making, NEAT is used to evolve an FNN, which determines the next target point based on the agent's current position and visited history. The network is trained using a fitness function (see Section 3.3) to optimise coverage, path efficiency and smoothness by penalising redundant visits, excessive path length and sudden direction changes. To ensure stable evolution, the NEAT algorithm is implemented with a set of predefined parameters that control evolutionary operations such as mutation probability and network complexity. Table 1 presents the detailed parameter settings used in the simulation.

**Table 1. Key parameters for the NEAT algorithm and simulation environment**

| Parameter                        | Value   |
|----------------------------------|---------|
| Population size                  | 50      |
| Fitness criterion                | Maximum |
| Activation function              | Tanh    |
| Bias mutation rate               | 0.7     |
| Node add probability             | 0.2     |
| Node delete probability          | 0.5     |
| Connection add probability       | 0.5     |
| Connection weight mutation rate  | 0.8     |
| Connection weight mutation power | 1.5     |
| Number of hidden nodes           | 8       |
| Compatibility threshold          | 3.0     |
| Max stagnation                   | 100     |
| Survival threshold               | 0.2     |

## 4.2 Experimental results and analysis of NEAT path planning

The experiments evaluate the adaptability of the NEAT algorithm in four scenarios (see Figures 7 and 8): a 2D regular grid, a circular region, an obstacle-filled environment and a triangular area. In the presence of obstacles, the path adapts to the distribution of obstacles, bypasses the impassable regions and maintains complete coverage of detection points. In a regular grid environment, the path shows a raster scan-like pattern, which reduces unnecessary turns and maintains a smooth and orderly structure. In irregular boundary scenarios, the path distribution closely aligns with the geometric features of the region, avoiding crossings and invalid backtracking.

The evolutionary process of NEAT shows obvious stage characteristics in Figure 9. In the initial stage (0-30 generations), there are multiple path strategies within the population, the neural network structure displays high diversity and the population

diversity index is recorded as  $D_s \approx 0.78$ . As depicted in the fitness curve (see Figure 10), individual fitness exhibits pronounced fluctuations during the early evolutionary phase, with a fitness variance of approximately  $\sigma_f^2 \approx 0.28$ , which suggests that the quality of path planning varies considerably between individuals. In the intermediate stage (30-80 generations), the species competition mechanism gradually selects more adaptive individuals, inefficient network topologies are eliminated and the mean value of fitness gradually increases. The paths become more structured, the backtracking phenomenon decreases and the overall path planning strategy tends to be stable. By the late 80th generation, the proportion of dominant species remains largely unchanged and the neural network structure becomes fixed, indicating that the NEAT algorithm has converged to a stable path planning mode. The final path demonstrates clear optimisation trends, including priority coverage of unvisited areas, reduction of redundant turns and reasonable obstacle avoidance.

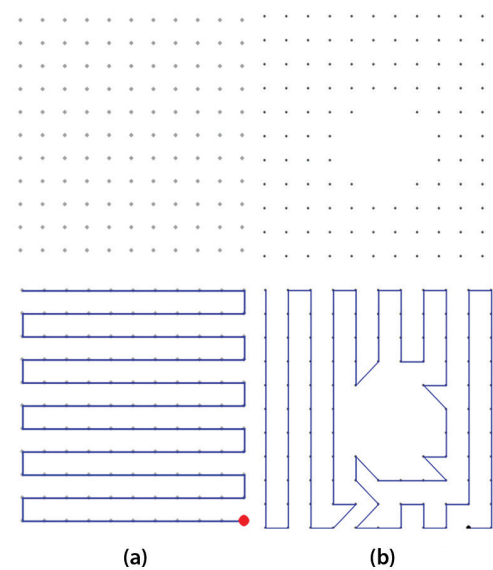


**Figure 7. NEAT-generated paths for different inspection area shapes: (a) a grid-like inspection area; and (b) a complex inspection area with multiple obstacles**

