DOR: 20.1001.1.27170314.2022.11.3.5.5

Research Paper

# Path Optimization of Moving Object in Presence of Obstacles Using Messy Genetic Algorithm for N-dimensional Space

**Ali Hasanabadi**[1*]
[1]Mechanical Engineering Department, University of Birjand, Birjand, Iran
[*]Email of the Corresponding Author: hasanabadi@birjand.ac.ir
*Received: August 10, 2022; Accepted: October 28, 2022*

**Abstract**
Optimizing the path of the movement of moving objects such as various robots in the industry can have a significant effect on reducing manufacturing and production time and costs. In this research, using a messy genetic algorithm, a new method is presented to optimize the movement path of a mobile object such as a robot in the presence of multiple obstacles. The movement path can be considered two-dimensional or multi-dimensional, and the obstacles in the path are assumed to be circles and spheres. The method used is that first, several chromosomes are created in the zero generation and their fitness is calculated. Then, using competitive selection, the parents of the new generation are created and from there the chromosomes of the next generation are made, and their fitness is calculated. This process continues until the considered condition, i.e. the ratio of the average fitness divided by maximum fitness in each generation is satisfied. Since in the messy genetic algorithm, the length of the chromosome can be variable, the proposed algorithm can examine all types of paths with a variable number of points depending on the existing obstacles and with high efficiency, find the shortest path with an approximate difference of 3.4 percentage compared to the ideal path. This method can optimize even paths with more than three dimensions.

**Keywords**
Optimization, Moving Object Path Design, Messy Genetic Algorithm

## 1. Introduction

Optimizing the path of the movement of moving objects such as various robots in the industry can have a significant effect on reducing manufacturing and production time and costs [1, 2]. In addition, providing a reliable algorithm can prevent the risks of the robot colliding with obstacles on its way and related complications [3].

Designing the moving path of a moving object has a very wide application. As an example, we can mention the design of the path of the robot arm for dangerous or repetitive tasks [4, 5], transportation of objects on the factory floor, types of flying robots in 3D space, and similar things. In all of these, the design of the path should be done considering the two factors of non-collision and optimality of the path.

Finding the shortest admissible path is an optimization problem. Designing the path of a moving object and optimizing it has been the interest of researchers for a long time. Most of the methods used by researchers for this purpose include the use of genetic algorithm methods [6], particle swarm [7-9], artificial potential field [10], and ant colony [11, 12].

Nasrollahy and Haj Seyyed Javadi [8] using the particle swarm optimization (PSO) method presented an approach for path planning of a mobile robot. They considered things like not colliding obstacles, shortening the path, and moving targets in the presented method.

Ge and Cui [10] investigated the limitations of path design using the potential field method and then presented a new function for path design. The potential field method is based on the fact that the entire space in which the robot can move is defined by a potential function in such a way that the proximity to the target reduces this energy and the proximity to obstacles increases it [13].

Another method that has been widely used by researchers for path planning is the genetic algorithm (GA) [14-17]. In the genetic algorithm, first, a population of chromosomes is created as the zero generation. Then, the fitness value of the chromosomes is calculated, and based on that, the chance of choosing them as the parents of the chromosomes of the next generation is determined. The chromosomes of the next generation are created by cutting the selected chromosomes of the previous generation and splicing them together. This process continues until a convergence condition is satisfied.

In this research, a new and powerful method for achieving the optimal path of robot movement using a messy genetic algorithm is presented. The advantage of this method compared to the conventional genetic algorithm is the change in the length of its chromosomes, which can allow various paths to be investigated. Due to the inherent capability of the algorithm, which is the diversity in the length of the chromosome, it is possible to evaluate and provide diverse paths with high quality and low computational cost not only in two-dimensional space but also in three-dimensional space and more.

## 2. Path Design

In this method, two points are determined as the starting and ending points of the path and based on them, the potential movement area of the robot is determined as a rectangle in two-dimensional mode or a cube in three-dimensional mode, as illustrated in Figure 1. The obstacles in the movement space of the robot are identified as circles (or spheres in three-dimensional space), in such a way that all obstacles are surrounded by the circles. Then, random paths with a different number of lines are considered. Paths that cross obstacles are considered inadmissible paths and discarded (the red one in Figure 1). The admissible paths (the blue ones in Figure 1) are entered into the optimization process to find the shortest path.

