Path Planning of Robot Tracking Using Hybrid of a Star and Genetic Algorithm

Simhadri Sireesha[1], Maradana Ramya sri[2], Lankalapalli Surya Prabha[3],

Seeramreddy shanmukha mani kishore[4]

Mrs.V. Suryakala[a] ,[a]Assistant Professor,[1234]Students

Department of Electronics and Communication Engineering

Sanketika Institute of Technology and Management, PM Palem, Visakhapatnam, India

---***---

Abstract - The goal of path design is to find the shortest and most efficient obstacle-free route from a starting point to a destination state. A map of the surroundings, as well as the start and target states, are needed for path planning. Path planning applications are such as Automated robot navigation, autonomous vehicle Robotic surgery, digital animation of characters, and others. Different algorithms provide different solutions to this problem, to find the best path in terms of distance and smoothness (minim number of rotations); the smoothness means decreasing power consumption since the rotations take a lot of power to be executed. A traditional genetic algorithm is used to find the best path, and then modification is used to improve the path's characteristics. The experimental results obtained using MATLAB Simulator indicate that the enhanced approach applied in the genetic algorithm provides much better outcomes, the path edges are minimized along with the path length.

Key words: Path Planning, Genetic Algorithm, Mobile Path

1. INTRODUCTION

The importance of mobile robots has grown over time as they have been used for medical, industrial, educational, and transportation purposes, among other things. Path planning emerges as a rich area of research that can benefit and provoke exciting applications such as aircraft trajectory planning, cruise missile path planning, and others. The optimal path is not just the shortest path, it is also defined as a collision-free trajectory. Each path comes with specific constraints depending on the area of application. The optimal solution for one mobile robot or application may not be considered optimal for another due to several factors such as the physical characteristics of the robot, the application environment, and the constraints imposed by the task. These factors can be interpreted as the starting and finishing time, the robot's ability and performance, and the number of nodes and obstacles ahead.

Several algorithms have been proposed for path planning, including Genetic Algorithm (GA), A* Algorithm, Ant Colony Optimization (ACO), and Simulated Annealing (SA), each with unique advantages. Then A* Algorith m, originally developed for use in gaming applications, has since been widely used in robotics, intelligent transportation, and automatic control A* Algorithm operates by using heuristics to efficiently determine the shortest path between a start and goal position in a known environment. It guarantees optimal solution but may require significant computational resources when navigating complex, high- dimensional spaces.

Genetic Algorithms (GA), on the other hand, provide global optimal solutions and handle constraints effectively. GA is advantageous in dynamic environments where the topology may change, requiring the algorithm to adjust and optimize paths iteratively. Traditional GA works by iterative evolution using a population of candidate solutions, with selection, crossover, and mutation operations refining the paths over multiple generations. GA has been successfully combined with other artificial intelligence techniques in fields such as image processing and data mining, demonstrating its versatility.

While A* Algorithm is deterministic and guarantees the shortest path, GA allows for smoother paths with reduced rotations, which is beneficial for reducing power consumption in mobile robots. Some research efforts have attempted to improve A* Algorithm by modifying its heuristics for smoother navigation, but it remains sensitive to computational complexity when dealing with dynamic or cluttered environments. A hybrid approach that integrates A* Algorithm for initial pathfinding and GA for smoothing and optimization could provide an enhanced path planning solution.

2. GENETIC ALGORITHM FOR PATH PLANNING.

Genetic algorithm is iterative algorithm work through several generation, each generation consist of multiple solutions these solutions called chromosome and each chromosome consist of several genes as shown in the Figure 1.