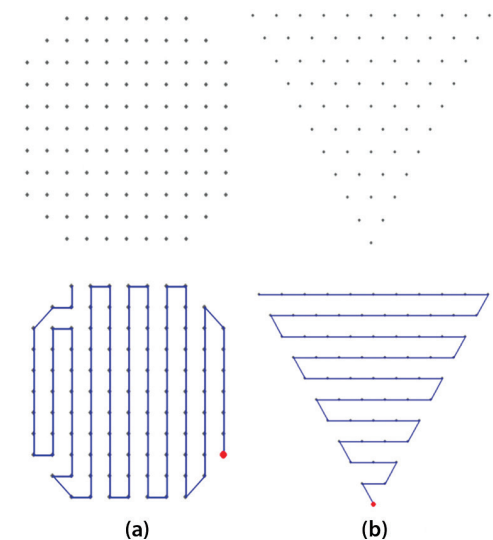


**Figure 8. Optimised inspection paths generated for circular (a) and triangular (b) environments**

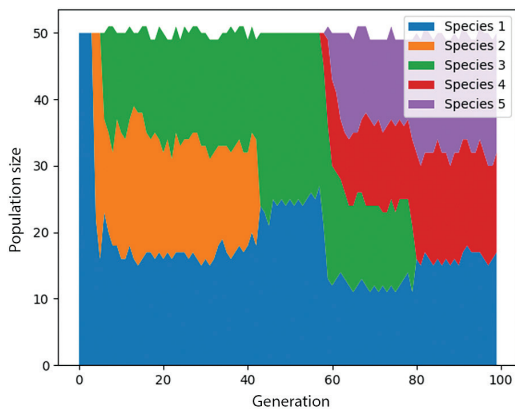


Figure 9. Distribution map of individuals in the population

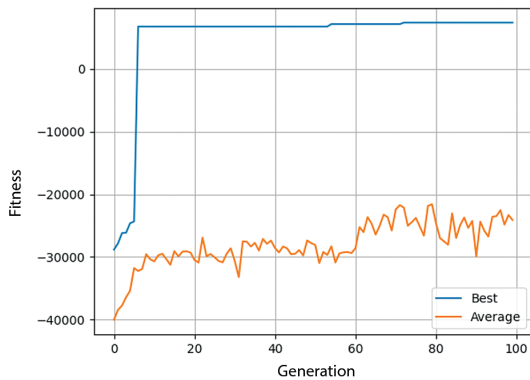


Figure 10. Evolutionary curve of population fitness over 100 generations

The experiments verify the planning capability of the NEAT algorithm in four typical 3D environments: a planar environment, a U-shaped surface, an obstacle-containing surface and a spherical surface (see Figures 11-14). In the planar environment, the algorithm generates raster-like scanning paths to achieve complete coverage of detection points without backtracking; in the U-shaped surface scenario, the paths closely conform to the geometric contours and the height weighting coefficients are dynamically adjusted to maintain the smoothness of the motion; in the environment of obstacle-containing surfaces, the algorithm autonomously avoids the gap region and restructures the path topology to ensure complete coverage; and in the closed spherical surface, the planned paths strictly adhere to surface constraints, avoiding any internal traversal and forming a continuous coverage loop.

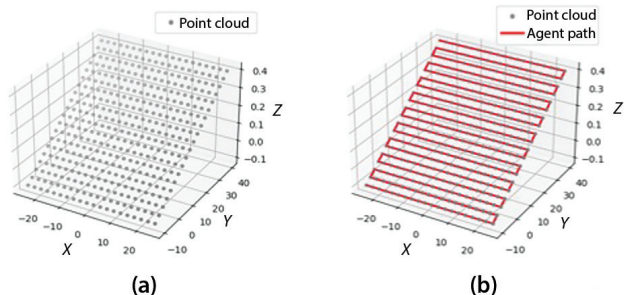


Figure 11. 3D planar inspection environment: (a) the initial point cloud representing the inspection area; and (b) the planned path following a raster scan pattern