The design of the route is defined point by point in such a way that the number of points is variable and the connecting lines between the points determine the path. The innovation of the method is that in the proposed algorithm, chromosomes with variable lengths are used in the implemented messy genetic algorithm, therefore as will be shown in the results section, the examined paths can have high flexibility and finally provide a high-quality path.

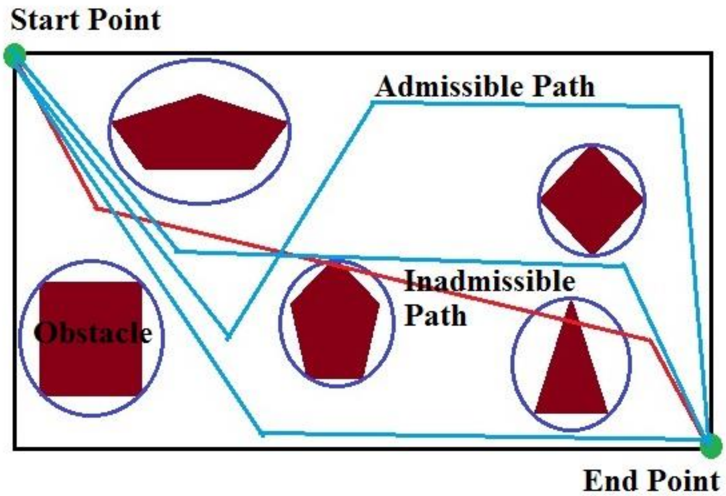


Figure 1. Schematic illustration of the path design in presence of the obstacles; every path that crosses the obstacles is inadmissible

## 3. Messy Genetic Algorithm

The method used in this research is based on the messy genetic algorithm [18]. For this purpose, all steps of the algorithm have been coded and executed in the software. During the execution of the code, the required inputs, as depicted in Figure 2, must be given to the software as follows: initial point, final point, space dimension, the number of chromosomes, the number of constraints, the constraints themselves, the number of points for the connected path should be checked from the point of view of violating the constraints, the condition of convergence (the ratio of the average value of the ranks to the maximum rank) and the maximum length of the chromosomes for the zero generation.

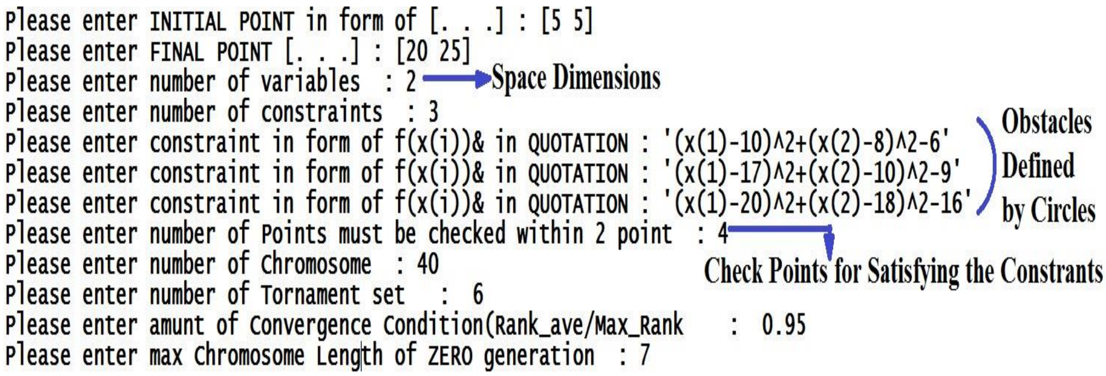


Figure 2. Assigned Initial inputs according to where the code of the algorithm is executed

Then, for each generation (including generation zero or otherwise), chromosomes must be made in the specified number as shown in Figure 3. The length of each chromosome includes an identification number, the starting point of the movement, an arbitrary number of intermediate points, and finally the endpoint of the movement. After each chromosome is made, the admissibility of its points, as well as the admissibility of other points (which, depending on the user's choice, are several times the number of the main points of the chromosome) related to the path connecting the points, are checked according to the obstacles. The meaning of admissibility is that no point of the considered path should be placed inside the specified obstacles.

