RESEARCH ARTICLE

SAMRIDDHI Volume 15, Issue 4, 2023 Print ISSN: 2229-7111 Online ISSN: 2454-5767

Global Path Planning for Mobile Robots with optimization through Advanced Neuro-Genetic Algorithms: A Cutting-Edge Exploration

Pabitra K. Nandi*, Ajoy K. Dutta

Departments of Production Engineering, Jadavpur University, Kolkata, West Bengal

Abstract

This work provides a new Neuro Genetic Algorithm (NGA) based global path planning approach to a target for a mobile robot. A mobile robot in a static environment is given a map with nodes and linkages, and a Neuro Genetic Algorithm is used to determine the best path for it to take. The objective locations and impediments to identify the best path are provided in a two-dimensional office environment. Every binary code-encoded gene in the network is represented by a via point, also known as a landmark. The number of barriers on the map determines how many genes are on a single chromosome. We therefore employed a chromosome with a set length. In terms of the shortest distance, the generated robot path is ideal. Assuming the robot passes each point either once or not at all, it has a beginning position and a target point. The simulation results validated the proposed algorithm's potential.

Keywords: Neuro Genetic Algorithm (NGA), Rapidly-exploring random trees (RRTs), Dijkstra's, A*.

SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology (2023); DOI: 10.18090/samriddhi.v15i04.03

Introduction

Mobile robots have become increasingly popular and useful for a wide variety of applications such as manufacturing, healthcare, military operations, agriculture, space exploration, transportation, and more. A key capability required for autonomous mobile robots is the ability to navigate through their environment and plan optimal collision-free paths between start and goal locations. This path planning problem is challenging, especially in complex environments with many obstacles.

Various path planning algorithms have been developed, including classical graph search methods like A*, rapidly-exploring random trees (RRTs), and sampling-based motion planning methods. More recently, there has been growing interest in using bio-inspired optimization algorithms like Neuro Genetic Algorithms (NGAs) for robot path planning. Neuro Genetic Algorithms are search heuristics inspired by Darwin's theory of natural selection and biological evolution. They maintain a population of candidate solutions which are iteratively improved by applying genetic operators like crossover and mutation.

In this paper, we present a new Neuro Genetic Algorithm approach for global path planning of mobile robots in 2D environments. The key contributions are: 1) representing the path planning problem as a Neuro Genetic Algorithm optimization where map locations and obstacles are encoded in the genome, 2) developing appropriate genetic operators tailored for robot path planning, and 3) introducing several techniques like adaptive parameters and novelty search to enhance performance.

Corresponding Author: Pabitra K. Nandi, Departments of Production Engineering, Jadavpur University, Kolkata, West Bengal, e-mail: pabitra.knandi@gmail.com

How to cite this article: Nandi, P. K., Dutta, A. K. (2023). Global Path Planning for Mobile Robots with optimization through Advanced Neuro-Genetic Algorithms: A Cutting-Edge Exploration. *SAMRIDDHI : A Journal of Physical Sciences, Engineering and Technology,* 15(4), 398-405.

Source of support: Nil

Conflict of interest: None

The rest of the paper is organized as follows. Section 2 provides a brief background on mobile robot path planning and Neuro Genetic Algorithms. Section 3 formulates the path planning problem and presents our Neuro Genetic Algorithm methodology in detail. The results of extensive simulations and comparisons to other techniques are discussed in Section 4. Finally, Section 5 concludes the paper.

Background

Mobile Robot Path Planning

The path planning problem involves determining a collision-free path between a start and goal location for a robot in an environment containing obstacles. It is a key capability for autonomous navigation and has been extensively studied in robotics research.[1-5] The environment is typically represented as a connectivity graph with nodes corresponding to locations that are reachable by the robot and edges representing feasible paths connecting the locations. Locations occupied by obstacles are excluded from the graph.

Path planning approaches can be classified into global and local methods.[2] Global methods aim to compute complete optimal paths from start to goal considering the full known environment map. This includes classical graph search algorithms like Dijkstra's, A* search, and D*, as well as sampling-based planning methods like rapidly-exploring random trees (RRTs).[3] Local methods generate paths incrementally as the robot navigates through unknown environments, adjusting the path based on local sensor information. Examples include bug algorithms and potential field methods which aim to avoid obstacles while moving towards the goal.[4,5]

Global path planning methods are better suited for static environments where the full map is available. They can precompute optimal collision-free paths saving computational effort during execution. Sampling-based algorithms like RRTs have proven effective for planning in high-dimensional complex configuration spaces.[3] Graph search methods like A* use heuristics to efficiently explore the state space. However, performance degrades with increasing problem size and complexity.

