Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Optimal Path Planning Approach for Static and Dynamic Using Genetic Algorithm

Kavya Ravishankar, Puspha Devaraj, Sharath Kumar Y H
Department of Information Science & Engineering, Maharaja Institute of Technology Mysore, Mandya, Karnataka 571477, India
**Correspondence: Kavya Ravishankar (kavyar.0821@gmail.com)*

ARTICLE INFO

Received: 26 Dec 2024

Revised: 14 Feb 2025

Accepted: 22 Feb 2025

ABSTRACT

Genetic algorithms are part of evolutionary computing, which is a rapidly growing area of artificial intelligence. Genetic algorithms are inspired by Darwin's theory of evolution. These processes mimic those in nature in such a way that subsequent populations are fitter and more adapted to their environment. As time and generations progress, they become better suited to their environment and if sufficient time is given they provide better and more optimal solutions. Starting from via even source to reach multiple goals is being proposed in generating an optimal trajectory. A sample is randomly selected from the configuration space. Each data point is represented as a step that helps the n traveling of a robot. By initiating the search process, the pursuit calculation endeavors to achieve the objective by developments, and from that position assisted regions are investigated. Simulation results demonstrate reaching the desired goal starting from a source in a dynamic environment.

Keywords: dynamic, pursuit, trajectory, environment

I. INTRODUCTION

In today's technology, vehicles have shown progress and have evolved in all types of transport applications. An automated Vehicle is an automatic transport vehicle that commutes objects or humans across locations without human command. The key element in the vehicle is to transport or to carry things safely avoiding accidents by understanding locations, and detecting source and destination locations by using global positioning systems. The challenge is to transport the object/human without the assistance of a driver or simply a driverless vehicle. The additional challenge is to plan their pathways to establish efficient routes. This route planning can be fixed or dynamic. To perform flexible transport in industrial production, vehicles are directed by following a line striping the floor or through the use of buried cords. The current system with its erstwhile commercial self-propelled vehicle uses a hidden cable from the primary stage. Planning is a term that means different things to different groups of people. Robotics addresses the automation of mechanical systems that would have sensing, actuation, and computation capabilities. A fundamental need in robotics is to have algorithms that convert the high-level specification of tasks from humans into low-level descriptions of how to move. The terms motion planning and trajectory planning are often used for these kinds of problems. The path-planning problem is usually defined as follows: "Given a robot and a description of its environment, a path is planned between two specific locations. The path must be collision-free (feasible) and satisfy certain optimization criteria". In other words, path planning is generating a collision-free path in an environment full of obstacles and optimizing it with respect to some criterion. Robotic mapping has been a highly active research area in robotics and AI for the last two decades. Robotic mapping addresses the problem of acquiring spatial models of physical environments through mobile robots. The mapping problem is generally regarded as one of the most important problems in the pursuit of building truly autonomous mobile robots. Despite significant progress in this area, it still poses great challenges. At present, robust methods are present only for mapping environments that are static and structured and of limited size. Mapping the unstructured, dynamic, or large-scale environments remains largely an open research problem. Path planning requires a clear understanding of the environment, representing the information in the environment in a system-understandable format and defining the configurations, building road maps for traversal, and finally

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

computing the optimal path.

II. RELATED WORK

With regard to static scenarios, there would be prior knowledge regarding the environment and the position of obstacles. Angus [1] solved path planning using Ant Colony Optimization and tested on the mound, volcano, valley terrains using both vector and multiplication combination methods. A popular swarm intelligence-based optimization algorithm used for path planning was the Imperialistic Competitive algorithm. Atashpaz-Gargari & Lucas [2] proposed an Imperialistic Competitive Algorithm inspired by imperialism for general optimization problems. Path planning can be viewed as a multi-objective optimization process with more than a single objective. The objectives could be the length, time, fuel economy, path safety. A non- dominated sorting genetic algorithm was proposed by Ahmed & Deb [3] for multi- objective optimization. The algorithm was tested using metrics like Hyper-volume and set coverage. The shortest path for Unmanned Combat aerial vehicles was found using the Imperialistic Competitive algorithm by Duan & Huang [4]. The partial concept of clustering was used in finding centers and distances between the centers. Among them, the shortest distance was calculated, and paths were made between clusters. Duchon et al. [5] proposed a Modified A* algorithm with modifications like theta*, phi*, Rectangular Symmetry Reduction, Jump Point Search and implemented on both symmetrical and asymmetrical environments. Ant Colony-based Decisional Navigation system was developed and implemented by Lazarowska [6] on USVs. A safe ship control including both the static and dynamic consideration was proposed through ACO. The finite Angle A* algorithm was suggested by Yang et al. [7] for path planning on satellite images using a modified Line of Sight with a branching factor of 16 for increasing the efficiency of the FAA* algorithm. Through the combination of genetic algorithm and the adaptive fuzzy logic controller, a hybrid algorithm was implemented by Bakdi et al. [8] for pathfinding on a two-wheeled indoor mobile robot.

