

# Adaptable Genetic Algorithm Path Planning of Mobile Robots Based on Gene Reallocating

Faten Abushmmala<sup>1,\*</sup>, Iyad Abuhadrous<sup>2</sup>

<sup>1</sup>Computer Engineering, Islamic University, Gaza, Palestine

<sup>2</sup>Computer Engineering, Palestine Technical College, Gaza, Palestine

## Email address:

Faten.ffa@gmail.com (Faten Abushmmala), iahadrous@ptcdb.edu.ps (Iyad Abuhadrous)

\*Corresponding author

## To cite this article:

Faten Abushmmala, Iyad Abuhadrous. Adaptable Genetic Algorithm Path Planning of Mobile Robots Based on Gene Reallocating. *Advances*. Vol. 3, No. 3, 2022, pp. 125-131. doi: 10.11648/j.advances.20220303.22

**Received:** August 31, 2022; **Accepted:** September 19, 2022; **Published:** September 29, 2022

---

**Abstract:** Mobile path planning is rich field of employing artificial intelligence and machine learning algorithms to obtain the most effective outcomes. The Path planning task is a problem. The goal of path design is to find the quickest and most obstacle-free route from a starting point to a destination state. A set of states (position and orientation) or waypoints can make up the path. A map of the surroundings, as well as the start and target states, are needed for path planning. Path planning applications are diverse and unlimited, such as Automated robot navigation, autonomous vehicle Robotic surgery, digital animation of characters, and others. Different algorithms provide different solutions to this problem; usually the metric used to evaluate certain path effectiveness doesn't take into consideration the physical attributes of the mobile robot. In this paper, an attempt is made 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.

**Keywords:** Path Planning, Genetic Algorithm, GA, 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 [1-3], cruise missile path planning [4] and others. The optimal path is not just the shortest path; it's also defined as a collision-free trajectory; each path comes with specific constraints depending on the area of application used in. What is considered optimal for one mobile robot or application may not be considered optimal for another. because of several factors such as the physical characteristics of the robot, the application environment, and the application constraints. These factors can be interrupted as the starting and finishing time, the robot's ability and performance, and the number of nodes and obstacles ahead. The Genetic Algorithm is one of

the search algorithms that depend on iterative attempts to find the optimal solution. The Genetic Algorithm (GA), Ant Colony Algorithm, and Annealing algorithm are natural-based algorithms used extensively in path planning. Qinggang Su et al. [5] propose a modification to the ant colony algorithm in finding the optimal path and path correction. To get fast convergence the obtained path is shorter and avoids redundant paths, which is a big disadvantage in the ant colony algorithm. It also provides path smoothness. Chunyu Ju et al. [6] discuss the work of the A-star algorithm in finding the best path under certain conditions. They improve and modify the A-star algorithm by using the shortest line segment between two points' notions to provide a much shorter path for certain environments.

Genetic Algorithms (GA), in particular, provide global optimal solutions and handle constraints very well. GA can provide an optimal solution in a topology that keeps updating. GA implies providing several solutions. GA is used in

combination with artificial algorithms in image processing applications and data mining. For its appealing characteristics, GA attracts researchers more and more to explore its potential. For instance, Chaymaa Laminia et al. [7] suggest a modification to traditional GA by improving the crossover operator in a static environment. The modification gives much better fitness values. Their work was compared with three other literatures and showed better results in comparison. Lee et al. [8] work on the initial population that is provided to GA before starting, which is usually set up as random. The proposed work suggests improved GA performance in finding the best path for mobile robots. On the other hand, Tuncer A. et al. [9] proposed modifications to the GA operations, more specifically on the mutation operator, to enhance GA work and the ability to find optimal paths for path planning problems in dynamic environments. Their work showed good results in a comparison of traditional GA solutions. The operator modifications make GA coverage much faster, which means finding the best path in a shorter execution time. The work constraints may depend on the robot or the environment, but fixed consideration must be included, such as the safety and the power consumption of the robot's movement. Different algorithms than genetic algorithm used for path planning such as A\* algorithm, the A\* algorithm was originally developed for use in the gaming industry [10-12], but as artificial intelligence has advanced, it has since been enhanced and customized for use in applications such as robot path planning, intelligent urban transportation, graph theory, and automatic control.