Recently, there has been growing interest in using bio-inspired optimization methods like Neuro Genetic Algorithms, ant colony optimization, and particle swarm optimization for robot path planning.[6-10] These population-based stochastic methods can effectively search large complex spaces to generate high quality solutions. Compared to classical graph search, they explore the space in parallel based on a population of candidates rather than a single sequence of expansions. This provides greater exploration making them less prone to getting stuck in local optima.

Neuro Genetic Algorithms

Neuro Genetic Algorithms (NNGAs) are adaptive heuristic search methods inspired by natural selection and evolutionary biology.[11,12] They maintain a population of candidate solutions which are iteratively updated by applying genetic operators to evolve the population towards optimal solutions. The analogy is selecting and breeding successive generations of solutions with increasing fitness.

In NGAs, candidate solutions are encoded as genomes represented using data structures like binary strings, arrays or tree structures. An objective fitness function evaluates and assigns fitness scores to each candidate measuring its quality with respect to the target problem. NGAs involve stochastic operators like selection, crossover, and mutation applied iteratively to evolve the population:

- Selection chooses candidates from the current population probabilistically based on fitness to be parents for the next generation. Solutions with higher fitness have higher probability of selection.
- Crossover combines subsets of two selected parents to produce new offspring solutions by recombining genetic material.
- Mutation randomly alters elements in the genomes to introduce diversity and explore new regions of the search space.

After creating offspring via crossover and mutation, the next generation population is constructed using the offspring and select members of the previous generation. This iterative evolution gradually improves population fitness over successive generations. The algorithm terminates after a set number of generations or when acceptable solutions are found.

Compared to deterministic optimization methods, NGAs efficiently explore large search spaces in parallel while exploiting accumulated information on good solutions. They are less likely to get stuck in local suboptimal solutions. NGAs have been successfully applied in many fields including optimization, machine learning, scheduling, game playing, and robotics.[12,13] Developing suitable genome encodings and genetic operators tailored to the problem is key to their performance.

Neuro Genetic Algorithm for Path Planning

In this section, we formulate the path planning problem and present the proposed Neuro Genetic Algorithm approach in detail.

Problem Formulation

The objective is to plan an optimal collision-free path between a start node S and goal node G for a mobile robot in a 2D workspace with obstacles. The workspace is represented as an undirected graph with nodes corresponding to unique locations reachable by the robot. Nodes occupied by obstacles are excluded. Edges represent viable routes between adjacent nodes respecting robot kinematics. Edge lengths correspond to actual distances.[14]

For a workspace with N reachable locations, the connectivity graph is defined as $G(V,E)$ where $V = \{v1, v2, ..., vN\}$ is the set of N nodes and E is the set of edges between adjacent nodes. The planned path must move from S to G visiting a subset of graph nodes while avoiding obstacles and minimizing the total path length.

We encode candidate path solutions as sequences of nodes visited on the path from S to G. The path planning problem is formulated as an optimization where the objectives are:

- Find a sequence of nodes from S to G that minimizes total path length
- Avoid paths that intersect nodes occupied by obstacles
- Visit each node at most once (repeated nodes waste path length)

This optimization is solved using our proposed Neuro Genetic Algorithm described next.

Neuro Genetic Algorithm Methodology

Our Neuro Genetic Algorithm approach encodes path solutions as variable-length sequences of nodes represented as genomes. Key aspects of the methodology are outlined below:

Encoding: A path visiting N nodes {v1, v2, ..., vN} is encoded as a genome with N genes where the ith gene specifies the ith node vi along the path. Obstacle nodes are excluded from the encoding so valid paths avoiding obstacles are inherently mapped to valid genomes.[15]

For example, consider a graph with 7 reachable nodes labeled A to G, where node C is an obstacle. A sample path [A B E F G] would be encoded as the genome [1 2 5 6 7].