The Piecewise Cubic Hermite Interpolating Polynomial is then applied on it to get a smoothened path. Huang et al. [9] experimented with a Time-Delayed Neural Network (TDNN) to find a globally optimal path on the online public Cordeau Instances and New York Instances. The principle behind this experiment was that the earliest auto-wave that hits the destination decides the shortest path. Comparisons with Dijkstra's and Pulse Coupled Neural Network proved the efficiency of TDNN based path planning. Fankhauser et al. [10] proposed an elevation mapping algorithm on real-time rough terrain. This algorithm was based proprioceptive localization from kinematic and inertial measurements. The algorithm was tested using four-legged robots and was able to reduce localization drift. A constrained multi-objective particle swarm optimization (MOPSO) was proposed by Wang et al. [11]. A rough terrain was initially modeled as workspace. The algorithm was simulated using a car-like mobile robot to provide a collision-free path with minimum distance and time. Han [12] proposed a COSPS (Critical Obstacles and Surrounding Point set) method that could lower the complexity in a three-dimensional scenario. A subset of grid points containing the obstacles in the 3D scenario is determined from the search space. They claim to reduce the search space by this containment, thereby reducing complexity. Zhong et al. [13] proposed a hybrid genetic algorithm – particle swarm optimization for path planning of automated guided vehicles with quay cranes and rail-mounted gantry cranes. The main aim of the algorithm was to determine the collision-free path avoiding deadlock conflicts. The algorithm was aimed at practical implementation at automated container terminals.

A probabilistic road map-based ant colony optimization approach was developed by Adolf et al. [14] for UAV 3D space planning. The experiments proved that a path planner could be integrated with a task planner for real-time uses. Willms et al. [15] proposed a real-time robot path planning system. The system was able to integrate safety margins around obstacles using the local penalty function and implemented it on various real-time simulations. Navigation planning in dynamic environments using a Genetic algorithm was proposed by Ucan et al. [16]. Simulations were done using USA waypoints of southern states. The algorithm achieved better results in terms of speedup, flexibility, quality in dynamic scenarios. Miao et al. [17] proposed an enhanced simulated annealing approach for dynamic robot path planning. Experiments were performed with both static and dynamic obstacles under different simulated environments. A different perspective in applying a heuristic method in routing and planning was experimented with by Yu & Wang [18]. A model-based fault diagnosis system through the vehicle steering of CyCab electric vehicle was instrumented to identify different faults abrupt fault, incipient fault, intermittent fault. The methods were classified into five categories based on their nature, and a comprehensive analysis of their complexity was also stated. Yang et al. [19] made an extensive survey on three-dimensional path planning algorithms. The algorithms were classified into five categories; sampling-based, nodal- based, mathematical-based, bio-inspired based, and multi-fusion based algorithms. The canonical definition, working, shortcomings, advantages of each algorithm has been detailed very extensively. The work addressed three queries pertaining to the taxonomy of 3D path planning algorithms, their working, and if they are suitable or not suitable to the application. An improved gravitational search-based multi- robot path planning algorithm in a dynamic environment was proposed by Das et al. [20]. The positions were updated using particle swarm optimization and greedy strategy. The simulations were done in C and Khepera II environments. Yin et al. [21] proposed a multi-objective path planning system for dynamic urban environments. Safety index maps were designed for both static and dynamic obstacles, using which the position of the obstacle was predicted. Extensive simulations on both synthetic and real-time datasets have been performed to verify the efficiency of the system. Elhoseny et al. [22] proposed a modified genetic algorithm for robot path planning for behavior analysis in dynamic environments. An effective routing protocol for vehicular ad hoc networks based on fuzzy logic, namely F-Ant, was proposed by Fatemidokht et al.[23], considering challenges like self- organization, dynamicity of vehicles. The simulations were done in the NS-2 simulator. The proposed method was able to achieve a high packet delivery ratio and low end-to-end delay. Zhang et al. [24] proposed an improved ant colony optimization method with a strengthened pheromone updating mechanism for solving constraint satisfaction problems. A set of benchmark CSP test cases were used for simulation and achieved better results based on convergence speed and quality of the solution. A stochastic dynamic environments-based path planning system was proposed by Subramani et al. [25] by considering the risks in uncertain and dynamic scenarios. Initially, the probability distribution of environmental flows and risks was predicted. The path with minimum risk is chosen as the optimum path. The system experimented with coastal and urban environments. Luo et al. [26] proposed an improved ant colony-based method for path planning of mobile robots. An improvement was made on pheromone initialization, pheromone update, deadlock avoidance, and simulations made in MATLAB. An automated weight updating grid-based path planning system was developed by Fransen et al. [27] for collision-free path planning for automated guided vehicles. The approach was a dynamic approach with grid-based representation where the vertex weights are updated under regular time intervals. The approach was found to have increased throughput with faster recovery from deadlock situations. An improved ant algorithm for dynamic path planning to handle traffic congestion was proposed by Wu et al. [28]. Other than the conventional distance factor, the algorithm considered factors like road length, incoming and outgoing traffic, road grade, number of lanes for path planning. Particle swarm optimization was used to optimize the parameters of ACO. The algorithm was tested in intersections of the Beijing area. Yuan et al. [29] proposed an RRT (randomly exploring rapid trees) cache method for path planning in real-time environments. Replanning of paths was done using the multi- query sampling technique with the relay node method. The method was verified for both static and dynamic scenarios in ROS with Aubo-i5 manipulator. The method quantifies the main attributes of urban road length, number of lanes, incoming and outgoing traffic flow.