| Chromosome identification number | First Point (specified by user) | Intermediates point (in random numbers) | Last point (specified by user) |
|---|---|---|---|

Figure 3. Chromosome length arrangement

In the next step, the fitness of each chromosome should be calculated. For the chromosomes of each generation, the fitness and thus the rank is calculated based on the inverse of the path distance.
Another important point that is included in the relevant code is searching for the elite of each generation and maintaining it and then updating it in the next generation, which has a great impact on maintaining the achievements and ultimately the quality of the answer.
After checking the convergence condition, if it is not satisfied, the next generation should be built. The convergence condition is the ratio of the average fitness to the best fitness in each generation. If it is satisfied, the algorithm will stop.
To create the chromosomes of the next generations, the parents must be selected first, and to avoid the roulette wheel and the mating pool, a tournament selection is used in which the parents are directly selected. In the tournament selection, for each chromosome of the chromosome pair, a chromosome set is taken randomly and the best of them are selected as father or mother based on their rank. Here, it should be noted that a chromosome set must contain distinct chromosomes (in terms of chromosome identification number), that is, when selecting a chromosome set, it is checked whether the newly selected member is different from the others in terms of number or not. If it is not distinct, this process should be continued until another distinct chromosome is selected.
In the next step, to create the chromosomes of the next generation, the crossover must be done, which is done based on cutting and splicing, and there is no mutation in the implemented messy genetic algorithm.
It should be mentioned that in the used algorithm, the number of chromosomes is always fixed, but the length of the chromosomes, which are the points of the path as depicted in Figure 1, is flexible and can be increased or decreased.
Another point is that to balance the length of the chromosomes, if the length of the chromosome chosen as the father or mother is less than half of the maximum length of the total chromosomes, the chromosome will not be broken. Otherwise, it will break randomly at one point in its length.
After each chromosome is constructed, the relevant constraints are evaluated directly. If any of the child's chromosomes is defective, the whole process is done from the beginning. It means the tournament selection and election of the father and mother's chromosomes and other steps must be done again. It should be noted that the major part of the corresponding code calculations will belong to this step.

## 4. Results and Discussion

In this section, to show the capability of the proposed method, the results of several examples with different space dimensions and several obstacles are reported. It should be noted that in all cases, the number of chromosomes in each generation is equal to 60, the number of population members in tournament selection is equal to 6, and the condition of convergence is equal to 0.95.

*4.1 First case: without obstacle and two-dimensional space with start point (0, 0) and endpoint (20, 20)*

With the above given inputs, after 206 iterations (generations), chromosome number 12 from generation 3 is presented as the best chromosome. The path provided by the software includes an intermediate point with coordinates (9.9296 9.3947) between the start and endpoints. Based on this, the distance obtained for the path is also 28.2944, which is in very good agreement with the theoretical and exact value which is 28.2843.

*4.2 The second case: two-dimensional space with a circular obstacle exactly in the middle of the area and with a radius of 5*

In this case, in the condition that the minimum possible length according to Figure 4a is equal to 30.0713, according to Figure 4b, the length of the path provided by the proposed algorithm is 30.6792 whereas the best possible polyline path length is 30.2372 according to Figure 4a. The proposed path has one intermediate point with the coordinate (6.4601, 14.8475) and two line segments.