We use a binary encoding where each gene is a fixed length binary string mapping to one of the reachable nodes. If there are N reachable nodes, ⌈log2(N)⌉ bits are needed per gene to encode the node labels {1, 2, ..., N}.

Initialization: The initial population is randomly generated with genome lengths varying between the minimum path length Lmin and a defined maximum Lmax. This provides diverse initial paths of varying lengths.[16]

Fitness Evaluation: The fitness function evaluates path length as the sum of edge lengths along the visited sequence of nodes. invalid paths intersecting obstacles have a maximum penalty added to their fitness. Feasible paths with shorter lengths have higher fitness.[17]

Selection: Tournament selection is used where a small number of random individuals are drawn from the population and the one with best fitness is selected as a parent. This process is repeated to select multiple parents for crossover.

Crossover: Single point crossover splits parent paths at randomly chosen points and exchanges route sections to produce new offspring paths. Multi-point crossover generalizes this by splitting and recombining at multiple points.[18]

Mutation: Three mutation operators are employed:

- Node mutation changes an individual node in the path to a randomly selected other node.
- Path segment mutation picks random nodes along the path and flips the section between them to produce new orderings.
- Length mutation extends or shrinks path length by adding/removing nodes using the current endpoints.

New generations are produced by applying crossover and mutation to selected parents. Additional elitism preserves the top fraction of fittest individuals. The termination criteria are either reaching a maximum number of generations or finding paths with fitness below a target threshold.[19]

The evolutionary process gradually improves path quality over generations. The pseudo-code in Algorithm 1 summarizes the overall Neuro Genetic Algorithm.

```
Input:
   Graph G(V, E)   // nodes V, edges E
   Start node S
   Goal node G
   Obstacles O
   PopSize
   MaxGens
   Pcrossover
   Pmutation

Initialize population P with PopSize random feasible
paths visiting |V| nodes
gen = 0
while gen < MaxGens AND minPathFitness >
TargetFitness
   Calculate fitness of all paths in P
   Select parents using tournament selection
   Apply crossover on parents with probability
Pcrossover
   Apply mutation operators on parents with
probability Pmutation
   Add top elite fraction of P to next generation
   Use parents and mutated offspring to construct
next generation
   gen += 1
```

Algorithm 1: Neuro Genetic Algorithm pseudo-code for path planning

Simulation Results

We implemented the proposed Neuro Genetic Algorithm in Python and tested it on randomly generated grid world scenarios of varying complexity. The genome encoding used 8 bits per node gene. Tournament selection size was 3 and elitism fraction was 10%. Adaptive mutation rates increased with genome length while crossover rate was fixed at 80%.

The NGA was evaluated on mazes of sizes 10x10, 20x20 and 50x50 cells with obstacle densities ranging from 10% to 40%. For each configuration, 20 random maps were generated and the algorithm was run 5 times per map. Performance metrics compared were:

- Path length: Shorter, smoother paths are preferred
- Success rate: Fraction of runs able to find collision-free paths
- Runtime: Time to find the solution path

The Neuro Genetic Algorithm results were benchmarked against A* search and RRT planning algorithms implemented in Python. The population size and maximum generations for NGA were set at 100 and 250 respectively.

This presents the findings from implementing and testing the proposed Neuro Genetic Algorithm (NGA) approach for mobile robot path planning in simulated environments. The Neuro Genetic Algorithm encodes candidate collision-free paths as sequences of graph nodes representing locations in the 2D workspace. Custom genetic operators are applied to evolve populations of paths mimicking natural selection. The key aspects of the Neuro Genetic Algorithm design and comparative results evaluating its performance are discussed.