III. PROPOSED GENETIC ALGORITHM

Figure 1 shows the proposed genetic algorithm work flow. The first step consists of the generation of the initial population. The fitness of each chromosome in the population is then evaluated. The fitness function used here considers both collision avoidance and smoothness of the path. Selection is done by the Roulette wheel method. There is a chance that the best solution (chromosome) is lost in the selection process. So a technique called Elitism is used here. Elitism is used to keep track of the fittest chromosome obtained during the process and ensures that the fittest chromosome is present in the forthcoming generations. This is followed by the application of reproductive operators. The proposed GA uses single point crossover and mutation. Thus a new population of chromosomes for the next generation is obtained. The process is repeated for the next generations. This is done iteratively for “n” generations until the algorithm converges to a single solution. This solution depicts an optimal path (that is both short and smooth) the robot can take. The modified genetic algorithm yields an optimal path in fewer generations than the traditional GA.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

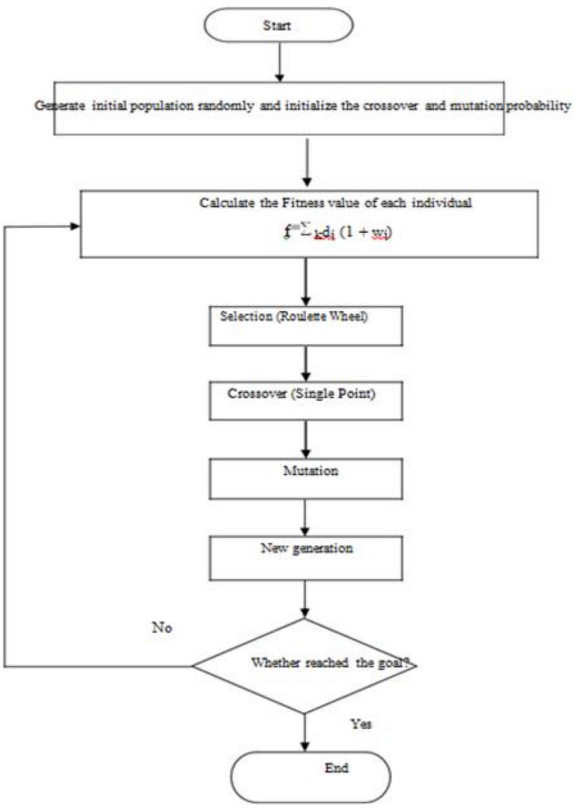


Figure 1: Flow chart for Genetic Algorithm

Step 1: Generating initial population.

The initial population is generated randomly.

Step 2: Optimization process.

Optimization consists of minimizing the Euclidean distance of the path.

Step 3: Fitness Function.

The fitness function used is applied to the chromosomes, the fitness function formula is as follows

dj	-	is the distance between two adjacent genes,
Wj	-	is the damping coefficient of jth gene
k	-	is the total grids a mobile robot will move over
d	-	1, if horizontal or vertical direction, 0.414 if diagonal

Step 4: Selection Process

The Roulette wheel selection is used to select the next generation. The concept of elitism is used to conserve the best solutions of each generation.

Step 5: Crossover operator

Crossover is used to combine and produce new chromosomes.

The crossover techniques used here is single point crossover and the crossover probability is 0.9. The mutation operator in a particular gene is the chromosome gets changed. The Mutation probability is 0.02.

Step 6: The algorithm termination condition

The algorithm terminates when it has run for a specific number of generations.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

IV. GENETIC ALGORITHM UNDER VARIOUS CONSTRAINTS

The source and the destination location are obtained from the user and the corresponding nodes are located in the two dimensional grid environment.

GA Agent in Obstacle Free Environment

The initial population which is the possible path between the given source and the destination is generated and displayed in the environment. For each chromosome, the fitness function is evaluated and the fitness values are sorted out in ascending order. Then each time two chromosomes from the initial population are selected as parents and crossover is performed to produce two off springs. This process is repeated for all the chromosome in the initial population .Then mutation is performed for each chromosome.

To calculate the Fitness: (Obstacle free environment)

```
BEGIN
  for all chromosomes in the initial population ; for all genes in
    the chromosome ;
      Calculate fitness using the Eqn (5.4) if the next gene is
     free in the path
           wj=0;
     elseif the next gene position is the target wj=-0.9;
     endif
    if the direction of moment is horizontal || vertical dj=1;
    else
           dj=0.414;
    endif endfor
  endfor
END
```

GA Agent with Static Obstacles

The obstacle location is fixed in the grid environment. While generating the initial population (which is the possible paths between the given source and the destination), the presence of the obstacle was detected. For each chromosome, the fitness function is evaluated and the fitness values are sorted out in ascending order. The path with the obstacle will have the highest fitness value. Then each time two chromosomes from the initial population are selected as parents and crossover is performed to produce two off springs. This process repeated for all the chromosomes in the initial population. Then mutation is performed for each chromosome.

To calculate the Fitness: (In the Obstacle present in the environment)

```
BEGIN
   for all chromosomes in the initial population ; for all genes in
     the chromosome ;
       Calculate fitness using the Eqn (5.4)
      if the next location is obstacle in the path wj=300;
      elseif the next gene position is the target wj=-0.9;
      endif
    if the direction of moment is horizontal || vertical dj=1;
    else
           dj=0.414;
     endif endfor
 endfor
END
```