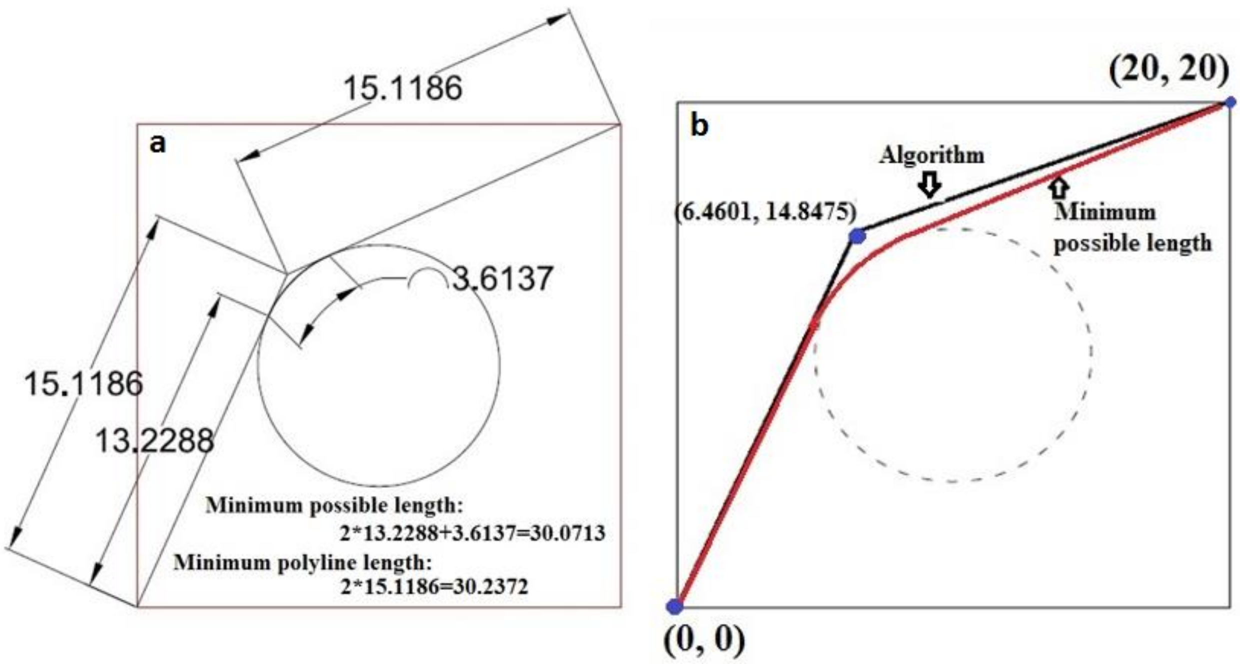


Figure 4. a) Minimum possible length and minimum polyline possible length, b) The proposed path by the algorithm

Figure 5 shows the increasing trend of the average fitness curve of generations. It should be noted that in this case, the best chromosome presented as the output of the algorithm was created in generation zero, however, to satisfy the condition of convergence and find a better possible path, the process has continued until the 125th generation. Considering that the algorithm preserves the elite of each generation and replaces it if a better elite is achieved, the best chromosome of all generations is presented as the final path.

*4.3 The third case: two circular obstacles in the centers (10,5) and (10,15) and with a radius of 4.5*

In this case, the shortest possible path with a length of 28.6701 is obtained according to Figure 6a by drawing two tangents and then a common tangent. After 14 generations, the best path with two points and a distance of 29.9465 is obtained.

To compare the path provided by the algorithm, two paths are compared in Figure 6b. In Figure 6, it can be seen that the algorithm can identify the right path well from a small distance between two obstacles and provide the desired path.