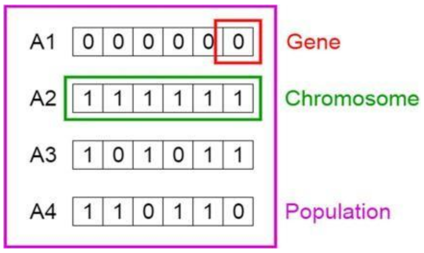


Figure 1. *Genetic algorithm.*

In the following section, GA is applied. Every GA needs to be defined. First, how is the chromosome going to be coded? Second, the population initialization which is a critical phase that impacts the outcome. Third, the needed fitness function to express the problem properly, finally, the GA operator's selection, crossover, and mutation. GA is basically working through multiple iterations (search) to find the optimal solution under certain constraints.

2.1 Environment Modeling

The studied environment in this paper will be static, random, and 2-D planer graphics with arbitrary irregular Figure 2 polygons. The safe distance will be the point from the start to the finish point without hitting any obstacles. A 2-D planar graphic, with varying levels of complexity, is used.

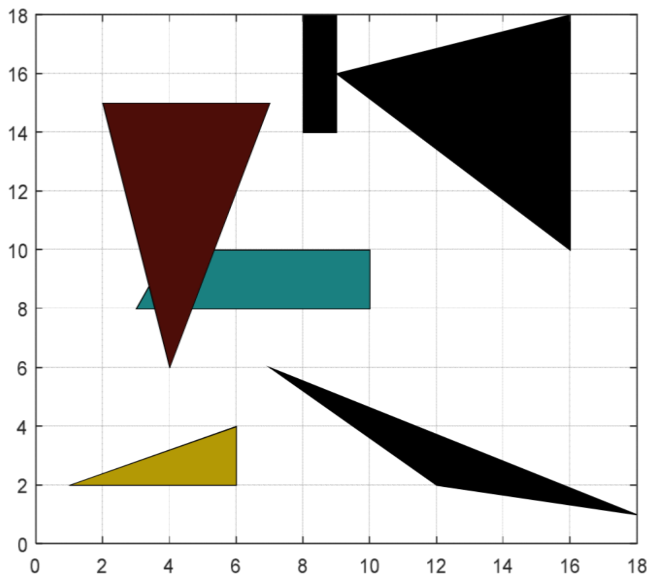


Figure 2. *2-D planer graphics with arbitrary irregular polygons.*

First, confirm that you have the correct template for your paper size. This template has been tailored for output on the A4 paper size.

2.2 Chromosome Encoding

There are s multiple scenarios of chromosome coding, it could be real, binary or tree coding etc. In this paper real coding chosen due to multiple reasons, firstly no time wasted in encoding and decoding these chromosome, secondly sometimes binary encoding falls under something called hamming cliff which means that any two similar chromosomes when they are in binary seems to be most a part and there is no indication that they are similar. Thirdly real coding enables us to apply multiple kinds of operators to obtained new solutions (chromosomes). After choosing the proper chromosome coding, the chromosome itself is a collection of points, these points constitute the chosen path or the line segments that together combined given path, each chromosome will be a set of points, where each chromosome is different solution, these points called also nods. Each chromosome will be the path form source (S) to destination (D) through different nodes (, , ,…,), each node is a point of (x, y) coordinates shown in Figure 3.

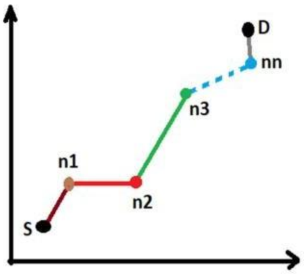


Figure 3. *Chromosome genes (nodes).*

2.3 Population Initialization

Population initializations are a very important step which can speed up converging time, and a void being trapped in local minima. The initial populations are a collection of multiple possible solutions. Each solution is called a chromosome. Each chromosome is a collection of nodes. The node is a point with x and y that represents its coordinates in the 2-D plane, which connects a segment line along the path. For that the initial population will be a random number of coordinates that must be bound by the work space, which starts from the start point (S) and finishes at the finish point (D).