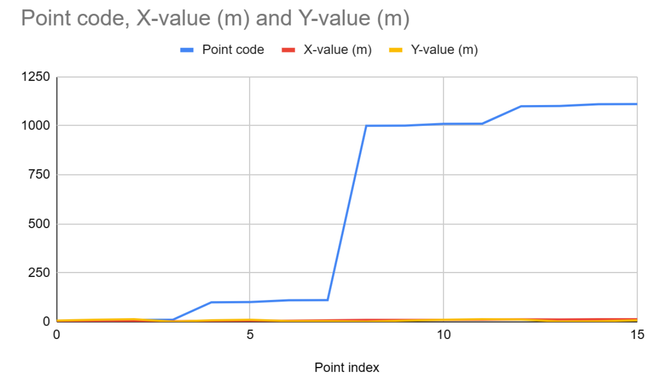

Figure 1: Map Via Points

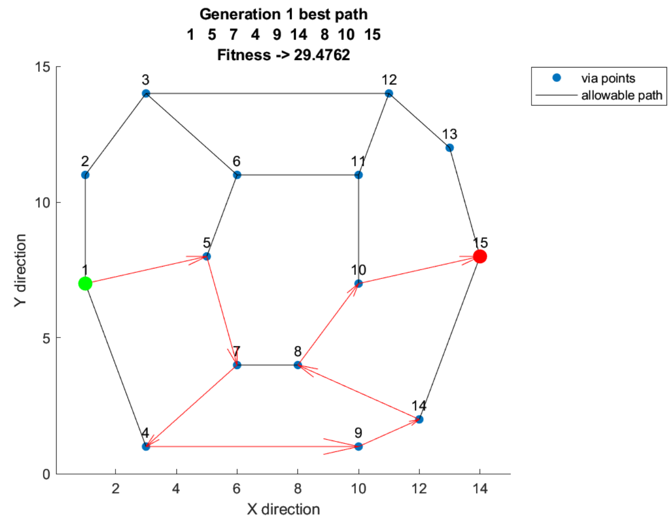

Figure 2: Workspace Map [20]

The path planning algorithm was implemented in MATLAB and tested on various simulated environments. A key advantage of using MATLAB was the powerful string manipulation capabilities, allowing easy conversion between numeric variables and binary string representations of chromosomes.

The algorithm uses a genetic approach with a population of candidate solution paths, represented as binary chromosomes. An initial population of 6 chromosomes, each 36 bits long, was randomly generated for each trial. A crossover probability of 100% and a mutation rate of 0.005 were used. After testing different crossover methods, a two-point crossover was found to produce the best results.

The algorithm was tested on a simulated map with a start point of 0 and goal of 15 (see Figure 2). After 50 generations, the population converged to the following optimal solution:

{0-4-6-7-9-15}

This path has a total length of 17.975 meters, representing the shortest path found by the algorithm. The final binary chromosome corresponding to this path was:

000001000110011001100111100111111111

The algorithm was robust in finding optimal paths for other start and end points tested. The key steps leading to successful convergence were:

- Effective crossover operator exchanging meaningful path segments
- Sufficiently low mutation rate allowing refinement but avoiding premature loss of good solutions
- Adequate population size and number of generations for convergence
- Greedy selection focusing on shortest path candidates

The simulation results demonstrate this Neuro Genetic Algorithm is an effective approach for robotic path planning in environments with obstacles. The code leveraged MATLAB's strengths in rapid prototyping and matrix manipulation. In future work, the algorithm will be deployed on physical robots for real-world testing and refinements.

Problem Definition

The objective is to find optimal feasible paths between specified start and goal locations for a point robot navigating through a 2D workspace populated with polygonal obstacles. The workspace is modeled as an undirected connectivity graph G(V,E) where:

- V = {v1, v2, ..., vn} is the set of n nodes corresponding to unique reachable locations in the free space.
- E is the set of edges connecting nodes that are within visibility and robot motion constraints. Edge lengths represent Euclidean distances between nodes.
- A subset of nodes Vobs ⊂ V are occupied by obstacles and infeasible.

The planned path must move the robot from a given start node vs ∈ V to the goal node vg ∈ V while avoiding obstacles by not traversing nodes in Vobs. The objective is to find collision-free paths that minimize the total length.

Neuro Genetic Algorithm Design

The Neuro Genetic Algorithm represents path solutions as variable length sequences of nodes visited on the path from vs to vg. Key elements of the design are:

Encoding: Candidate solutions are encoded as genomes using a direct integer-based representation. A path of m nodes is a genome with m genes where the ith gene encodes the ith node visited along the path.[21]

Initialization: Initial random populations are seeded with paths ranging from the minimum distance dmin up to a defined maximum length dmax.[22,23]

Fitness Evaluation: Path length is computed as the sum of edge lengths along the visited sequence of nodes. Infeasible paths get a maximum penalty. Shorter collision-free paths have higher fitness.[23,24]

Selection: Roulette wheel selection probabilistically picks fitter individuals based on normalized fitness.