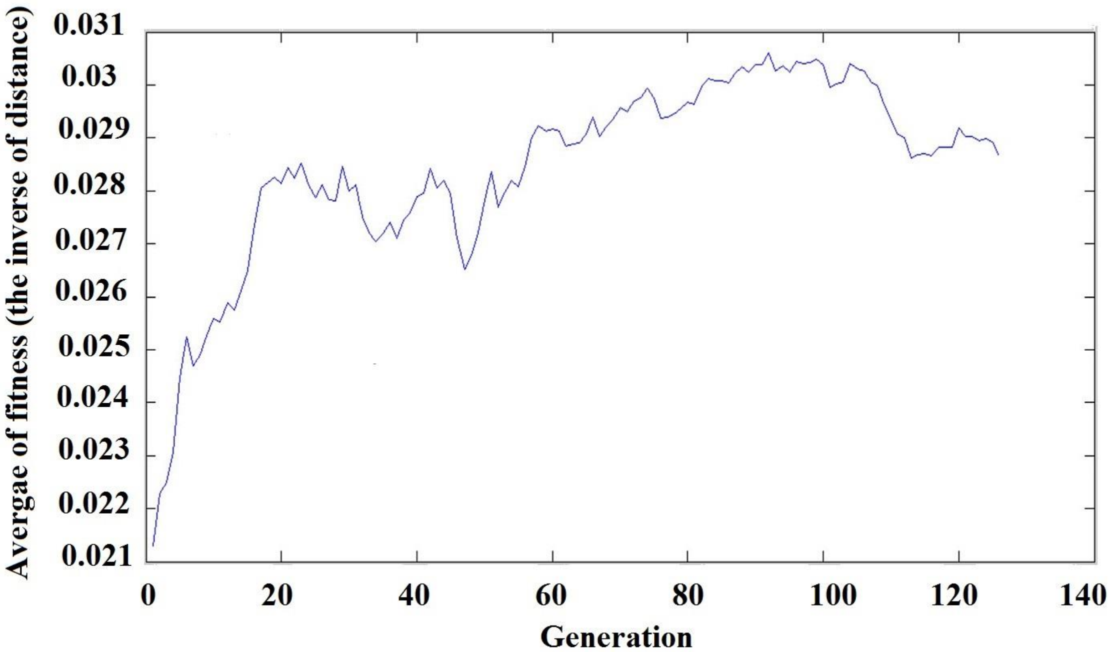


Figure 5. The trend of average fitness improvement for successive generations

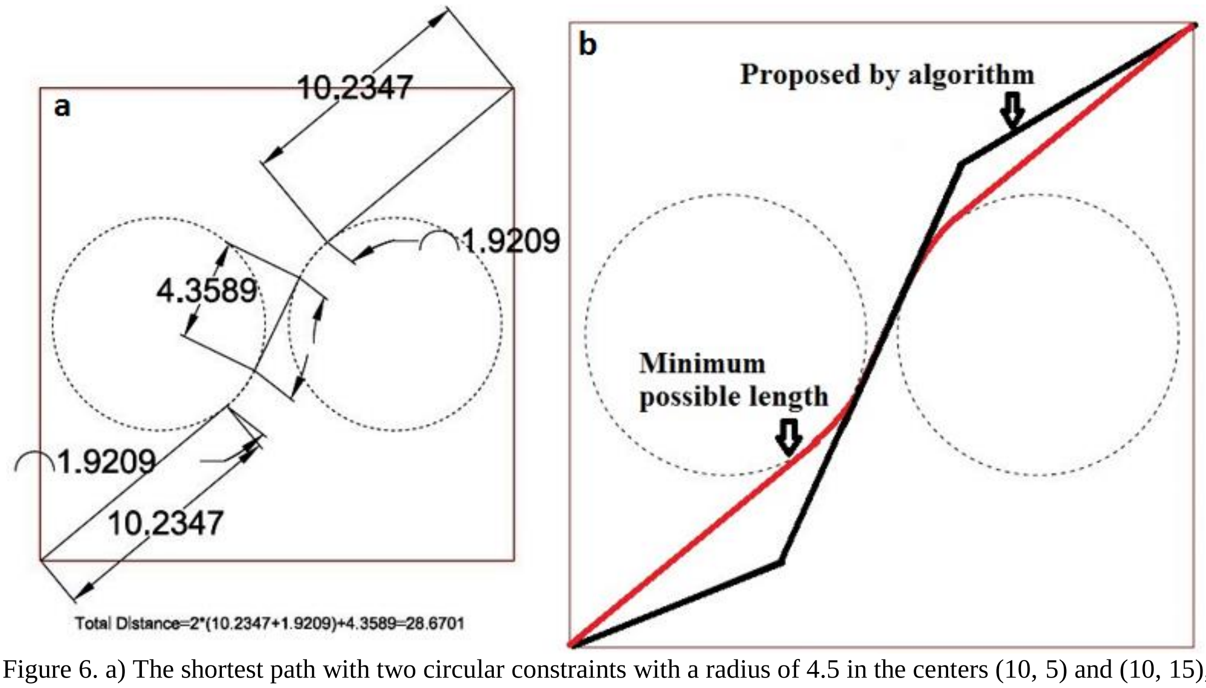


Figure 6. a) The shortest path with two circular constraints with a radius of 4.5 in the centers (10, 5) and (10, 15), b) Comparison of the suggested path of the algorithm and the best possible path