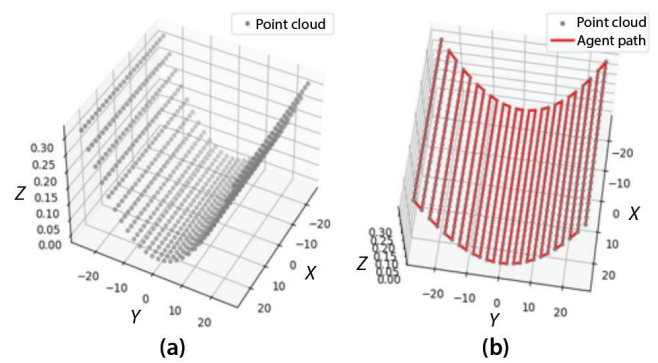


Figure 12. 3D U-shaped inspection environment: (a) the point cloud representing the curved inspection area; and (b) the planned path adapting to surface curvature while maintaining raster scanning

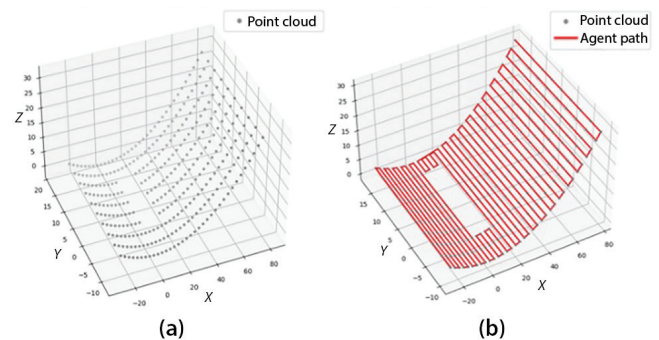


Figure 13. 3D curved surface with an obstacle: (a) the inspection area containing a central hole; and (b) the planned path adapting to the surface while avoiding the hole

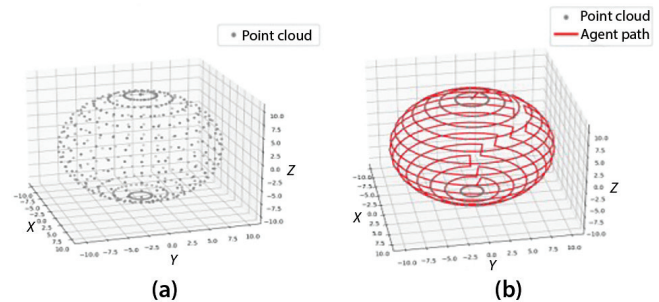


Figure 14. 3D spherical inspection environment: (a) the point cloud representing a closed spherical surface; and (b) the planned path constrained to the surface without internal traversal

### 4.3 Quantitative evaluation of path planning algorithms

To comprehensively and objectively assess the performance of the algorithms proposed in this study, a multi-dimensional evaluation system is established and the following normalisation method is used for the uniform scale processing.

For inverse metrics (for example path length, computational complexity, redundant visits), the normalisation formula is defined as:

$$X' = 0.05 + 0.75 \times \frac{X_{\max} - X}{X_{\max} - X_{\min}} \dots\dots\dots (11)$$

For positive metrics (for example obstacle avoidance, path smoothness), where higher values indicate better performance, the normalisation formula is given by:

$$X' = 0.05 + 0.75 \times \frac{X - X_{\min}}{X_{\max} - X_{\min}} \dots\dots\dots (12)$$

Here,  $X_{\min}$  and  $X_{\max}$  represent the minimum and maximum values of each metric across all evaluated algorithms. The normalised results are mapped onto the interval [0.05, 0.8] to prevent extreme values from disproportionately influencing the evaluation.

The final comprehensive score is calculated as follows:

$$S = 0.2P_L + 0.3O_A + 0.1C_C + 0.2S_S + 0.2R_V \dots (13)$$

The weighting coefficients are determined based on industrial inspection requirements:

- Obstacle avoidance (0.3) and dynamic continuity (path smoothness 0.2, redundant visit rate 0.2) are considered dominant factors.
- Path length (0.2) and computational efficiency (0.1) are of secondary importance.