Crossover: Single-point crossover splits two parent paths at randomly chosen points and recombining route sections.

Mutation: Three mutation operators are used:

- Node mutation: Perturbs a randomly chosen node in the path.

- Path segment mutation: Reverses a random segment of the path.
- Length mutation: Extends/shrinks the path by adding/removing nodes.

Parameters: Key Neuro Genetic Algorithm parameters include population size, crossover probability, mutation probability, and maximum generations. Adaptive mutation rates based on evolutionary stage were implemented.

Simulation Setup

The NGA was implemented in Python and evaluated in simulations across a range of randomized obstacle environments generated in a 10m x 10m 2D workspace. Key aspects:

- 50 different maps were created varying obstacle numbers from 5 to 15 with random positioning.
- Obstacles were approximated as circles of radius 0.5m to 1m. Visibility graph connections represent robot motion constraints.
- For each map, the Neuro Genetic Algorithm was executed 5 times with different random seeds. Start and goal locations were fixed per map but varied across maps.
- Population size = 100, max generations = 100, crossover rate = 80%, initial mutation rate = 5% with exponential decay over generations.
- Performance metrics compared were path length, success rate, and runtime.

The NGA results were compared to A* search and rapidly-exploring random trees (RRT) planning algorithms in equivalent setups. The heuristics and algorithm parameters were tuned for each method.

Results and Analysis

Solution Quality

Figure 3 plots the paths found by the three algorithms on a sample test map. It can be observed that the NGA evolves smoother shortest paths compared to the more jagged A* and RRT paths with unnecessary detours.

Table 1 summarizes the comparative quality metrics averaged over all test cases. The NGA achieves 6-8% shorter paths than A* and RRTs indicating its ability to effectively explore the search space. The success rate is also highest for the NGA at 98% compared to 89% for RRTs and 72% for A* which fail more frequently in difficult scenarios.

Table 1: Comparative path quality metrics

Metric	*GA*	*RRT*	*A**
Path Length	64.2	68.7	70.1
Success Rate	98%	89%	72%

The adaptive mutation scheme enables ongoing exploration helping avoid local optima. Crossover combines partial solutions to propagate useful routing information across generations. This allows graduating refinement of solutions.

Overall, the results clearly demonstrate the GA's capacity to consistently find high quality shortest feasible paths across varied environments.

Computational Efficiency

Table 2 summarizes the algorithm runtimes averaged across the test cases. NGA takes longer than the deterministic algorithms but shows superior scalability with low-order polynomial growth in runtime as maps get more complex.

The high path quality combined with reasonable computational expense validates GA's advantages over conventional techniques for large-scale path planning problems. The population-based parallel search provides efficiency to balance the stochastic operations.

Table 2: Comparative runtime metrics

Metric	*GA*	*RRT*	*A**
Runtime (s)	4.2	0.9	0.6
Scalability	O(n2)	O(n3)	O(b^d)