### *4.4 The fourth case: path optimization in four-dimensional space*

The starting point is considered with coordinates (0, 0, 0, 0), and the endpoint with coordinates (20, 20, 20, 20). One obstacle in the middle of the space is considered in Figure 8. In this case, it is not possible to visualize the geometrical situation. In any case, the output of the algorithm after 46 generations will be as shown in Figure 9.

```
Please enter INITIAL POINT in form of [. . .] : [0 0 0 0]
Please enter FINAL POINT [. . .] : [20 20 20 20]
Please enter number of variables : 4
Please enter number of constraints : 1
Please enter constraint in form of f(x(i))& in QUOTATION :
'(x(1)-10)^2+(x(2)-10)^2+(x(3)-10)^2+(x(4)-10)^2-25'
Please enter number of Points must be checked within 2 point : 5
Please enter number of Chromosome : 60
Please enter number of Tornament set : 6
Please enter amount of Convergence Condition(Rank_ave/Max_Rank : 0.95
Please enter max Chromosome Length of ZERO generation : 5
```

Figure 8. Inputs for 4-dimensional space

```
Number_of_Iteration =46
Maximum_Rank =0.0237
Selected_Chromosome_for_Solution =29
Point_Number_0f_Solution =1
Coordinate of POINTS of selected chromosome are :
12.4675 6.8351 16.0493 11.2891
The_Best_total_distance =42.1634
```

Figure 9. Algorithm output for the inputs of Figure 8

### *2.5 The fifth: path optimization for a practical case*

Figure 10 demonstrates the layout of a typical woodworking workshop. In this workshop, several types of machines are used in different workstations. Let's assume that parts are to be transported from workstation A to workstation B by a transporter.

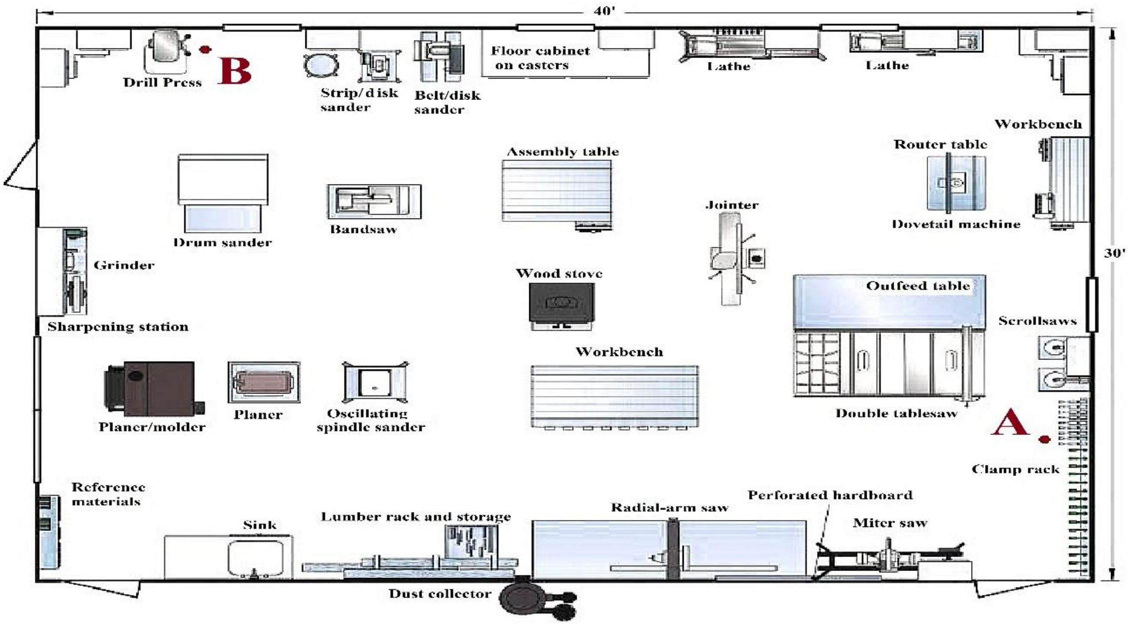


Figure 10. Arrangement of equipment in a woodworking workshop [19]

As depicted in Figure 11, point A is the starting point with coordinates (0, 0) and point B is 9.8 meters and 6.56 meters to the left and above point A, respectively. All obstacles are presented by rectangles in this case. After 156 iterations, the algorithm provides a path with a length of 12.9881 m, while the best point-to-point path has a length of 12.1224 m, containing an error of about 7.1%.