2.4 Fitness Function

The metric used to evaluate the solutions' feasibility is the fitness function; of course, any fitness function used in path planning must take path length under consideration. When a new set of solutions (chromosome) came up, each solution was tested against the fitness function. If its fitness was 0, then this solution hit an obstacle. Otherwise, it will be measured by the inverse of the path length.

2.5 Genetic Algorithm Operators

First, Select operator: which select best solutions based on fitness function, and make sure it will pass on to the next generation, the selection method is roulette (means fitness zero not passed on to next generation).

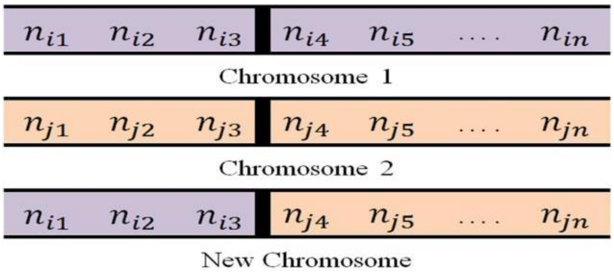


Figure 4. *Chromosome single crossover operators.*

Second, Crossover operator: which means two chromosome of the same generation used to give new off spring, by mixing their gene. Two types of crossover operator there is one and two point crossover. In this paper single point crossover used; shown in Figure 4.

Third, the Mutation operator: choose a solution with genes and mutate it. The mutation differs based on the gene coding. If it's binary, the ones become zeros and vice versa. If its real number is the complement of the number (gene) shown in Figure 5. The mutation is achieved randomly with a certain probability to ensure the diversity; the mutation probability in this paper is set up to be 0.01.

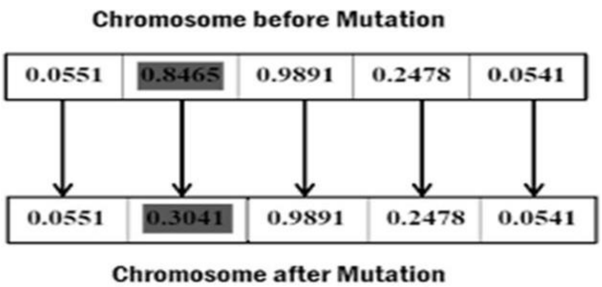
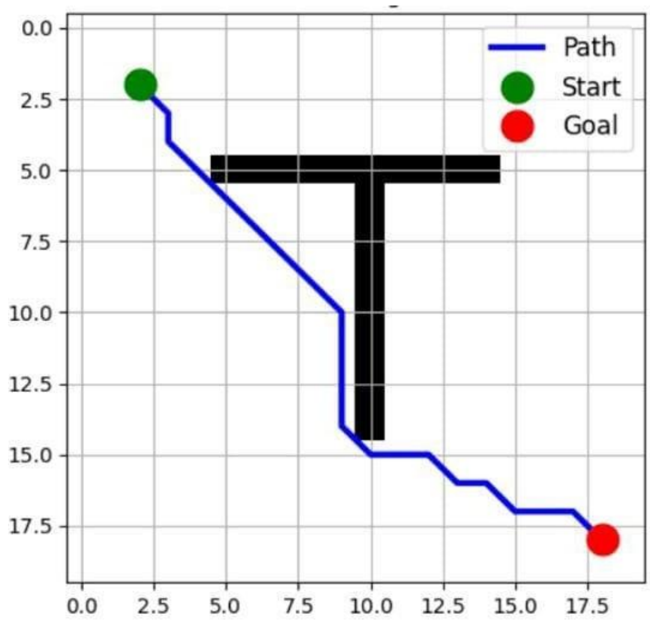