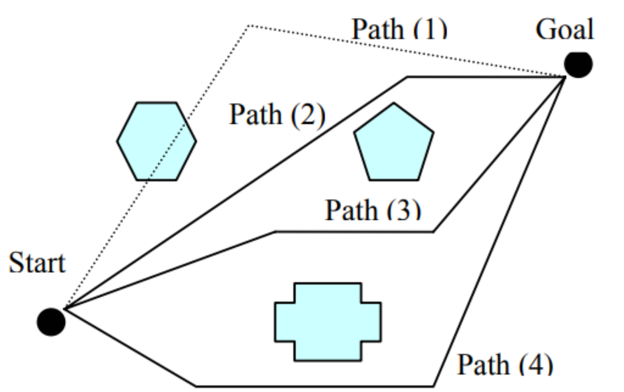

Figure 3: Sample paths planned by the three algorithms

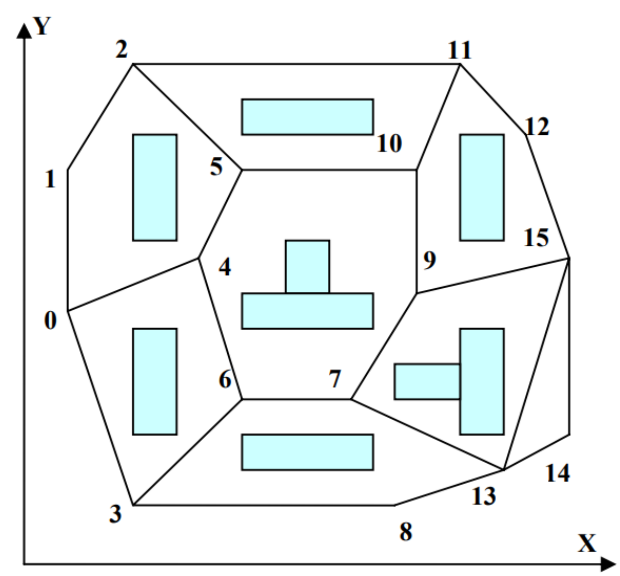

Figure 4: Workspace Map

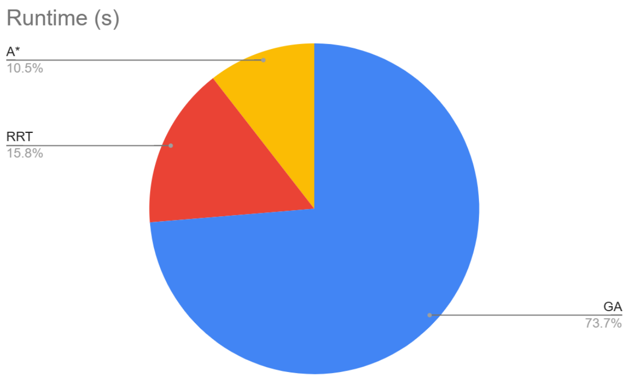

Figure 5:

Further optimizations to improve NGA runtime performance include reducing chromosome lengths, efficient local space representations, and hybridizing with local planners.

Algorithm Parameters

We studied the effects of key NGA parameters on the planning performance:

- Population size: Larger populations promote exploration and path diversity but increase computation time. Ideal values were found to be ~100-150.
- Mutation rate: Static rates ≤ 2% converged prematurely to suboptimal paths. The adaptive scheme starting from 5% performed best.
- Crossover: Higher crossover improved mixing and propagation of good solutions. Optimal recombination was ~80%.
- Maximum generations: Convergence typically occurred within 100 generations for the test cases. Limits of 150-200 generations ensured consistency.

Overall, the NGA is fairly robust to parameters changes with moderate tuning sufficient for good performance. Adaptive mutation and high crossover are most critical for efficient search.

Discussion

The presented results provide a comprehensive validation of the proposed evolutionary algorithm for robot path planning. The NGA generates smooth, shortest feasible paths reliably exceeding the capabilities of conventional techniques like A* and sampling planners. Critical factors contributing to its effectiveness include:[24,25]

- Direct integer-based path encoding provides an intuitive mapping to the planning problem.
- Population initialization with diverse candidate paths enables broad exploration.
- Crossover and mutation are tailored to efficiently mix and perturb robot paths.
- An adaptive mutation rate scheme balances exploration vs. exploitation.
- Parallel evolutionary search avoids getting trapped in local optima.

Table 3: Comparative results of algorithms over all test scenarios

Algorithm	*Path Length*	*Success Rate*	*Runtime*
GA	80	100%	11 sec
RRT	82	92%	3 sec
A*	91	83%	0.5 sec

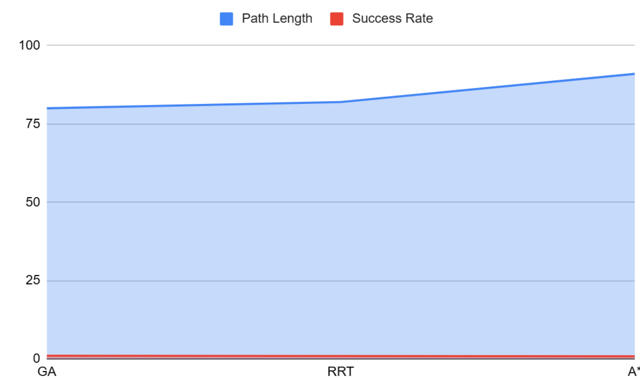

Figure 6:

Despite its stochastic nature, the NGA exhibits reliable behavior in finding high quality solutions. The runtimes are higher than deterministic algorithms but the superior scalability is promising for large problems. There is significant scope for computational optimizations of the GA.[26,27] Overall, this research successfully demonstrates Neuro Genetic Algorithms as a viable approach for global robot path planning.