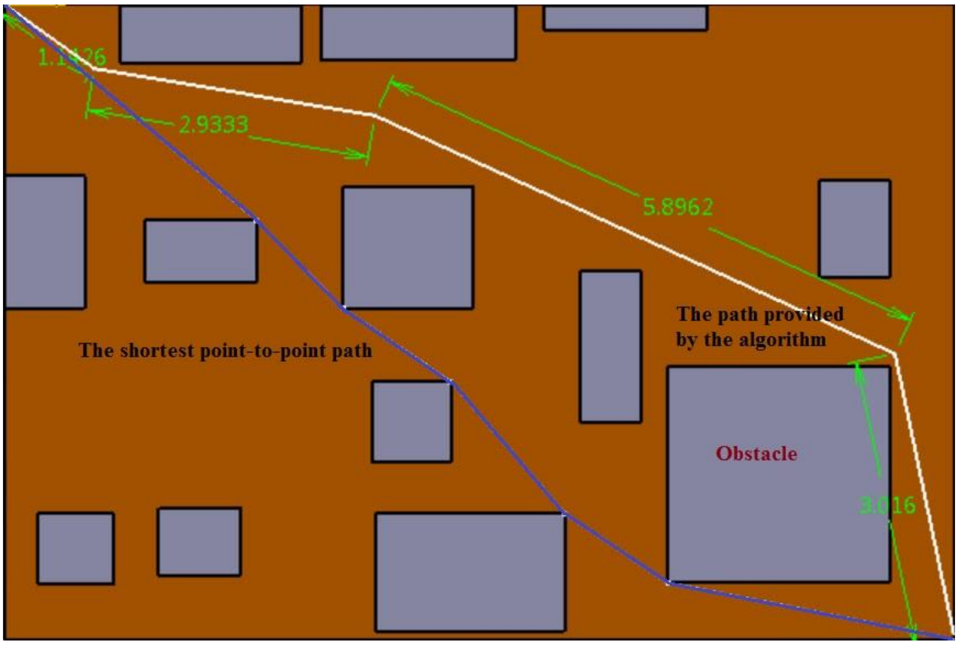


Figure 11. Comparison of the real shortest path and the shortest path provided by the algorithm in presence of rectangular obstacles

In the unconstrained mode, in the first case, even with a high number of chromosomes, the program gets the answer quickly. However, the calculation time increases rapidly with the increase in the number of chromosomes and especially the number of constraints, and it may take several minutes for the number of chromosomes to be 40 and with 5 checkpoints in the connecting lines to satisfy the convergence condition. In the constrained state, because every time a chromosome is created, the admissibility of the child chromosomes and the points of the connected lines must be checked, and if even one point is not admissible, the whole process of tournament selection and cutting and splicing must be done again, so it takes a lot of time.

Another important point is that the use of elitism greatly increases the quality of the chromosome output of the algorithm. In almost all runs, the selected chromosome was in a generation other than the last generation. Also, the examination of the output of several executions shows that the best chromosome was made in the first few generations (below ten generations), and the next generations did not provide a better chromosome.

By calculating the difference between the ideal path and the path provided by the proposed algorithm for three two-dimensional investigated cases, according to Table 1, it is determined that the average error value will be equal to 3.4%.