The quantitative evaluation shows that the comprehensive performance score of NEAT is 130% higher than that of traditional methods, as presented in Table 2, with obstacle avoidance success rates and path smoothness both exceeding 0.95. The algorithm balances multi-objective conflicts through neural network topology evolution: it reduces path length by 19.7% compared to Dijkstra's method, minimises redundant access to below 5% and maintains computational complexity at a manageable level. The analysis of key parameters reveals that the height weight coefficients are automatically optimised by evolution, effectively suppressing the drastic fluctuations in the vertical direction; the curvature constraints are implicitly incorporated into the network's decision-making process to ensure the path continuity and smooth transitions.

Experiments reveal the core advantages of NEAT in 3D planning: it autonomously learns the mapping between environmental features and motion strategies through the neuroevolutionary mechanism and achieves a dynamic balance among surface continuity, obstacle avoidance and coverage integrity. This capability makes NEAT highly suitable for complex industrial surface inspection tasks, offering a novel paradigm for path optimisation under multiple constraints. Although NEAT requires greater offline training effort compared to reactive planners, its learned policy can be reused across similar inspection tasks, allowing the training cost to be amortised in high-volume or batch inspection settings.

**Table 2. Normalisation parameters and final scores of different path planning algorithms**

| Method   | Final score | Path length norm | Obstacle avoidance norm | Computation norm | Smoothness norm | Redundant norm |
|----------|-------------|------------------|-------------------------|------------------|-----------------|----------------|
| NEAT-GK  | 0.918       | 0.85             | 0.95                    | 0.85             | 0.95            | 0.95           |
| A*       | 0.375       | 0.42             | 0.15                    | 0.36             | 0.31            | 0.17           |
| Dijkstra | 0.361       | 0.45             | 0.18                    | 0.05             | 0.27            | 0.21           |
| MLP      | 0.398       | 0.38             | 0.22                    | 0.58             | 0.33            | 0.15           |
| GA       | 0.267       | 0.35             | 0.05                    | 0.25             | 0.05            | 0.05           |

## 5. Conclusion

In this study, a path planning framework fusing the NEAT algorithm and graph theory is proposed to effectively address the multi-objective optimisation problem in complex environments for non-destructive inspection tasks. By combining the neural evolution mechanism with topological constraints, the algorithm achieves a dynamic balance among coverage integrity, motion efficiency and the environmental adaptability of the inspection path. Experimental results show that the method generates globally optimal coverage paths in both 2D and 3D scenarios. It also exhibits superior obstacle avoidance capability compared with traditional algorithms in obstacle environments and effectively suppresses redundant motion through a weighted path optimisation strategy. The core innovations lie in three aspects: integrating neural network topology evolution with graph-based path planning; driving neural networks to accurately capture spatial topological constraints through a multi-objective fitness function; and optimising path smoothness and motion continuity via fitness function design.

Although the 3D simulation results confirm the feasibility of the proposed framework in complex inspection scenarios, practical ultrasonic robotic systems are subject to sensor noise, localisation uncertainty and mechanical inaccuracies, which may lead to path deviation, reduced scan precision or incomplete coverage. The NEAT algorithm offers some robustness through its dynamic topology evolution and adaptive path strategies. Future work will address three practical challenges: verifying the performance of the system under sensor drift in real-world conditions; scaling up the algorithm for large and complex inspection surfaces; and integrating motion constraints from real robotic systems into the planning framework.