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

4.1 GA Agent with Dynamic Obstacles

The obstacle location is chosen in random in grid environment. While generating the initial population (which is the possible paths between the given source and destination), the presence of obstacle was detected. For each chromosome the fitness function is evaluated and the fitness values are sorted out in ascending order. The path with the obstacle will have the highest fitness value. Then each time two chromosomes from the initial population are selected as parents and crossover is performed to produce two off springs. This process is repeated for all the chromosomes in the initial population. Then mutations are performed for each chromosome. A chromosome is the complete path from the robot start position "S" to the target position "T" .A genome is a point (vertex of the obstacles in the modeled environment), which the path (chromosome) could be completely made by connecting them sequentially. According to the problem formulation, the length of the algorithm chromosomes (number of genomes) is dynamic. A fixcd-length chromosome is not suitable for a complex environment, especially, in a dynamic environment. In order to reach the target, in a more obstacles environment, longer chromosome may be needed which means a fixed- length chromosome may not be enough. Therefore, variable-length chromosome is better suitable in a dynamic and many obstacles environment. In the proposed model, a novel approach with a variable-length chromosome is proposed. Using this proposed method, a mobile robot can find the shortest path in a static environment and the near-optimal obstacle-free path in a dynamic complex environment. In the above three cases, new population of chromosomes for the next generation is computed with different fitness values. And for the next generation, fitness is computed and also the crossover, mutation process is done. This process is repeated for 'n' generations. Finally, the shortest optimal path is obtained for the selected source and goal.

V. RESULTS

The approach was tested by means of simulations. A number of scenarios were generated from easy to more challenging ones. The simulations were performed on a system with Intel i3 processor (2.2 GHz) with 3 GB RAM. Two scenarios are discussed here. The first scenario consisted of a simple single obstacle in the middle of the paths of robots from source to goal. Robots were generated at the corners which were supposed

to travel to the other corner. The graph started the generation process from the four corners and continued till the four sub-graphs met and later the process continued exploring the entire robotic map. Figure 2 shows the map with the source and goal with static obstacles. Figure 3 shows the source and goal with dynamic obstacle added to map 1. Finally, in figure 4 shows the Resultant Path. Similarly, Figure 5,6 and 7 shows the source and goal with dynamic obstacle added to map 2 and Resultant Path. Figure 8,9 and 10 shows the source and goal with dynamic obstacle added to map 3 and Resultant Path. Table 1 shows the results of 8 maps before dynamic obstacles and also shows Path length 1,2,3 and Processing time. Table 2 shows the results of 8 maps after adding dynamic obstacles and also shows Path length 1,2,3 and Processing time. Figure 2,5,8 shows the various possible paths to reach the target from the source in the grid environment. The initial population is generated by detecting the presence of static obstacles. It shows the shortest optimal path for the robot to reach the defined target in the presence of static obstacles. While generating the initial population, the obstacle locations are detected. The optimal path is obtained by calculating the fitness for the initial population. Then after performing crossover and mutation, the optimal path is obtained. Figure 3,6,9 shows the grid environment with the moving obstacles. In the grid environment, the user defined source location; target location and the obstacle location were chosen in the grid environment. It shows the various possible paths to reach the target from the source in the grid environment. The initial population is generated by detecting the presence of Dynamic obstacles. The comparison of the number of generations to reach the optimal path was shown in Figure 4,7,10. Analysis also shows that the execution time decreases as the number of generations increases. It is clear that the algorithm shows the vast difference for a dynamic environment whereas there is a only light differences in the execution time when the number of generations.