This paper takes these constraints under consideration, finding an optimal path using a modified genetic algorithm operator based on the gene reallocating along the path, in order to minimize the rotation angle and provide path smoothing, which ultimately leads to reduced power consumption; of course, a shorter path is still achieved. The results proved the proposed system's feasibility in path planning. The rest of the paper is organized as follows: Section 2 presents genetic algorithms for path planning; the paper's methodology is introduced in Section 3; simulation and results are presented in Section 4; and finally, conclusions in Section 5.

## 2. Genetic Algorithm for Path Planning

Genetic algorithm [13] 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.

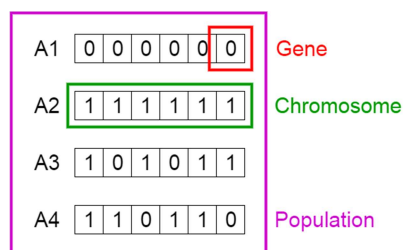


Figure 1. Genetic algorithm.

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.

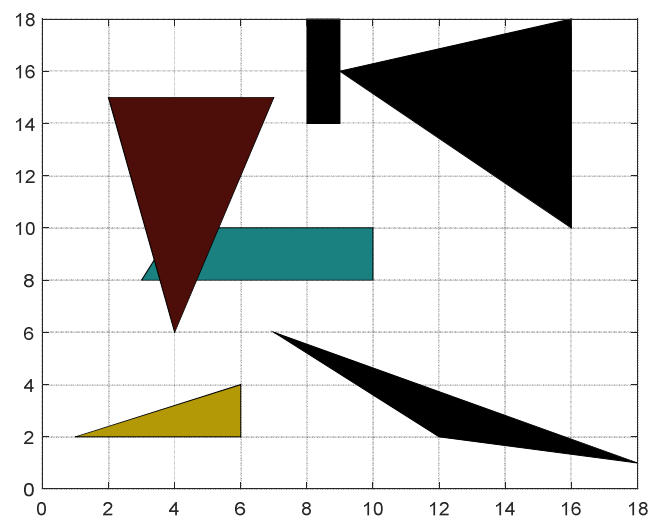


Figure 2. 2-D planer graphics with arbitrary irregular polygons.

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 ( $n_1, n_2, n_4, \dots, n_n$ ), each node is a point of (x, y) coordinates shown in Figure 3.

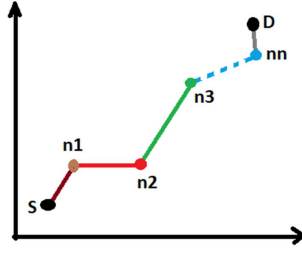


Figure 3. Chromosome genes (nodes).

### 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). Assuming the start point to be  $S(x_s, y_s)$  and the finish point D to be  $(x_d, y_d)$ , the generated population points  $(x_i, y_i)$  must be less than  $(x_d, y_d)$ , and bigger than  $(x_s, y_s)$ .

Where  $x_{i+1} \in [x_s, x_d]$ ,  $y_{i+1} \in [y_s, y_d]$

$$\begin{cases} x_{i+1} = (x_d - x_i) * rand + x_i, \\ y_{i+1} = (y_d - y_s) * rand, \end{cases} \quad (1)$$

Where  $x_{i+1} \in [x_s, x_d]$ ,  $y_{i+1} \in [y_i, y_d]$

$$\begin{cases} x_{i+1} = (x_d - x_s) * rand \\ y_{i+1} = (y_d - y_{i+1}) * rand + y_i, \end{cases} \quad (2)$$

Where rand state for function in MATLAB which gives every time different random variables range between 0 to 1.

### 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.

$$FitnessFunction = 0$$

$$FitnessFunction = \frac{C}{\sum_{i=1}^{n-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}} \quad (3)$$

C is constant number; it can be any Constant number.

