Genetic Algorithm-Based Path Planning for Autonomous Mobile Robots

Areej Alabbadi
Dept. of Electrical Engineering
Princess Sumaya University for Technology
Amman 11941, Jordan
are20208172@std.psut.edu.jo

Awos Kanan
Dept. of Computer Engineering
Princess Sumaya University for Technology
Amman 11941, Jordan
a.kanan@psut.edu.jo

2023 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) | 979-8-3503-2405-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/JEEIT58638.2023.10185855

***Abstract*—In this paper, a Genetic Algorithm is used to solve the path planning problem for autonomous mobile robots in static environments. The goal of the path planning problem is to find a valid and practical path between two points while avoiding obstacles and optimizing a number of criteria including path length, safety, and distance from obstacles. A quality function is proposed to evaluate the optimization approach for different scenarios. Experimental results show that enhanced solutions can be achieved in less time using optimal values of the search algorithm parameters.**

***Index Terms*—Path Planning, Evolutionary Algorithms, Genetic Algorithms, autonomous robots, Metaheuristic Optimization.**

I. Introduction

Missions requiring autonomous robots and vehicles have included exploring Mars, operating nuclear power reactors, and keeping watch over hostile forces on the battlefield. The development of more intelligent autonomous mobile robots for future battles to minimize human casualties is one of these applications. Path planning in violent situations ,which is the application of the paper's subject, is one of the most significant challenges facing the development of such systems. It is important to note that path planning is essential for achieving autonomous control of the mobile robot which supports the efficient use of it. Realizing effective path planning in complex environments with various obstacles and uncertainty while taking into account various requirements such as obstacle avoidance, trajectory validity, real-time planning capability, and a sufficient path length, is still a significant challenge [1].

A genetic algorithm (GA) [2] is a search technique that relies on the survival of the fittest individuals among a set of individuals in each generation. The mobile robot path planning problem can be formulated as a GA-based search process with the solutions encoded as chromosomes. Genetic operators, including selection, crossover, and mutation, are repeatedly applied in a stochastic way to generate new solutions starting with an initial population of random chromosomes. The set of chromosomes in a given population are evaluated using a suitable problem-dependent fitness function. Good chromosomes have a high probability to be selected for the crossover and mutation operations to pass their good characteristics to the next generations to obtain an optimal or near-optimal solution after generating sufficient number of generations based on some stopping criteria [3].

Several approaches have been proposed in the literature to optimize the autonomous mobile robots path planning problem. Liu proposed enhanced mutation and selection operators have been proposed to solve the Unmanned Aerial Vehicle (UAV) swarms path planning problem [4]. Simulation results show that the proposed genetic operators allow the algorithm to efficiently achieve optimal solutions. Y. Pehlivanoglu and P. Pehlivanoglu [5] proposed an enhanced population initialization method to optimize autonomous UAVs path planning for target coverage problems at the aim of accelerated search convergence. R. Shivgan and Z. Dong proposed an energy-efficient GA for drones path planning [6]. The proposed approach relies on minimizing the number of turns in the paths with a reduction of energy consumption of up to 5 times compared to greedy search. W. Rahmaniar and A. Rakhmania [7] presented path planning improvements for a mobile robot using genetic algorithms in an environment .The GA operators are proposed to accelerate the evolution of individual populations in the path.

The performance and efficiency of genetic algorithms depend on the encoding approach used to map real-world solutions to chromosomes accessible to the algorithm, the genetic operators used during the search process, and the parameter settings of these operators [8]. In this work, the path planning problem for mobile robots is formulated as an optimization problem that can be solved using genetic algorithms. Several genetic operations are used and systematically tuned to find optimal paths. The proposed search methodology is evaluated for different scenarios using a quality function that measures both the path quality and the speed of the search process.

The remaining of this paper is organized as follows; Section II presents the formulation of the autonomous mobile robot path planning problem as an optimization problem. The evaluation criteria used to evaluate the proposed approach is provided in Section IV. The proposed genetic search methodology is introduced in Section III. Experimental results are presented in Section V. Finally, Section VI concludes the paper.

II. Problem Formulation

A. search space

In a two-dimensional plane, the mission area of dimensions $L \times W$ include N navigation points (x, y) and X static obstacles. The goal of a mobile robot is to reach a destination point (xn, yn) from an initial point (x0, y0) through a set of navigation points while avoiding obstacles in its path. There are (n-1)! possible solutions. The search will start with a population of P randomly selected individuals, where each individual represents a complete solution from the initial state to the goal state. Using a selection operator, the fittest solution will be selected to produce the offspring. Then, in a specific individual, a different type of cross-over will occur. The mutation operator will then apply for each offspring to improve the population diversity.