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/
Research Article

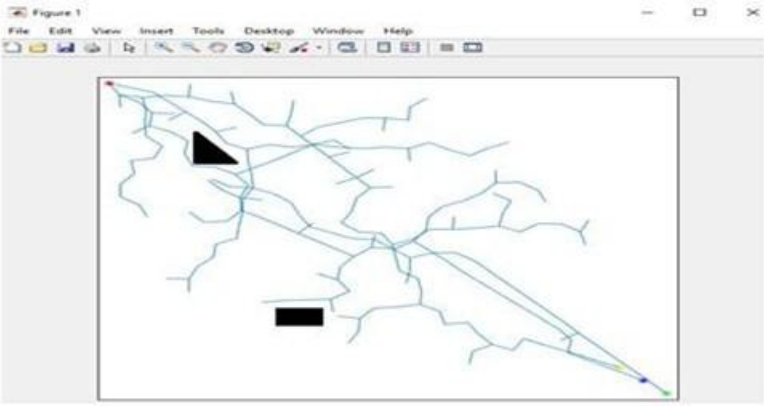


Fig 2: Map 1- Path generated from source to goals

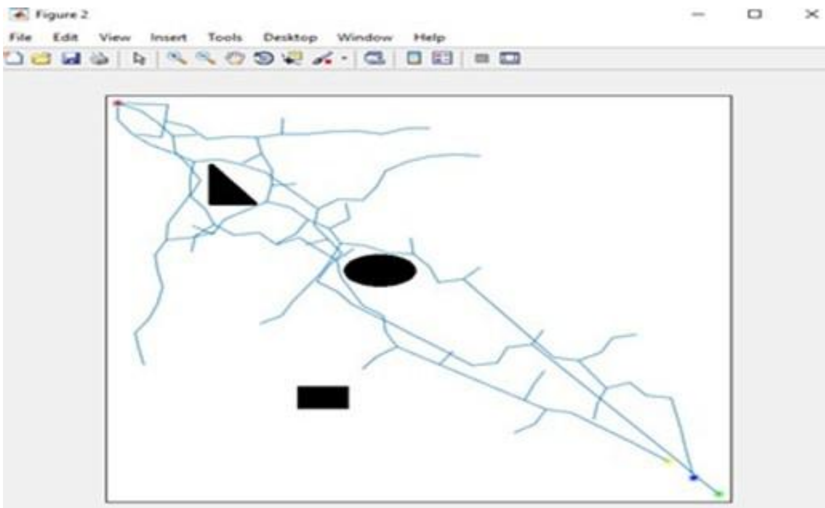


Fig 3: Map 1- Path obtained after adding dynamic obstacle

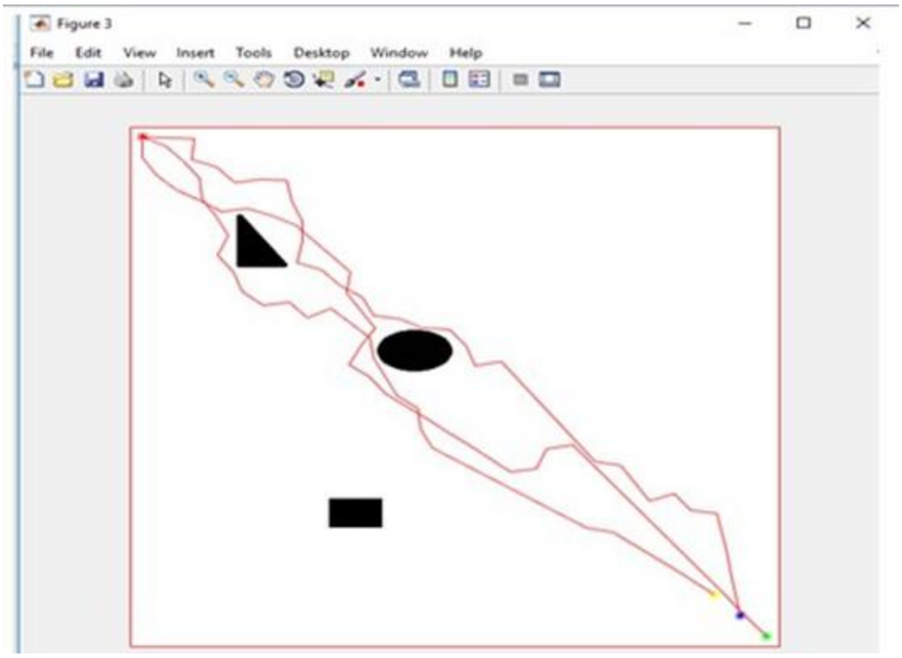


Fig 4: Map 1- Resultant Path

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

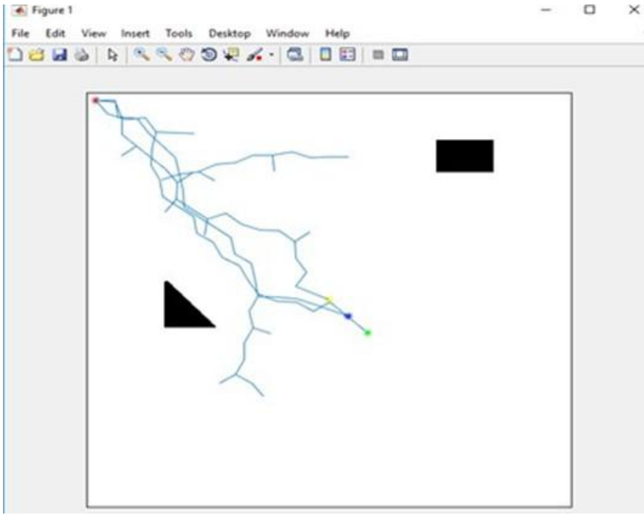


Fig 5: Map 2- Path generated from source to goals

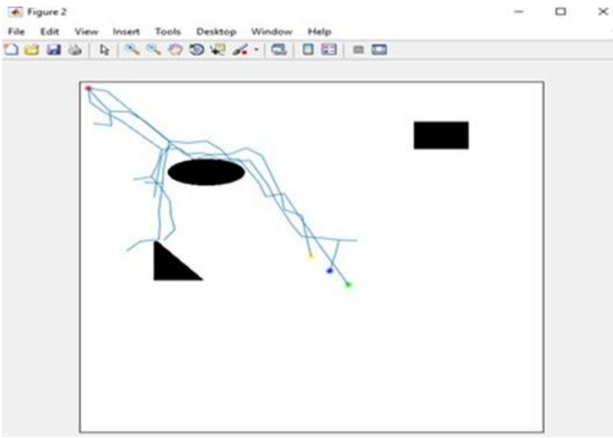


Fig 6: Map 2- Path obtained after adding dynamic obstacle

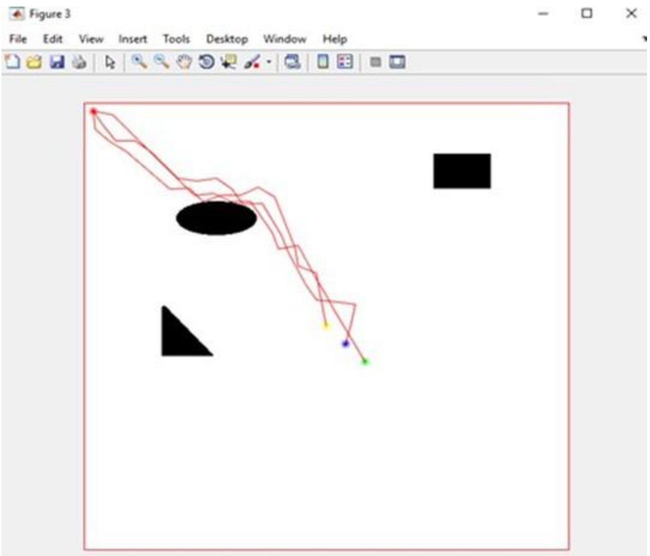


Fig 7: Map 2- Resultant Path

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

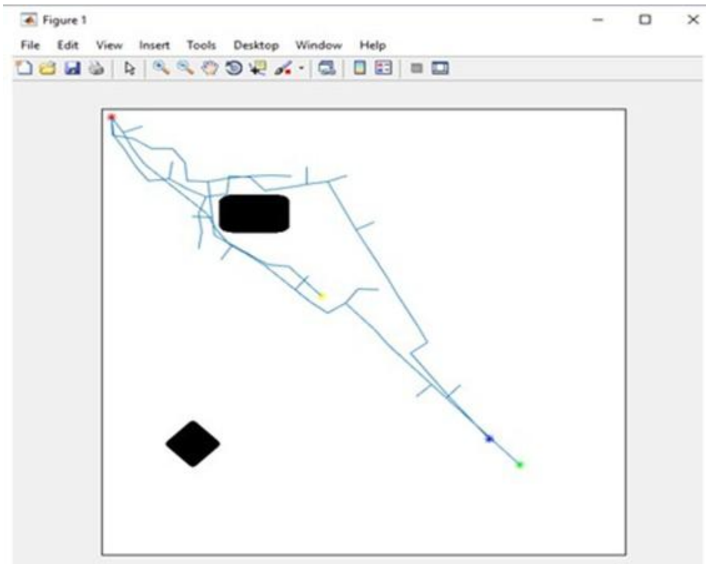


Fig 8: Map 3- Path generated from source to goals

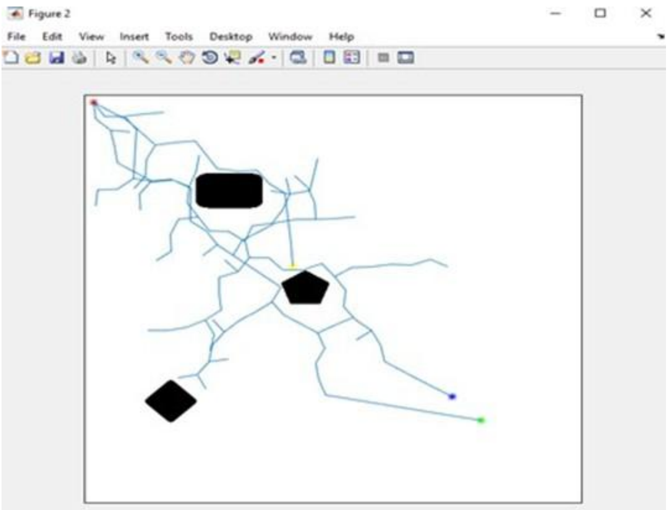


Fig 9: Map 3- Path obtained after adding dynamic obstacle

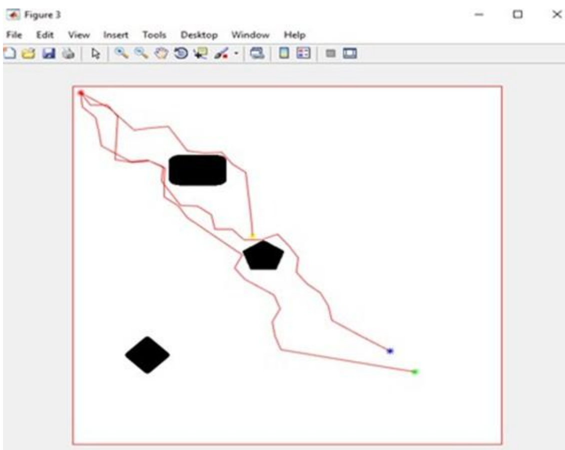


Fig 10: Map 3- Resultant Path

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 1: Experimental Results before generation of dynamic obstacle