### 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).

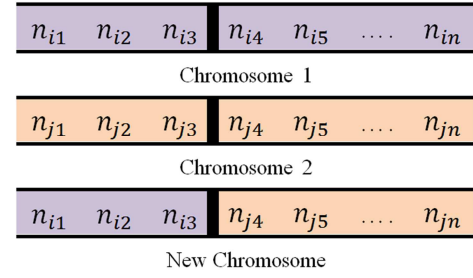


Figure 4. Chromosome single crossover operators.

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.

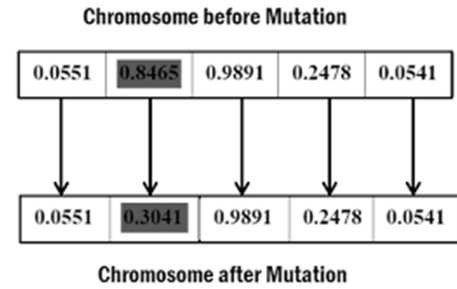


Figure 5. Chromosome mutates operators.

## 3. Methodology

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, vertex identification operator and gene reallocating operator the whole process shown in Figure 6.

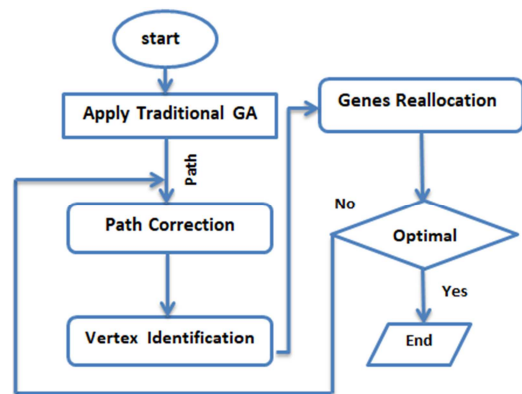


Figure 6. Flow chart showing the methodology.

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.

### 3.1. Path Correction Operator

Each solution is group of points that joins the line segments that represents the path, in attempt to enhance the selected path these points changed, and must be re mapped into the enhanced line. In this operator must identify the mapping process see Figure 6.

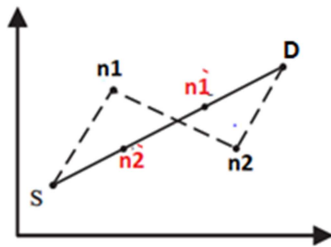


Figure 7. Path correction.

Let's assume that a path consisting of S, D, was found during the work, an alternative path which could give a direct line between S and D without hitting any obstacles, which means that there is no need for, for that, must be projected into the new path through the following equations:

$$\begin{cases} x_j = \frac{j*(x_k - x_i)}{k-i} + x_i \\ y_j = \frac{j*(y_k - y_i)}{k-i} + y_i \end{cases} \quad (4)$$

The projection of,  $n_1, n_2$  will be  $n_1'$  and  $n_2'$ , the main purposes of path correction Figure 7 to get a shorter path (straight lines always shorter), secondly to smoothen the path.

### 3.2. 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). To solve this issue, first we need to identify the nodes that are on the top of large rotation angles and call them vertex. To obtain the value of each rotation angle, we use the following equation. If it's less than 180, then it needs further analysis and is called a vertex. The angle between the two lines of the  $i+1$ -th node is:

$$\theta_{i+1} = \arccos \left\{ \frac{(x_i - x_{i+1})(x_{i+2} - x_{i+1}) + (y_i - y_{i+1})(y_{i+2} - y_{i+1})}{\sqrt{((x_i - x_{i+1})^2 + (y_i - y_{i+1})^2)((x_{i+2} - x_{i+1})^2 + (y_{i+2} - y_{i+1})^2)}} \right\} \quad (5)$$

### 3.3. 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 ( $k_1, k_2$ , 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 happed.