Figure 5. *Chromosome mutates operators.*

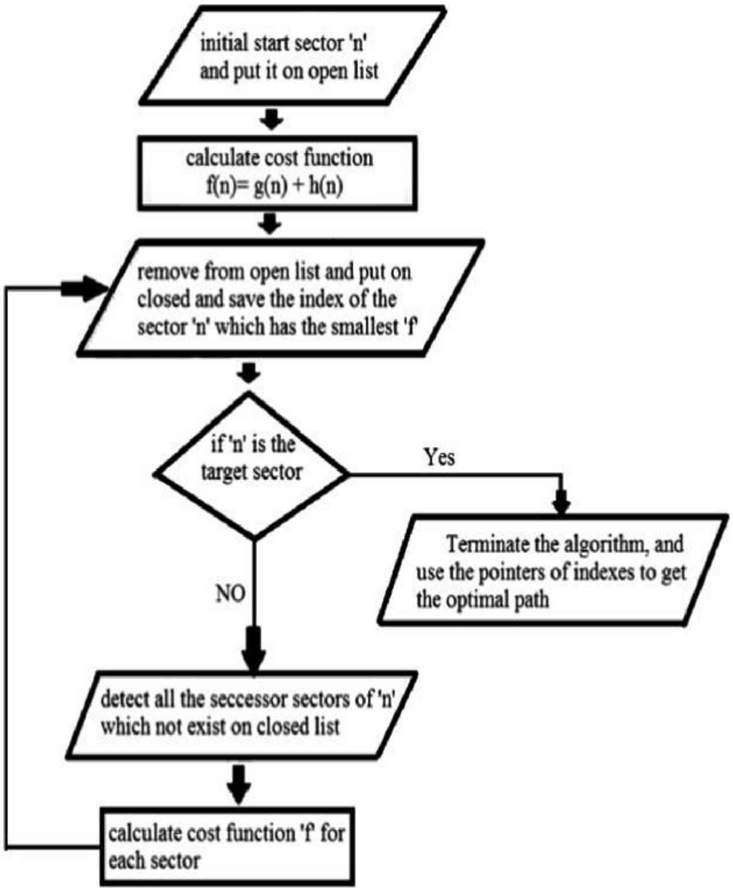
3. A* ALGORITHM

A star Algorithm is a method used to find the best or most efficient way to solve a problem, often in navigation, search, or optimization tasks.

One well-known example of this algorithm, which is used in computer science for pathfinding—like finding the shortest route in a video game or GPS navigation, also Solving Puzzles etc ...



3.1 FLOW CHART OF A STAR ALGORITHM



4. BENEFITS OF A STAR+GENETIC ALGORITHM HYBRID

1. **Faster and Smarter Pathfinding:** The hybrid approach balances speed and exploration, making it ideal for dynamic environments.

2. **Adaptability to Changing Environments:** Useful for autonomous vehicles, drones, and real-world navigation where roads or terrain can change.

3. **Optimized Solutions Beyond Shortest Path**

4. **Better Performance in Large or Complex Search Spaces**

5. METHOGOLOGY

This paper contribution is new gene operator added to the traditional work of genetic algorithm; this operator is added to enhance the GA performance. This operator consists of several steps which are: Path correction operator, vertx identification operator and gene realocating operator the whole process shown in Figure6.