## References

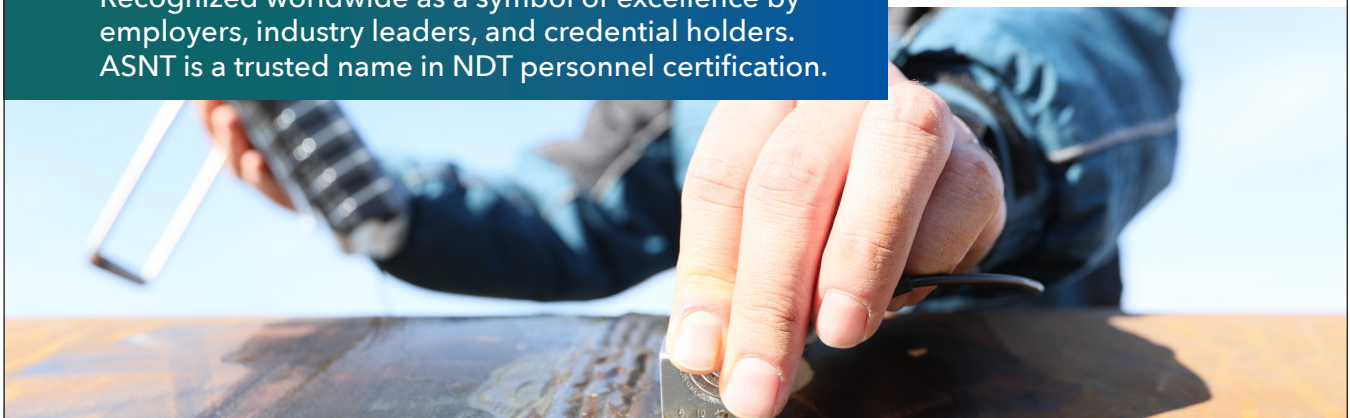
1. L W Schmerr Jr, 'An ultrasonic system', In: L W Schmerr Jr (ed) Fundamentals of Ultrasonic Nondestructive Evaluation: A Modeling Approach, pp 1-13, Springer International Publishing, Cham, Switzerland, 2016. DOI: 10.1007/978-3-319-30463-2\_1
2. J Riise, C Mineo, S G Pierce, P I Nicholson and I Cooper, 'Adapting robot paths for automated NDT of complex structures using ultrasonic alignment', AIP Conference Proceedings, Vol 2102, No 1, 040006, May 2019. DOI: 10.1063/1.5099756
3. E L Lawler, The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization, Wiley, Chichester, West Sussex, UK/New York, USA, 1987. Available at: <http://archive.org/details/travelingsalesma00lawl> (Accessed: 9 October 2024).
4. M Zhang, M Sutcliffe, P I Nicholson and Q Yang, 'Efficient autonomous path planning for ultrasonic non-destructive testing: a graph theory and K-dimensional tree optimisation approach', Machines, Vol 11, No 12, 1059, December 2023. DOI: 10.3390/machines11121059
5. R Long, J Russell and P Cawley, 'Ultrasonic phased array inspection using full matrix capture', Insight: Non-Destructive Testing and Condition Monitoring, Vol 54, No 7, pp 380-385, July 2012. DOI: 10.1784/insi.2012.54.7.380
6. E Galceran and M Carreras, 'A survey on coverage path planning for robotics',