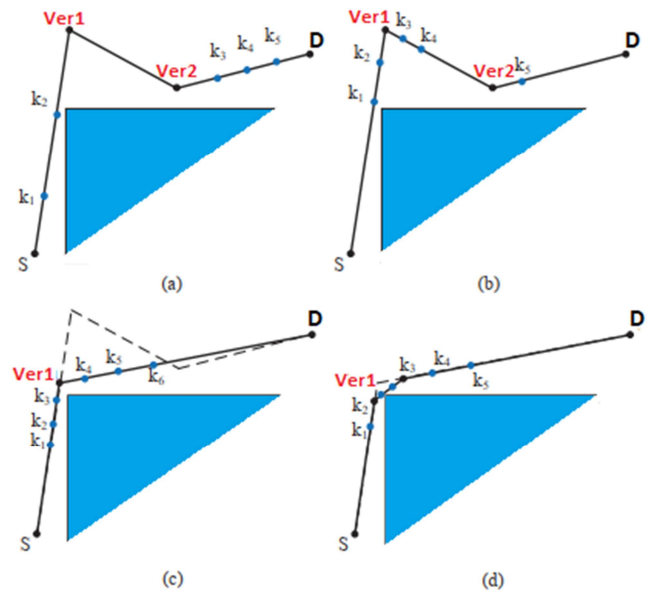


Figure 8. Gene Reallocation.

The probability of getting more genes is related to the angle of vertex value. The larger the angle, the more the angle attracts variable genes. The following equations express this idea:

$$p_i = \frac{\theta_i}{\sum_{i=1}^k \theta_i} \quad (6)$$

Each vertex has two sides, the longer the side the more variable genes; the following shoes the probability of doing so:

$$\begin{cases} P_{left} = \frac{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}}{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}+\sqrt{(x_{i+2}-x_{i+1})^2+(y_{i+2}-y_{i+1})^2}} \\ P_{right} = \frac{\sqrt{(x_{i+2}-x_{i+1})^2+(y_{i+2}-y_{i+1})^2}}{\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2}+\sqrt{(x_{i+2}-x_{i+1})^2+(y_{i+2}-y_{i+1})^2}} \end{cases} \quad (7)$$

As you can see the length is critical in giving higher probability.

To map the new location of the variable genes the following equations indicate the new coordinate in relation to the old coordinate for the left side:

$$x_k = \frac{k*(x_{i+1}-x_i)}{2(n+1)} + \frac{x_{i+1}+x_i}{2} \quad (8)$$

$$y_k = \frac{k*(y_{i+1}-y_i)}{2(n+1)} + \frac{y_{i+1}+y_i}{2} \quad (9)$$

And for the right side:

$$x_k = \frac{k*(x_{i+2}-x_{i+1})}{2(m+1)} x_{i+1} \quad (10)$$

$$y_k = \frac{k*(y_{i+2}-y_{i+1})}{2(m+1)} y_{i+1} \quad (11)$$

## 4. 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.

### 4.1. Irregular Environment

Figure 9 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.

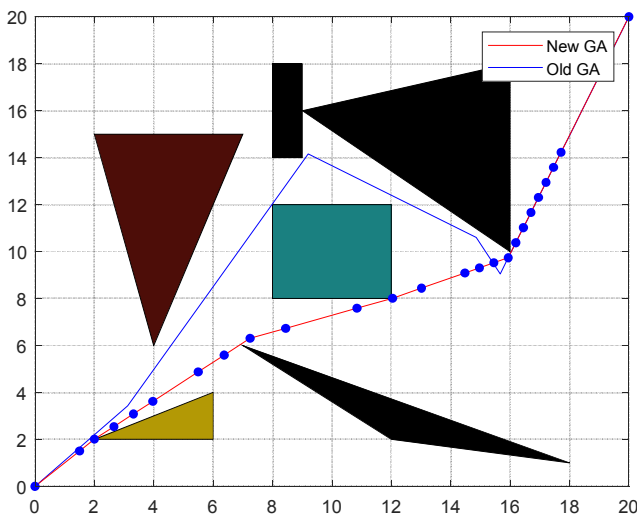


Figure 9. Results on irregular environment.

### 4.2. Narrow Winding Environment

Figure 10 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.