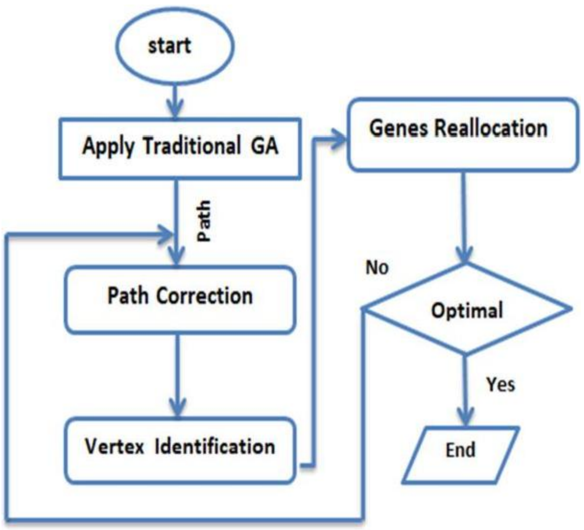


Figure 6. *Flow chart showing the methodology.*

Starting with the environment inserted, traditional GA is applied and the path is obtained. The path is taken along with the environment to the path correction operator, which will be applied as much as possible to provide smoothness and shorten the path. The vertex identification operator will be applied, the vertex becomes known and this information is used by the genes reallocation operator. If the modified path is still not optimal, re-do the path correction and vertex identification until the optimal features are achieved.

5.1 Path correction operator

Enhances the path by identifying direct lines between points (start and destination) to shorten the path and improve smoothness.

Reduces the path length by projecting a direct line between start (S) and destination (D) while avoiding obstacles.

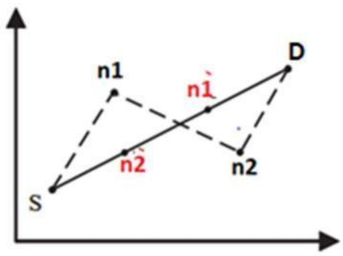


Figure 7. *path correction operator*

5.2 Vertex Identification Operator

After the path correction, the fitness is becoming better and better because the path is shortened. Obviously, another issue emerged which is the existence of large rotation angles, these angles needs further analysis. The path correction operator can't resolve this issue, since the nodes distributed along the path are not around these angles (vertex).

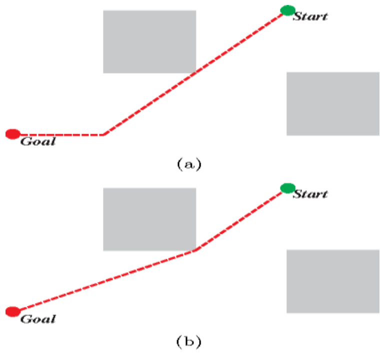


Figure 8. *vertex identification operator*

5.3 Gene Reallocation Operator

After identifying the vertex gene, the other nodes are called variable genes, called variables because they need to be reallocated and redistributed along the path. The purpose of reallocating is to be able to apply a path correction operator since the initial setup failed to be applicable to path correction. The reallocation applies in certain conditions. Firstly, the rotation angles, if they are large, then more genes need to be allocated there because more smoothing is required, and the smoothing, as we said earlier, happened using path correction operators. Secondly, the number of genes distributed on the left and right sides of the vertex is determined by the longest side (longest sides mean more genes). Then path correction can be applied again and again until no further smoothing can be accomplished.

In the following figure 8, in Figure (a) the two vertex identified the rest of the nodes called variable genes (< ,< , etc). Figure (b), the variable genes pulled closer to the vertexes, Figure (c) the path correction is applied the vertex 2 gone, and certain smoothing is obtained, the vertex one angle still less than 180 and need much more work and analysis, So more variable genes are pulled toward the vertex. The genes distrusted left and right the vertex according to the length. Figure (d) path correction is applied again and more smoothing

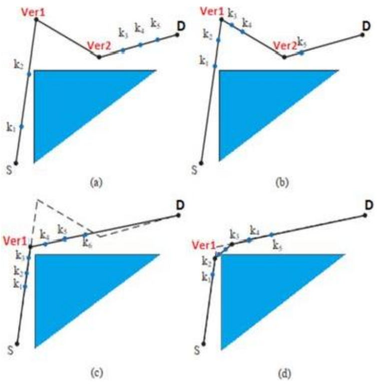


Figure 9. *Gene Reallocation*

6. SIMULATION EXPERIMENTS

Several 2-D environment model used to test our proposed system, in comparison with traditional Genetic algorithm, these environment are: irregular environment (IE), narrow winding environment (NWE) and complex maze environment (CME) which defined in [14, 15]. The GA parameters are: initial population size is 100; crossover probability is 0.8; mutation probability is 0.01; the number of generation is 30.