B. Model Constraints

A safe distance Rmin from the obstacle to the robot is considered. The distance between the robot and the obstacle should not be shorter than safe distance [9] which is expressed by:

$$D_{ij} - R_{min} \geq 0 \tag{1}$$

where Dij is the distance between the robot in i-th navigation point and the j-th obstacle.

C. Fitness Function

The path is made up of a series of navigation points (x, y) from i=0 to i=n, where (xi, yi) is the coordinate in two dimensions for the i-th navigation point and (xn, yn) is the goal point. Our goal is to minimize the path length while taking into account our constraint .The path length, that is determined by these locations, can be expressed using Euclidean or Manhattan distance measures as shown in Equations 2 and 3, respectively, with Manhattan distance used in this work.

$$H = \sum_{i=0}^{n} \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2} \tag{2}$$

$$H = \sum_{i=0}^{n} (x_{i+1} - x_i) + (y_{i+1} - y_i) \tag{3}$$

III. Methodology

A. Encoding Approach

Binary, real-number, and float encoding are the three main types of chromosomal encoding. Binary-based individual encoding is simple to use and takes less time to calculate. However, there is an issue when converting from a continuous variable to a binary number, which makes it challenging for the method to find an accurate solution for high-dimensional or continuous optimization problems. In this study, real number encoding, which has great calculation accuracy and can be easily optimized using genetic algorithms, is used to address the problem. where each navigation point will be represented with a number between 0 and n, where 0 represents the initial state and n represents the goal state. So each individual must start with zero and end with n. The most natural way to present the tour is probably by using path representation. For example, if n equals 8, a tour 0→1→4→7→2→5→3→6→8 can be represented simply as (0 1 4 7 2 5 3 6 8) .

B. Obstacles avoidance

To make sure that the algorithm works safely and the robots navigate without any probability of crushing any obstacles, a safe distance Rmin from the obstacle to the robot is considered. The algorithm checks at first the distance between all navigation points and the defined obstacles to ensure it is a safe distance. The algorithm then generates the population and calculate the distances for all solutions and all obstacles. If the distance is larger than the safe margin, the algorithm will work with the same initialization. Otherwise, the algorithm will break and increase population size by 20 and regenerates new chromosomes.

C. Adaptive Restart Condition

In this study, an enhanced GA called adaptive restarting GA [10] is adopted to improve the global search capability of the algorithm. With an adaptive restarting procedure, the proposed GA can jump out of the local optima and find the global optimum with a high success probability. This approach depends on the value addressed for the adaptive restart condition. The algorithm normally starts the exploration with an evaluation, crossover, and mutation operators. Each generation will save the solution with the minimum cost function until the number of generation numbers equals the value of the adaptive restart condition. The algorithm will calculate the absolute difference between the current optimal and previous solutions and then calculate the sum of the differences. If the sum is less than 0.00001, that means there is no enhancement of the solution, which means the algorithm is stuck in the local optimum so that the algorithm will break and restart. Otherwise, the algorithm starts the new generation with the best solutions. The check of enhancement will repeat each generation until the algorithm reach the desired number of generations.

IV. Evaluation Approach

In order to test the effectiveness of the algorithm and find the best value for tuning parameters such as population size, the number of generations, the Number of crossover chromosomes, and the Number of the mutated chromosome, different scenarios of robot path planning will be carried out by tuning the algorithm parameters while taking into consideration the time to reach the optimal solution . Each scenario's quality function will be calculated and compared to the rest. The case with the maximum quality will be considered an optimal solution. The quality function is a function of total path length, population size,the number of pairs of chromosomes required for crossover, the Number

Table I: Impact of population size

Population size	Time (second)	path length	Quality function
10	4.35	123.8	4.78
30	4.28	121.6	5.14
50	3.76	121.1	6.32
70	3.24	120.5	7.80
90	2.65	119.9	10.11
100	2.58	119.9	10.67
150	2.97	120.3	10.54

of chromosomes required for mutation, and the Number of generations. Which is expressed by:

$$Q = [0.6\,[1/L] + 0.1P + 0.1G + 0.1C + 0.1M]/T \qquad (4)$$

Where:
L: The total path length
T: Time to find the solution
P: Population size
G: The number of generation
C: Number of pairs of chromosomes required for crossover
M: Number of chromosomes required for mutation

V. Results and Discussion

Experimental results reported in the following subsections are obtained using MATLAB GA solver that has the advantage of performing mathematical computations in a simple way while focusing on tuning the algorithm parameters to enhance the quality of the obtained solutions.