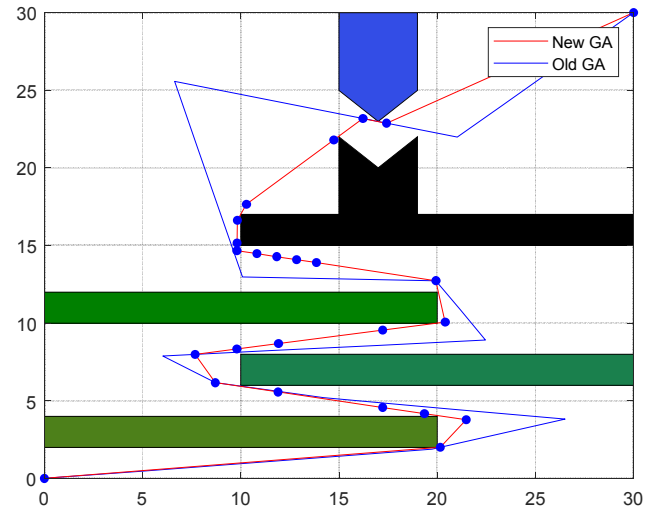


Figure 10. Results on narrow winding environment.

### 4.3. Complex Maze Environment

Finally; compel maze environments (CME) model which shown in Figure 11. 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.

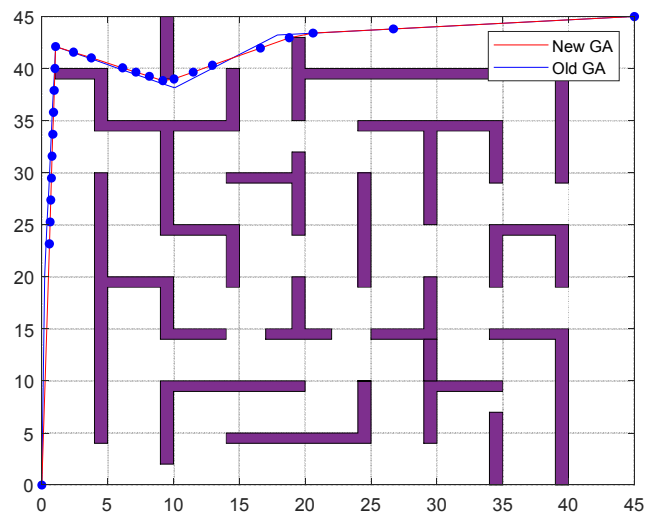


Figure 11. Results on compel maze environments.

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,  $c=1.5$ .

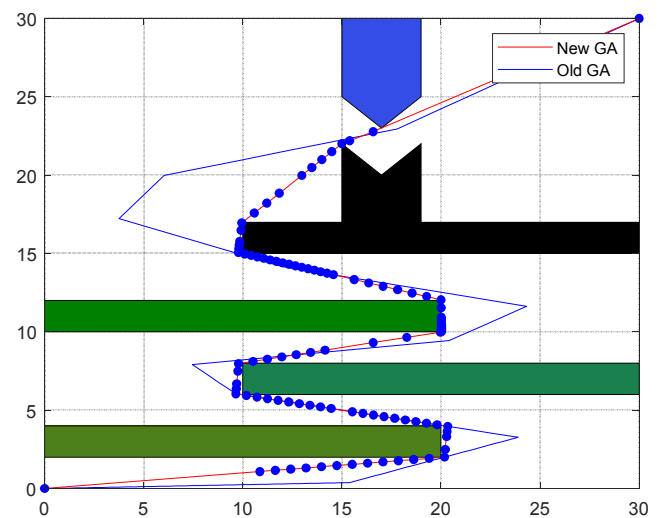
Env.	B-PathLeng	B-AngleSum	A-PathLeng	A-AngleSum
IE	33.1975	148.2627	29.9169	70.5937
NWE	132.1320	734.3380	84.7895	651.0391
CME	129.0956	745.7954	77.7448	210.0316

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 12, 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. It is clear from the previous comments and from the numerical results in the tables that the enhanced GA with the suggested added operators 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 12.** The NWE with 100 genes.