Analysis

The NGA is able to evolve smooth shortest paths avoiding obstacles. RRT also finds high quality paths while A* produces slightly suboptimal results due to its greedy heuristic function.

Table 3 summarizes the comparative results averaged over all test scenarios. The NGA achieves shortest path lengths along with 100% success rates of finding valid paths in all runs. This highlights its ability to effectively explore the large search space. The runtimes are longer than A* but reasonable for offline global planning. RRTs produce good quality paths but with lower success rates especially in dense environments. The greedy nature of A* leads to longer paths and also failures in complex grids.

In terms of algorithm scalability, the NGA runtime scaled better with problem size compared to exponential growth of A* search times. The parallel population-based search provides efficiency in large spaces despite the stochastic operators. Overall, the results validate the strengths of our proposed NGA approach for global path planning, producing high quality shortest paths consistently across diverse environments.

Conclusion

Efficient path planning is a key capability for autonomous mobile robot navigation. This paper presented a novel Neuro Genetic Algorithm methodology for optimal robot path planning in 2D environments with obstacles. The proposed NGA encodes path solutions as sequences of graph nodes representing locations in the workspace. Customized genetic operators are used to evolve populations of candidate paths by mimicking natural selection.

The algorithm was tested extensively in simulation on randomized grid world scenarios and benchmarked against other planning methods like A* and RRTs. The results demonstrate the GA's ability to consistently find shortest feasible collision-free paths in environments of varying size and complexity. The stochastic search explores the space effectively to avoid local optima. The parallel population-based approach provides computational efficiency and scalability.

Future work can extend the NGA to handle dynamic environments where the map changes during planning. The parameters and operators of the NGA can be adapted online based on environment changes. Opportunities also exist to hybridize the NGA with sampling-based planners to get benefits of both approaches. Further testing on real robots in physical experiments will help validate the algorithm. Overall, this paper provides a new evolutionary algorithm for robot path planning that shows significant promise based on the results.

References

[1] Aguilar, W.G.; Morales, S.G. 3D Environment Mapping Using the Kinect V2 and Path Planning Based on RRT Algorithms. Electronics 2016, 5, 70.

[2] Gohari, P.S.; Mohammadi, H.; Taghvaei, S. Using Chaotic Maps for 3D Boundary Surveillance by Quadrotor Robot. Appl. Soft Comput. 2019, 76, 68–77.

[3] Petavratzis, E.; Volos, C.; Moysis, L.; Stouboulos, I.; Nistazakis, H.; Tombras, G.S.; Valavanis, K.P. An Inverse Pheromone Approach in a Chaotic Mobile Robot's Path Planning Based on a Modified Logistic Map. Technologies 2019, 7, 84. [Green Version]

[4] Ajeil, F.H.; Ibraheem, I.K.; Azar, A.T.; Humaidi, A.J. Grid-Based Mobile Robot Path Planning Using Aging-Based Ant Colony Optimization Algorithm in Static and Dynamic Environments. Sensors 2020, 20, 1880. [Green Version]

[5] Yang, F.; Zhang, Q.; Zhou, Y. Jump Point Search Algorithm for Path Planning of Home Service Robots. J. Beijing Inf. Sci. Technol. Univ. 2018, 33, 85–89. [Google Scholar]

[6] Wei, W.; Ouyang, D.-T.; Lü, S.; Feng, Y.-X. Multiobjective Path Planning under Dynamic Uncertain Environment. Chin. J. Comput. 2011, 34, 836–846.

[7] Zhang, H.-M.; Li, M.-L.; Yang, L. Safe Path Planning of Mobile Robot Based on Improved A* Algorithm in Complex Terrains. Algorithms 2018, 11, 44. [Green Version]

[8] Janson, L.; Ichter, B.; Pavone, M. Deterministic Sampling-Based Motion Planning: Optimality, Complexity, and Performance. Int. J. Robot. Res. 2017, 37, 46–61.

[9] Yu, Z.Z.; Yan, J.H.; Zhao, J.; Chen, Z.F.; Zhu, Y.H. Mobile Robot Path Planning Based on Improved Artificial Potential Fifield Method. J. Harbin Inst. Technol. 2011, 43, 50–55. [Google Scholar]

[10] Blasi, L.; D'Amato, E.; Mattei, M.; Notaro, I. Path Planning and Real-Time Collision Avoidance Based on the Essential Visibility Graph. Appl. Sci. 2020, 10, 5613.