A. Impact of Population Size

In this subsection, we investigate the impact of the population size on the performance of the proposed GA-based path planning approach for N=10, number of generations=100, number of chromosomes required for crossover=0.5P, and the Number of chromosomes required for mutation=0.5P. The population size is varied from 10 to 150. As shown in Table I, increasing the population size decreases the time required to reach the optimal solution at first, and then for a large value of P, the time starts increasing. At the same time, decreasing the total path length, which is what we were looking for, then for a large value of P, the path length increases again. Hence, a trade-off between the time needed to reach the optimal solution, the quality function value and the total path length is needed. The case with P equals 100 is the best with search time of 2.58 s, quality value of 10.67, and total path length of 119.9. Figure 1 shows the total travel distance (path length) of obtained solutions with generations for P=100.

B. Impact of Crossover Operator

The selection operator relies on the quality of the chromosomes in the population to determine which parents will be used for crossover to mate and generate new offspring. The crossover then brings up a new offspring based on the exchange point chosen with particular parts of the parents.it is more likely that the new offspring would contain good parts of their parents, and consequently perform better as

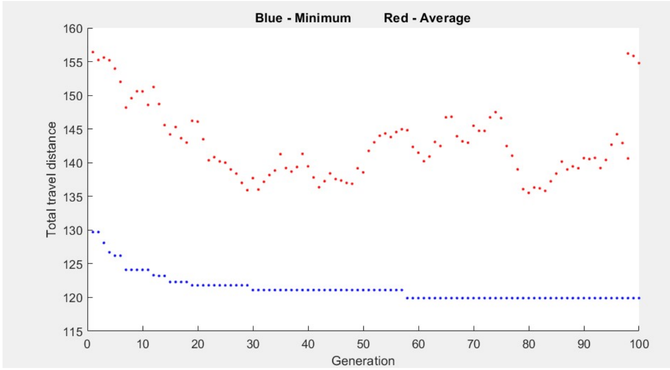

Figure 1: p=100, c=25, M=50 ,G=100

Table II: Impact of crossover rate

Crossover rate	Time (second)	path length	Quality function
10	4.63	123.8	5.6
30	3.95	121.8	7.36
50	3.68	121.1	8.64
70	2.15	119.9	14.6
90	2.46	119.9	13.82
100	2.65	119.9	12.83

compared to their parents [11]. Table II shows the impact of crossover on the quality of the solution for N=10, number of generations=150, population size=100,and the Number of chromosomes required for mutation equals 0.5C. As shown in this table,increasing crossover rate (or the number of chromosomes required for the crossover) decreases the time needed to find the best solution and the total path length while increasing the value of the quality function. Large crossover rate results in more exploration during the search process with a constant rate of mutation (exploitation). The solution's quality decreases because the time to reach the optimal solution will increase. Figure 3 illustrates how the quality of the solution improved in the case of a crossover rate equal to 0.7.

C. Impact of Mutation Operator

Mutation is a small random tweak in the chromosome to get a new solution. It is used to maintain and introduce diversity in the genetic population and is usually applied with a low probability. The GA gets reduced to a random search if the probability is very high. The mutation is the part of the GA related to the exploitation of the search space. This section examines different cases with different mutation rates to test the influence of exploitation rate on time to reach the optimal solution, the total path length, and the value of the quality function. From the previous section, the best value for P is 100, for G is 150, and For C is 35.

As shown in Table III, for a mutation rate from 10 to 30, increasing the number of chromosomes required for the mutation will decrease the time to reach the best solution and the total path length while increasing the value of the quality function. However, when the mutation rate increases, or the exploitation rate increases, the total path length increase, and the time to reach the optimal solution will decrease.

Table III: Impact of mutation rate

Mutation rate	Time (second)	path length	Quality function
10	2.51	120.3	11.75
20	2.25	119.9	13.55
30	2.04	119.9	15.44
40	2.37	120.3	13.71
50	2.98	120.5	11.24