Table 1. The error percentage of the proposed paths compared to the ideal paths

| Case | First | Second | Third | Fifth |
|---|---|---|---|---|
| Error (%) | 0.03 | 2.02 | 4.45 | 7.1 |

## 5. Conclusions

In this research, the messy genetic algorithm, which can use chromosomes with variable lengths, was investigated to optimize the path of moving objects in the presence of obstacles. The use of a messy genetic algorithm provides the possibility of using polyline paths with various numbers of line segments which increases flexibility in finding the right path. The results of multiple executions of the algorithm with different obstacles showed that the proposed method can identify the admissible paths among the obstacles with high quality and provide the shortest path with an approximate average difference of 3.4% compared to the ideal path. The proposed algorithm can bypass obstacles even in multi-dimensional spaces and provide the optimal path without collision for the moving object.

## 6. References

[1] Tzafestas, S.G. 2013. Introduction to Mobile Robot Control. Elsevier.

[2] Mallahi Kolahi, P. and Mosayebi, M. 2021. Optimal trajectory planning for an industrial mobile robot using optimal control theory. Journal of Modern Processes in Manufacturing and Production. 10(3): 25-34.

[3] Matveev, A.S., Savkin, A.V., Hoy, M. and Wang, C. 2016. Survey of algorithms for safe navigation of mobile robots in complex environments. Safe Robot Navigation among Moving and Steady Obstacles. Butterworth-Heinemann, Elsevier.

[4] Esmaili, M. and Saadat, M. 2020. Path planning and control of an industrial robot used for opening tap hole of an electric arc furnace. Journal of Modern Processes in Manufacturing and Production. 9(4): 5-14.

[5] Esmaeili, M. and Saadat, M. 2019. Dynamic modeling of a robot manipulator for opening the tap hole of an electric arc furnace. Journal of Modern Processes in Manufacturing and Production. 8(4): 17-24.

[6] Nazarahari, M., Khanmirza, E. and Doostie, S. 2019. Multi-objective multi-robot path planning in continuous environment using an enhanced genetic algorithm. Expert Systems with Applications. 115(1): 106-120.

[7] Ever, Y.K. 2017. Using simplified swarm optimization on path planning for intelligent mobile robot. Procedia Computer Science. 120(1): 83-90.

[8] Nasrollahy, A.Z. and Javadi, H.H.S. 2009. Using particle swarm optimization for robot path planning in dynamic environments with moving obstacles and target. Proceeding Third UKSim European Symposium on Computer Modeling and Simulation. Athens, Greece.

[9] Halder, R.K. 2021. Particle swarm optimization in global path planning for swarm of robots. Springer International Publishing.

[10] Ge, S.S. and Cui, Y.J. 2000. New potential functions for mobile robot path planning. IEEE Transactions on Robotics and Automation. 16(5): 615-620.

[11] Brand, M., Masuda, M., Wehner, N. and Xiao-Hua, Y. 2010. Ant colony optimization algorithm for robot path planning. Proceeding of International Conference on Computer Design and Applications. Qinhuangdao, China.

[12] Dai, X., Long, S., Zhang, Z. and Gong, D. 2019. Mobile robot path planning based on ant colony algorithm with a heuristic method. Frontiers in Neurorobotics. 13(1): 1-9.

[13] Latombe, J.C. 2012. Robot motion planning. Springer Science & Business Media.

[14] Lamini, C., Benhlima, S. and Elbekri, A. 2018. Genetic algorithm based approach for autonomous mobile robot path planning. Procedia Computer Science. 127(1): 180-189.

[15] Nagib, G. and Gharieb, W. 2004. Path planning for a mobile robot using genetic algorithms. International Conference on Electrical, Electronic and Computer Engineering, Cairo, Egypt.

[16] Jianping, T. and Yang, S.X. 2003. Genetic algorithm based path planning for a mobile robot. Proc. IEEE International Conference on Robotics and Automation. Taipei, Taiwan.

[17] Ou, J., Hong, S.H., Ziehl, P. and Wang, Y. 2022. Gpu-based global path planning using genetic algorithm with near corner initialization. Journal of Intelligent & Robotic Systems. 104(2): 1-17.

[18] Goldberg, D.E., Korb, B. and Deb, K. 1989. Messy genetic algorithms: Motivation, analysis, and first results. Complex Systems. 3(5): 493-530.

[19] Https://www.Woodmagazine.Com/workshop/inspiring-shops/group-dynamic.