6.1 Irregular Environment

Figure 10 shows IE (irregular environment) model, the start point from (0, 0) and the finish point at (20, 20), in blue old traditional GA, and in red the modified enhanced GA through our operators. As its noticed the red path is much shorter and the genes allocated along critical areas where collision can occurred.

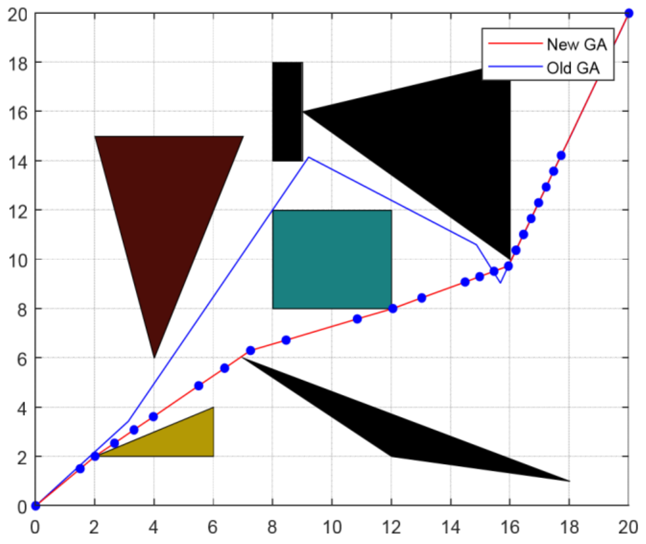


Figure 10. *Results on irregular environment.*

6.2 Narrow Winding Environment

Figure 11 shows the second environments mode which called narrow winding environment (NEW) with starting point (0, 0) and finishing point at (30, 30), is environment is the hardest since its contains too many angles, and consumes much processing time. We need to minimize the rotation angle besides the path length, as its shown on the figure the Enhanced work shows better performance in term of rotation angles and in terms of path length.

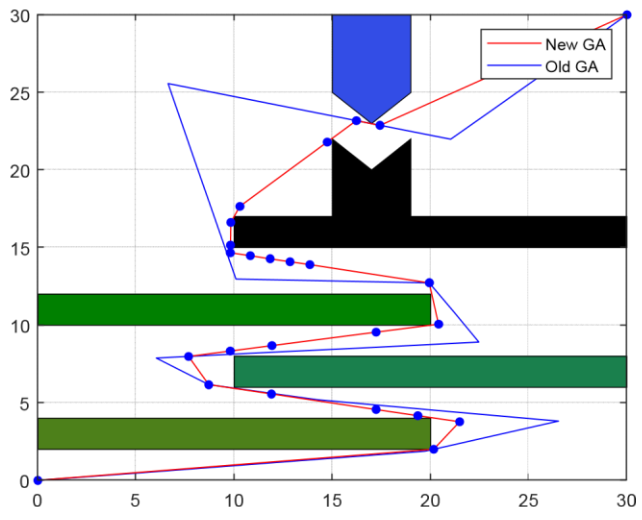


Figure 11. *Results on narrow winding environment.*

6.3 Complex Maze Environment

Finally; compel maze environments (CME) model which shown in Figure 12. The start point at (0, 0) and the finish point at (45, 45), the path as smooth as possible, the genes concentrated around the corners, and of course to Collision happening.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 04 | April - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