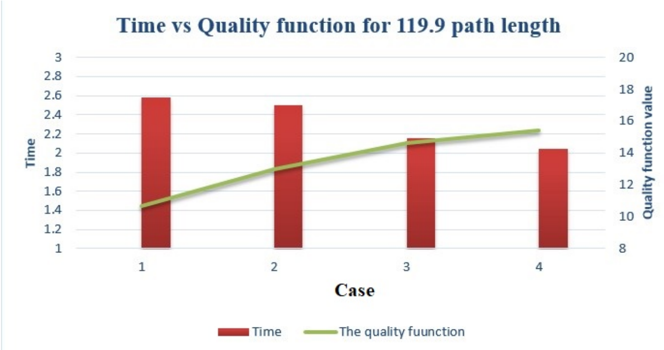

Figure 2: Time vs quality function for case 1, case 2, case 3, and case 4

D. Overall Performance and discussion

Population size, number of generations, crossover rate, and mutation rate are very important parameters that directly influence the quality of the genetic algorithm solution. To study the overall impact of these parameters, different cases were tested by changing one parameter while others were constant. Starting with random initialization for G, C, and M. we changed P from 10 to 150. The best value of population size was determined, which is the population size gets the best solution with minimum time and high-quality function value. To test the influence of the number of generations, we started with random initialization for only C and G while using the best P value found in the first section. Then for the number of chromosomes required for crossover, the best value found for population size and the number of generations used, and so on.

In Figure 2, the first case represents the case with the best value for P and the random value for the rest three parameters, which explains the low value for the quality function and a large value for the time. In the second case, only two parameters have a random value, while the rest equals its best value. As shown in the figure for the four cases, the quality function value increased from 10.67 to 15.44, and the time decreased from 2.58 to 2.04.

Based on the obtained results in this section, it is clear that large-enough population size leads to better convergence of the algorithm to an optimal solution. Nevertheless, having a large population size will not be a good idea when the search space is small. The crossover and mutation rates are hyper-parameters that control the rate at which solutions are subjected to crossover and mutation operations. The higher the crossover rate, the more crossovers perform, so the more diversity (in terms of solutions/chromosomes) is introduced.Mutation, on the other hand, can prevent premature convergence but if the mutation rate is very high, the algorithm becomes a random search. Hence, the value of the mutation rate is typically not high.

VI. Conclusion

This work investigates the impact of the main parameters of the genetic algorithm, including population size, mutation rate, crossover rate, the number of generations, and the population size on the performance of efficiency of the path planning problem for autonomous mobile robots. Experimental results show that systematic tuning of these parameters result in faster convergence of the algorithm to better solutions. As a future work, we intend to to increase the algorithm's strength in the path planning problem to handle a swarm of robots taking into account different robot constraints.

References

[1] H. Liu, X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan, "An autonomous path planning method for unmanned aerial vehicle based on a tangent intersection and target guidance strategy," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3061–3073, 2020.

[2] K. Sastry, D. E. Goldberg, and G. Kendall, "Genetic algorithms," *Search methodologies: Introductory tutorials in optimization and decision support techniques*, pp. 93–117, 2014.

[3] A. Alzahrani and A. Kanan, "Machine learning approaches for developing land cover mapping," *Applied Bionics and Biomechanics*, vol. 2022, 2022.

[4] J. Liu, "An improved genetic algorithm for rapid uav path planning," in *Journal of Physics: Conference Series*, vol. 2216, no. 1. IOP Publishing, 2022, p. 012035.

[5] Y. V. Pehlivanoglu and P. Pehlivanoglu, "An enhanced genetic algorithm for path planning of autonomous uav in target coverage problems," *Applied Soft Computing*, vol. 112, p. 107796, 2021.

[6] R. Shivgan and Z. Dong, "Energy-efficient drone coverage path planning using genetic algorithm," in *2020 IEEE 21st International Conference on High Performance Switching and Routing (HPSR)*. IEEE, 2020, pp. 1–6.

[7] W. Rahmaniar and A. E. Rakhmania, "Mobile robot path planning in a trajectory with multiple obstacles using genetic algorithms," *Journal of Robotics and Control (JRC)*, vol. 3, no. 1, pp. 1–7, 2022.

[8] A. Vlasov, A. Khomchenko, A. Faizliev, S. Mironov, and A. Grigoriev, "Parameter tuning of a genetic algorithm for finding central vertices in graphs," in *Journal of Physics: Conference Series*, vol. 1784, no. 1. IOP Publishing, 2021, p. 012009.

[9] C. Qi, C. Wu, L. Lei, X. Li, and P. Cong, "Uav path planning based on the improved ppo algorithm," in *2022 Asia Conference on Advanced Robotics, Automation, and Control Engineering (ARACE)*. IEEE, 2022, pp. 193–199.

[10] S. D. Dao, K. Abhary, and R. Marian, "An adaptive restarting genetic algorithm for global optimization," in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1, 2015.

[11] A. Hassanat, K. Almohammadi, E. Alkafaween, E. Abunawas, A. Hammouri, and V. S. Prasath, "Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach," *Information*, vol. 10, no. 12, p. 390, 2019.