## 5. Conclusion

Using Genetic Algorithm to find optimum path investigated several times in the past [16-18], but in this paper, the enhancement came after executing the traditional genetic algorithm as a new added phase. In this paper, an enhanced genetic algorithm is executed by adding three new operators after obtaining the solutions from the classic traditional algorithm. The main goal is to minimize the power energy by minimizing the rotation angles. This goal will ultimately achieve another goal, which is to shorten the length of the path. The first path correction operator solves the problem of unnecessary rotations when there are no obstacles, but the second problem of hard angles around obstacles requires further analysis and optimization, which is carried out using two operators: vertex identification and gene reallocation. Once the vertex of the angles is defined, the gene is re-distributed to help the path correction operator work much more easily and provide a smoother and shorter path. The experimental results

show tremendous performance, showing that the three operators add high values to the traditional GA. We also found out that the number of genes must depend on the complexity of the environment and is related to the number of vertex.

## 6. Recommendation

The main recommendation that would be to investigate more complex environments and setting up customized number of genes based on the environments (per application).

## References

- [1] Rudnick-Cohen, E. S., Herrmann, J., & Azarm, S. (2015). Risk-based path planning optimization methods for uavs over inhabited areas. *Journal of Computing & Information Science in Engineering*.
- [2] Kamyar, R., & Taheri, E. (2014). Aircraft optimal terrain/threat-based trajectory planning and control. *Journal of Guidance Control & Dynamics*, 37 (2), 466-483.
- [3] Liang, H. F., Fan, G. C., & Dong, Y. F. (2005). Generation of target aircraft's flight track in air combat plan simulation. *Computer Simulation*, 22 (12), 32-34.
- [4] Xian, Y., He, B., Liu, G., & Lei, G. (2012). Cruise missile route planning based on quantum immune clone algorithm. *Journal of Information & Computational Science*, 9 (8), 2097-2105.
- [5] 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).
- [6] 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.
- [7] Chaymaa Laminia, Said Benhlila 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.
- [8] 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.
- [9] Tuncer, A., & Yildirim, M. (2012). Dynamic path planning of mobile robots with improved genetic algorithm. *Computers & Electrical Engineering*, 38 (6), 1564-1572.
- [10] Belhaous, S.; Baroud, S.; Chokri, S.; Hidila, Z.; Naji, A.; Mestari, M. Parallel implementation of A\* search algorithm for road network. In *Proceedings of the 2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, Marrakech, Morocco, 28–30 October 2019; pp. 1–7.
- [11] Dong, G.; Yang, F.; Tsui, K. L.; Zou, C. Active Balancing of Lithium-Ion Batteries Using Graph Theory and A-Star Search Algorithm. *IEEE Trans. Ind. Inform.* 2021, 17, 2587–2599.
- [12] Liu, H.; Shan, T.; Wang, W. Automatic Routing Study of Spacecraft Cable based on A-star Algorithm. In *Proceedings of the 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, Chongqing, China, 12–14 June 2020; pp. 716–719.
- [13] Lance D. Chambers, (2019) "The Practical Handbook of Genetic Algorithms: New Frontiers, Volume II, Volume 2", Publisher, CRC Press, 2019, ISBN 1420050079, 9781420050073, Length- 464 pages.
- [14] Zhang, Q., & Ding, L. (2016). A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems. *Expert Systems with Applications*, 60 (C), 183-189.
- [15] Nandini, D., & Seeja, K. R. (2017). A novel path planning algorithm for visually impaired people. *Journal of King Saud University - Computer and Information Sciences*.
- [16] Dao, S. D., Abhary, K., & Marian, R. (2017). "A bibliometric analysis of genetic algorithms throughout the history". *Computers & Industrial Engineering*, 110.
- [17] Ghiduk, A. S. (2014). "Automatic generation of basis test paths using variable length genetic algorithm". *Information Processing Letters*, 114 (6), 304-316.
- [18] Zhang, Q., & Ding, L. (2016). "A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems". *Expert Systems with Applications*, 60 (C), 183-189.