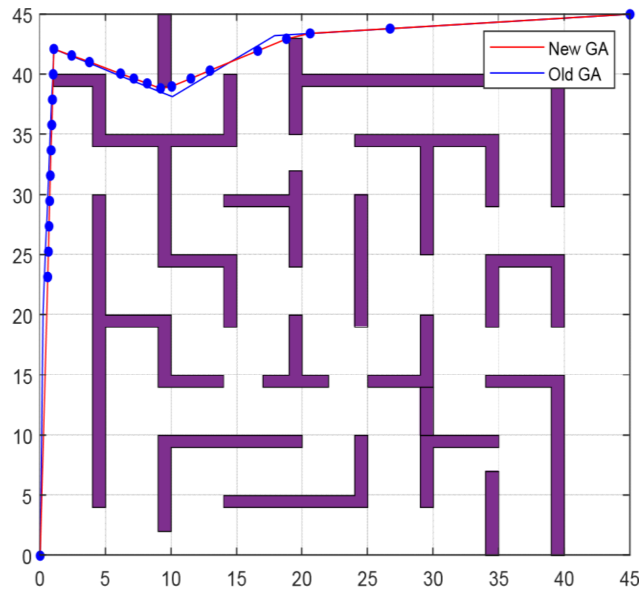


Figure 12. *Results on compel maze environments.*

In Table 1, The number of genes distributed among the trajectory was 25 genes with c (constant in the fitness function) is 1.5 equation (3), as shown in Table 1 the path before and after the enhancement process is much shorter, the path is much smoother (minimum rotation) which means less power consumption. For choosing the proper C value, several values chosen, c=1.5, 10, 20 in Table 2 and Table 3.

Table 1. *The Results with # of genes=25, c=1.5.*

Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	37.1643	266.3442	30.0056	74.3501
NWE	126.9587	766.1434	94.9717	605.9191
CME	88.5064	197.8526	87.7036	177.3879

Table 2. *The Results with # of genes=25, c=10.*

Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	33.2453	139.2297	29.9795	68.6291
NWE	114.7109	698.0758	91.0472	638.8225
CME	81.7010	404.2745	77.9695	219.3877

In Table 2, The number of genes is fixed to 25, the c in the fitness is changed to 20.

Table 3. *The Results with # of genes=25, c=20.*

Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	32.7884	127.8072	29.9143	70.6130
NWE	121.2033	753.6703	91.0861	702.9812
CME	94.5215	469.5714	77.8433	221.4575

As it's clear from Table 2 and Table 3 that higher c better, but not shown significantly, for that whatever c will be the effect insignificant.

Table 4. *The Results with # of genes=30, c=1.5.*

Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	32.7599	153.2898	29.9727	70.8815
NWE	128.2248	691.4552	88.5098	651.9988
CME	89.0326	204.4110	85.1827	164.9513

No trying to change the number of genes, 25, 30, 50 and 100 shown in Tables 4, 5 and 6. The more the number of genes the more path become smoother and path length is better with more genes than 25, after the number of genes become 30 no significant changes in the path length.

Table 5. *The Results with # of genes=50, c=1.5.*

Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	33.1975	148.2627	29.9169	70.5937
NWE	132.1320	734.3380	84.7895	651.0391
CME	129.0956	745.7954	77.7448	210.0316

In Table 6, the number of genes increased, the path is smoother with slight difference from the table 5 where number of genes to be 50. After certain number of genes no more smoothing can happen. This is highly dependable on the type of environment.

Table 6. *The Results with # of genes=100, c=1.5.*

Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	32.0902	119.8864	29.8784	70.0165
NWE	106.8251	652.5868	84.4018	614.9595
CME	91.1346	340.1424	85.1109	161.8311

In Figure 13, it is shown that the more genes, the smoother the path, but the more genes, the longer the execution time. That the more complex the environment, the greater the number of genes must be to be able to escape the path if needed. That the more complex the environment, the greater the number of genes must be to be able to escape the path if needed It is clear from the previous comments and from the numerical results in the tables that the enhanced GA with the suggested added operates decreases the angles summation (the path is more smooth) along with making the path much shorter. The process of speeding up the convergence is done by multiple procedures. First, the population initiations which were executed in the related work space only. The path correction operator eliminates the need for unneeded angles, which can be avoided by doing simple steps. The operator that might consume time will be the gene-reallocated operator. Gene reallocation means bringing the genes to the areas that need work (vertex, angles). The number of genes must not be fixed, but must be related to the environmental complexity. The C constant must also not be a static value; it should be a number related somehow to the longest path from the start to finish.