	Path Length 1	Processing Time(secs)	Path Length 2	Processing Time(secs)	Path Length 3	Processing Time(secs)
Map 1	759	4.7750	746	5.5954	729	5.0637
Map 2	401	0.9817	420	2.6707	457	1.6374
Map 3	327	0.8317	628	1.6713	570	2.0759
Map 4	115	0.2211	280	0.4811	746	2.4028
Map 5	66	0.1604	284	0.2717	519	0.5324
Map 6	651	0.1824	321	0.6612	737	4.2408
Map 7	169	0.2812	338	0.5366	230	0.3760
Map 8	103	0.3546	375	0.7767	825	1.8195

Table 2: Experimental Results after generation of dynamic obstacle

	Path Length 1	Processing Time(secs)	Path Length 2	Processing Time(secs)	Path Length 3	Processing Time(secs)
Map 1	685	2.5979	766	5.1327	784	5.4029
Map 2	405	1.1825	429	1.8193	432	1.5052
Map 3	331	0.7367	586	3.2394	664	4.3957
Map 4	114	0.2188	259	0.4595	755	2.8565
Map 5	63	0.1056	318	0.3005	585	1.1546
Map 6	95	0.1802	305	0.5651	750	5.98
Map 7	188	0.3609	422	9.5571	256	0.8949
Map 8	96	0.2209	376	1.8879	610	2.0076

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table 3: Experimental Results of GA