- Robotics and Autonomous Systems, Vol 61, No 12, pp 1258-1276, December 2013. DOI: 10.1016/j.robot.2013.09.004
7. K Felsner, K Schlachter and S Zambal, 'Robotic coverage path planning for ultrasonic inspection', Applied Sciences, Vol 11, No 22, 10512, January 2021. DOI: 10.3390/app112210512
8. C Mineo, S G Pierce, P I Nicholson and I Cooper, 'Robotic path planning for non-destructive testing: a custom MATLAB toolbox approach', Robotics and Computer-Integrated Manufacturing, Vol 37, pp 1-12, February 2016. DOI: 10.1016/j.rcim.2015.05.003
9. Y Hu, J Hao, W Yang, C Guo and P Jiang, 'Motion point calculation method for robotic ultrasonic non-destructive testing', Proceedings of the 2018 IEEE Far East NDT New Technology & Application Forum (FENDT 2018), Xiamen, China, pp 85-89, 6-8 June 2018. DOI: 10.1109/FENDT.2018.8681988
10. Y Li, L Zeng, K Tang and C Xie, 'Orientation-point relation-based inspection path planning method for five-axis OMI system', Robotics and Computer-Integrated Manufacturing, Vol 61, 101827, February 2020. DOI: 10.1016/j.rcim.2019.101827
11. Y Chen, B Tan and L Zeng, 'Inspection path planning of free-form surfaces based on improved cuckoo search algorithm', Measurement and Control, Vol 56, No 7-8, pp 1321-1332, September 2023. DOI: 10.1177/00202940231157422
12. A Poole, M Sutcliffe, G Pierce and A Gachagan, 'Autonomous, digital-twin free path planning and deployment for robotic NDT. Introducing LPAS: locate, plan, approach, scan using low-cost vision sensors', Applied Sciences, Vol 12, No 10, 5288, January 2022. DOI: 10.3390/app12105288
13. C Mineo, D Cerniglia and A Poole, 'Autonomous robotic sensing for simultaneous geometric and volumetric inspection of free-form parts', Journal of Intelligent & Robotic Systems, Vol 105, No 3, 54, June 2022. DOI: 10.1007/s10846-022-01673-6
14. B Akgun and M Stilman, 'Sampling heuristics for optimal motion planning in high dimensions', Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, California, USA, pp 2640-2645, 25-30 September 2011. DOI: 10.1109/IROS.2011.6095077
15. L Han, L He, X Sun, Z Li and Y Zhang, 'An enhanced adaptive 3D path planning algorithm for mobile robots with obstacle buffering and improved Theta\* using minimum snap trajectory smoothing', Journal of King Saud University – Computer and Information Sciences, Vol 35, No 10, 101844, December 2023. DOI: 10.1016/j.jksuci.2023.101844
16. R S Lim, H M La and W Sheng, 'A robotic crack inspection and mapping system for bridge deck maintenance', IEEE Transactions on Automation Science and Engineering, Vol 11, No 2, pp 367-378, April 2014. DOI: 10.1109/TASE.2013.2294687
17. P a/l Rajendran and M Othman, 'A comparative study on ant-colony algorithm and genetic algorithm for mobile robot planning', In: R Ghazali, N M Nawi, M M Deris, J H Abawajy and N Arbaiy (eds), Recent Advances on Soft Computing and Data Mining, pp 286-295, Springer Nature, Cham, Switzerland, 2024. DOI: 10.1007/978-3-031-66965-1\_28
18. X Xie, Z Yan, Z Zhang, Y Qin, H Jin and M Xu, 'Hybrid genetic ant colony optimisation algorithm for full-coverage path planning of gardening pruning robots', Intelligent Service Robotics, Vol 17, No 3, pp 661-683, May 2024. DOI: 10.1007/s11370-024-00525-6
19. K O Stanley and R Miiikkulainen, 'Evolving neural networks through augmenting topologies', Evolutionary Computation, Vol 10, No 2, pp 99-127, June 2002. DOI: 10.1162/106365602320169811
20. E Papavasileiou, J Cornelis and B Jansen, 'A systematic literature review of the successors of 'neuroevolution of augmenting topologies'', Evolutionary Computation, Vol 29, No 1, pp 1-73, March 2021. DOI: 10.1162/evco\_a\_00282
21. Z Fanchao, M Y H Low, J Decraene, S Zhou and W Cai, 'Self-adaptive mechanism for multi-objective evolutionary algorithms', Lecture Notes in Engineering and Computer Science, Vol 2180, March 2010.

©2025 TWI Ltd

## Set yourself apart with ASNT | CERTIFICATION.

Recognized worldwide as a symbol of excellence by employers, industry leaders, and credential holders. ASNT is a trusted name in NDT personnel certification.



**ASNT CERTIFICATION  
SERVICES, LLC.**



For more information on our available certifications and to learn how you can earn your own, visit: [asntcertification.org](https://asntcertification.org)

Copyright of Insight: Non-Destructive Testing & Condition Monitoring is the property of British Institute of Non-Destructive Testing and its content may not be copied or emailed to multiple sites without the copyright holder's express written permission. Additionally, content may not be used with any artificial intelligence tools or machine learning technologies. However, users may print, download, or email articles for individual use.