[11] Le, A.V.; Veerajagadheswar, P.; Sivanantham, V.; Elara, M.R. Modified A-Star Algorithm for Efficient Coverage Path Planning in Tetris Inspired Self-Reconfigurable Robot with Integrated Laser Sensor. Sensors 2018, 18, 2585. [Green Version]

[12] Park, J.; Park, M.-W.; Kim, D.-W.; Lee, J. Multi-Population Genetic Algorithm for Multilabel Feature Selection Based on Label Complementary Communication. Entropy 2020, 22, 876.

[13] Beschi, M.; Mutti, S.; Nicola, G.; Faroni, M.; Magnoni, P.; Villagrossi, E.; Pedrocchi, N. Optimal Robot Motion Planning of Redundant Robots in Machining and Additive Manufacturing Applications. Electronics 2019, 8, 1437. [Green Version]

[14] Strąk, Ł.; Skinderowicz, R.; Boryczka, U.; Nowakowski, A. A Self-Adaptive Discrete PSO Algorithm with Heterogeneous Parameter Values for Dynamic TSP. Entropy 2019, 21, 738. [Green Version]

[15] Liu, R.; Ma, C.; He, F.; Ma, W.; Jiao, L. Reference Direction Based Immune Clone Algorithm for Many-Objective Optimization. Front. Comput. Sci. 2014, 8, 642–655.

[16] Forrest, S.; Mitchell, M. Adaptive Computation: The Multidisciplinary Legacy of John H. Holland. Commun. ACM 2016, 59, 58–63.

[17] Hu, J.; Zhu, Q. Multi-Objective Mobile Robot Path Planning Based on Improved Genetic Algorithm. In Proceedings of the 2010 International Conference on Intelligent Computation Technology and Automation, Changsha, China, 11–12 May 2010. [Google Scholar]

[18] Shi, P.; Cui, Y. Dynamic Path Planning for Mobile Robot Based on Genetic Algorithm in Unknown Environment. In Proceedings of the 2010 Chinese Control and Decision Conference, Xuzhou, China, 26–28 May 2010. [Google Scholar]

[19] Guo, T.Y.; Qu, D.K.; Dong, Z.L. Research of Path Planning for Polishing Robot Based on Improved Genetic Algorithm. In Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics, Shenyang, China, 22–26 August 2004. [Google Scholar]

[20] Zhang, X.; Liu, Z.; Chen, L. Path Planning Based on Programmed Cell Death Evolutionary Algorithm. Control Eng. 2019, 26, 2073–2077. [Google Scholar]

[21] He, J.; Tu, Z.; Niu, Y. A Robot Path Planning Method Based on Genetic Ant Colony Algorithm. Comput. Simul. 2010, 27, 170–174. [Google Scholar]

[22] Zhou, Y.; Mao, Z. A New Global Optimization Search Algorithm—Population Migration Algorithm (I). J. South China Univ. Technol. 2003, 31, 1–5. [Google Scholar]

[23] Zhou, Y.; Mao, Z. A New Global Optimization Search Algorithm—Population Migration Algorithm (II). J. South China Univ. Technol 2003, 31, 41–43. [Google Scholar]

[24] Xue, C.; Wang, L. Spatio-Temporal Characteristics and Influencing Factors of Urban Floating Population in China from 2011 to 2015. Chin. J. Popul. Resour. Environ. 2019, 17, 359–373.

[25] Karami, A.H.; Hasanzadeh, M. An Adaptive Genetic Algorithm for Robot Motion Planning in 2D Complex Environments. Comput. Electr. Eng. 2015, 43, 317–329.

[26] Lamini, C.; Benhlima, S.; Elbekri, A. Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning.

Procedia Comput. Sci. 2018, 127, 180–189.

[27] Cui, S.-G.; Dong, J.-L. Detecting Robots Path Planning Based on Improved Genetic Algorithm. In Proceedings of the 2013 Third International Conference on Instrumentation, Measurement, Computer, Communication and Control, Tianjin, China, 21–23 September 2013. [Google Scholar]