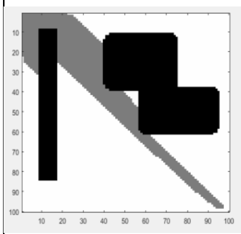
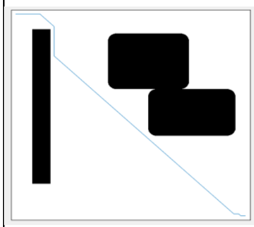
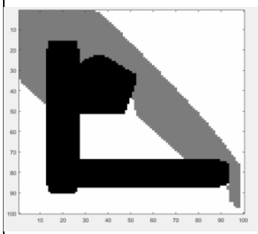
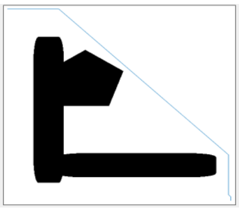
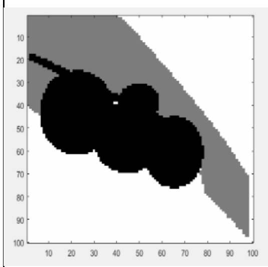
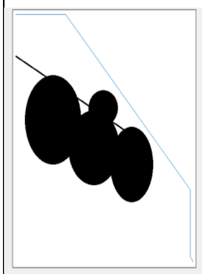
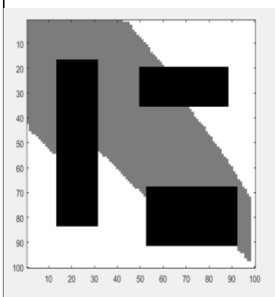
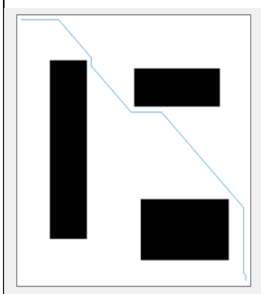
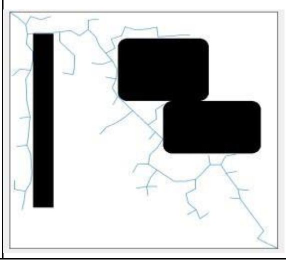
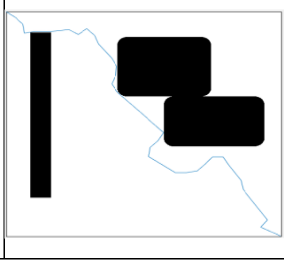
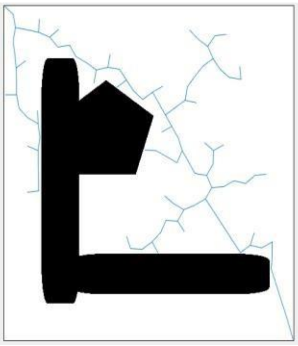
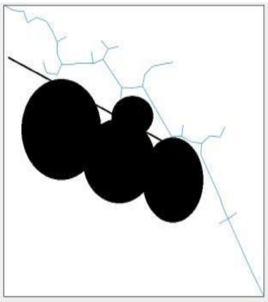
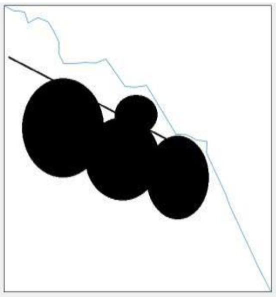
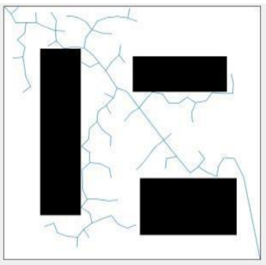
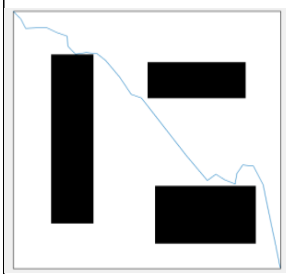
Sl.No.	Graph	Path	Path Length	Processing Time(secs)
1			719	4.184
2			743	4.816
3			757	4.719
4			763	5.189

Table 4: Experimental Results for RRT

Sl. No.	Tree	Path	Path Length	Processing Time(secs)
1			414	0.484

Journal of Information Systems Engineering and Management
2025, 10(50s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

2			**640**	**1.249**
3			**712**	**1.074**
4			**753**	**1.390**

VI. CONCLUSION

The proposed genetic algorithm agent incorporates domain specific knowledge into problem specific genetic operators. The developed GA also features an efficient evaluation method that is greatly beneficial for evolving good solutions from infeasible solutions. The most important modification applied to the algorithm is the fitness evaluation. Due to this modification, the optimal path is obtained in 8 generations. It is clear that the algorithm shows the vast difference for a dynamic environment, whereas there are only slight differences in the execution time when the number of generations more. The path cost decreases nearly linearly as the number of generations increases and again this strategy is the most effective for a dynamic environment. Dynamic obstacles with different shapes will be embedded with the capability to sense along by the robots. The GA work will be integrated with any local path planning algorithm, so that obstacle avoidance is predicted and avoided with the same. GA finds the optimal path selection in the global environment. The integration of local avoidance algorithm like the GJK will support the GA for local obstacle prediction and avoidance.

REFERENCES

1. Angus, D. (2005). Solving a unique Shortest Path problem using ant colony optimisation. Communicated by T. Baeck, (January), 1- 26.