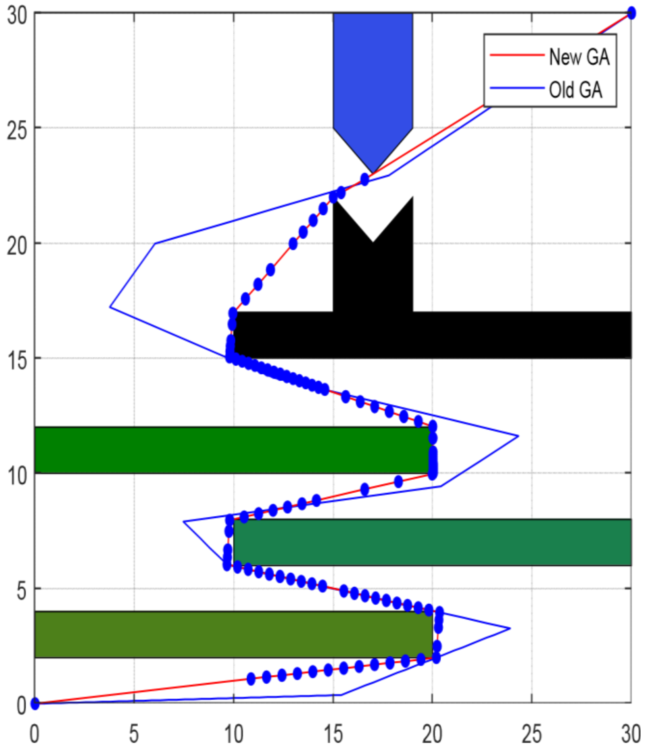


Figure 13. *The NWE with 100 genes.*

7. CONCLUSION

The combination of Genetic Algorithm (GA) and A star Algorithm creates a powerful hybrid that balances exploration and efficiency in pathfinding and optimization problems. A star is precise and fast, making it ideal for finding optimal paths in structured environments. GA is adaptive and flexible, allowing it to explore multiple solutions and handle dynamic, complex environments.The hybrid approach overcomes A star's local optima limitations and GA's slow convergence, resulting in a more robust and intelligent search strategy.

8. FUTURE SCOPE

- Autonomous Vehicles & Smart Transportation
- Self-driving cars
- Smart traffic systems
- Robotics & AI Navigation
- Drones and robots
- Video Game AI & Virtual Environments
- Space Exploration & Satellite Navigation
- Healthcare & Biomedical Applications
- Smart Agriculture & Resource Management

9. REFERENCES

[1] Qinggang Su, Wangwang Yu and Jun Liu. (2021), "Mobile Robot Path Planning Based on Improved Ant Colony Algorithm" 2021 Asia-Pacific Conference on Communications Technology and Computer Science (ACCTCS).

[2] Chunyu Ju, Qinghua Luo and Xiaozhen Yan. (2020). "Path Planning Using an Improved A-star Algorithm", 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), 2166-5656/20/$31.00 ©2020 IEEE, DOI 10.1109/PHM-Jinan48558.2020.00012.

[3] Chaymaa Laminia, Said Benhlima and Ali Elbekria, (2018) "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning", The First International Conference On Intelligent Computing in Data Sciences,-2018 Published by Elsevier B. V.

[4] Lee, J., & Kim, D. W. (2016). An effective initialization method for genetic algorithm-based robot path planning using a directed acyclic graph. Elsevier Science Inc.

[5] Tuncer, A., & Yildirim, M. (2012). Dynamic path planning of mobile robots with improved genetic algorithm. Computers & Electrical Engineering, 38 (6), 1564-1572.