2. Atashpaz-Gargari, E., & Lucas, C. (2007, September). Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. In 2007 IEEE congress on evolutionary computation (pp. 4661-4667). Ieee.
3. Ahmed, F., & Deb, K. (2013). Multi-objective optimal path planning using elitist non-dominated sorting genetic algorithms. Soft Computing, 17(7), 1283-1299.
4. Duan, H., & Huang, L. (2014). Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning. Neurocomputing, 125, 166-171.
5. Duchoň, F., Babinec, A., Kajan, M., Beňo, P., Florek, M., Fico, T., & Jurišica, L. (2014). Path planning with modified a star algorithm for a mobile robot. Procedia Engineering, 96, 59-69.
6. Lazarowska, A. (2014). Ant Colony Optimization based navigational decision support system. Procedia computer science, 35, 1013- 1022.
7. Yang, J. M., Tseng, C. M., & Tseng, P. S. (2015). Path planning on satellite images for unmanned surface vehicles. International Journal of Naval Architecture and Ocean Engineering, 7(1), 87-99.
8. Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., & Bouzouia, B. (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robotics and Autonomous Systems, 89, 95-109.
9. Huang, W., Yan, C., Wang, J., & Wang, W. (2017). A time-delay neural network for solving time-dependent shortest path problem. Neural Networks, 90, 21-28.
10. Fankhauser, P., Bloesch, M., & Hutter, M. (2018). Probabilistic terrain mapping for mobile robots with uncertain localization. IEEE Robotics and Automation Letters, 3(4), 3019-3026.
11. Wang, B., Li, S., Guo, J., & Chen, Q. (2018). Car-like mobile robot path planning in rough terrain using multi-objective particle swarm optimization algorithm. Neurocomputing, 282, 42-51.
12. Yang, L., Qi, J., Song, D., Xiao, J., Han, J., & Xia, Y. (2016). Survey of robot 3D path planning algorithms. Journal of Control Science and Engineering, 2016.
13. Zhong, M., Yang, Y., Dessouky, Y., & Postolache, O. (2020). Multi-AGV scheduling for conflict-free path planning in automated container terminals. Computers & Industrial Engineering, 142, 106371.
14. Adolf, F., Langer, A., e Silva, L. D. M. P., & Thielecke, F. (2007). Probabilistic roadmaps and ant colony optimization for UAV mission planning. IFAC Proceedings Volumes, 40(15), 264-269.
15. Willms, A. R., & Yang, S. X. (2008). Real-time robot path planning via a distance-propagating dynamic system with obstacle clearance. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 38(3), 884-893.
16. Uçan, F., & Altılar, D. T. (2012). Using genetic algorithms for navigation planning in dynamic environments. Applied Computational Intelligence and Soft Computing, 2012.
17. Miao, H., & Tian, Y. C. (2013). Dynamic robot path planning using an enhanced simulated annealing approach. Applied Mathematicsand Computation, 222, 420- 437.
18. Yu, X., Chen, W. N., Gu, T., Yuan, H., Zhang, H., & Zhang, J. (2018). ACO-A*: Ant colony optimization plus A* for 3-D traveling in environments with dense obstacles. IEEE Transactions on Evolutionary Computation, 23(4), 617-631.
19. Yang, L., Qi, J., Song, D., Xiao, J., Han, J., & Xia, Y. (2016). Survey of robot 3D path planning algorithms. Journal of Control Science and Engineering, 2016.
20. Das, P. K., Behera, H. S., Das, S., Tripathy, H. K., Panigrahi, B. K., & Pradhan, S. K. (2016). A hybrid improved PSO-DV algorithm for multi-robot path planning in a clutter environment. Neurocomputing, 207, 735-753.
21. Yin, C., Xiao, Z., Cao, X., Xi, X., Yang, P., & Wu, D. (2017). Offline and online search: UAV multiobjective path planning under dynamic urban environment. IEEE Internet of Things Journal, 5(2), 546-558.
22. Elhoseny, M., Tharwat, A., & Hassanien, A. E. (2018). Bezier curve based path planning in a dynamic field using modified genetic algorithm. Journal of Computational Science, 25, 339-350.
23. Fatemidokht, H., & Rafsanjani, M. K. (2018). F-Ant: an effective routing protocol for ant colony optimization based on fuzzy logic in vehicular ad hoc networks. Neural Computing and Applications, 29(11), 1127-1137.
24. Zhang, Q., & Zhang, C. (2018). An improved ant colony optimization algorithm with strengthened pheromone updating mechanism for constraint satisfaction problem. Neural Computing and Applications,

30(10), 3209-3220.
25. Subramani, D. N., & Lermusiaux, P. F. (2019). Risk-optimal path planning in stochastic dynamic environments. Computer Methods in Applied Mechanics and Engineering, 353, 391-415.
26. Luo, Q., Wang, H., Zheng, Y. He, J. (2019) "Research on path planning of mobile robot based on improved ant colony algorithm." Deep Learning of Big Data Analytics.
27. Fransen, K. J. C., van Eekelen, J. A. W. M., Pogromsky, A., Boon, M. A. A., & Adan, I. J. B. F. (2020). A dynamic path planning approach for dense, large, gridbased automated guided vehicle systems. Computers & Operations Research, 123, 105046.
28. Wu, C., Zhou, S., & Xiao, L. (2020). Dynamic Path Planning Based on Improved Ant Colony Algorithm in Traffic Congestion. IEEE Access, 8, 180773-180783.
29. Yuan, C., Liu, G., Zhang, W., & Pan, X. (2020). An efficient RRT cache method in dynamic environments for path planning. Robotics and Autonomous Systems, 131, 103595.