Enhanced Genetic Algorithm Based on Node Codes for Mobile Robot Path Planning

Dr. Mohamed Jasim Mohamed* **Mrs. Farah S. Khoshaba***

e-mail: moh62moh@yahoo.com **e-mail**: farah_sami76@yahoo.com

Received on: 21/9/2011
Accepted on: 22/5/2012

Abstract: In this paper, a new Enhanced Genetic Algorithm (EGA) is used to find the best global path planning for a mobile robot according to a specific criterion. The EGA is enhanced by a new encoding method, new initial population creation method, new crossover and mutation operations as well as new additional operations correction operation and classification operation. The study considers the case when the mobile robot works in a known static environment. The new proposed algorithm is built to help the mobile robot to choose the shortest path without it colliding with the obstacles allocated in a working known environment. The use of grid map in the environment helps to locate nodes on the map where all nodes are assigned by coordinate values. The start and the target nodes of the required path are given prior to the proposed algorithm. Each node represents a landmark that the mobile robot either passes through only one time or never passes through during its journey from start node to the target node. Two examples of known static mobile robot environments with many obstacles in each one are studied and the proposed algorithm is applied on them. The results show that the proposed algorithm is very reliable, accurate, efficient and fast to give the best global path planning for the two cases.

Keywords: **Mobile Robot, Optimization, Genetic Algorithm, Fitness Function, Global Path Planning**

*** Control and Systems Engineering Department / University of Technology**

1. INTRODUCTION

Optimization is maybe the process of adjusting the inputs to, or characteristics of a device, mathematical process, or experiment to find the minimum or maximum output or result (i.e. Optimization is the process of making something better) [1].

Path planning is one of the most important fields of research in the area of robotics. It is usually defined as generating a collision-free path, between two specific locations in an environment with obstacles and optimizing it with respect to some criterion [2]. Path planning can be divided into two major categories: local path planning and global path planning.

Local Path Planning, which is also called on-line path planning, is a set of algorithms that plan and execute a path at the same time. This task is based on feedback from a variety of sensors such as sonar and laser. Every task, like obstacle avoidance and backing, is analyzed based on data obtained from these sensors.

The Global Path Planning which is also called off-line path planning is a set of algorithms that process the existing data before the vehicle starts moving. In this way the user can visualize the path and detect possible errors in the algorithm. This method provides the safest possible path planning [3]. In [4], the authors present an effective method to achieve both obstacle-avoidance and target-tracking for an autonomous mobile robot in an indoor environment. They employ a wall-following algorithm using neural network pattern recognition to avoid obstacles. An autonomous mobile robot reaches a given target by the tracking algorithm. In case obstacles are detected by sonar sensors, an autonomous mobile robot avoids collision with obstacles by the wall-following algorithm. They propose a simple making method to avoid being trapped in a local minimum which was a serious problem in local path planning. Kolushev and Bogdanov [5] presented a novel approach to multi-agent optimal path planning using graph representation of environment models. When planning the path of each robot, the graph model of environment is dynamically changed for path correction and collision avoidance. The new algorithm applies changes of robot's paths and speeds to avoid collisions in a multi-agent environment. A method based on Ant Colony Optimization Meta-Heuristic (ACO-MH) to find the optimal path in a previously defined static search map is presented in [6]. The proposed algorithm supports the avoidance of dynamic obstacles; that is, once the optimal path is found and the robot starts navigating, if the robots route is interrupted by a new obstacle that was sensed at time *t*, it will recalculate an alternative optimal path from the actual robot position in order to surround this blocking object and reach the goal. Also [7] proposed a path planning for a mobile robot based on chaos GA where a reasonable coding way and fitness function are used. The chaos operation is added to the GA; the solution obtained by chaos GA may not only satisfy the shortest but also the most effective path to avoid the collision with an obstacle. Kambhampati & Davis in [8] presented an approach to automatic path planning based on a quad-tree representation. They demonstrated the merits of quad tree-based path planning and also discussed in detail a method of staged path planning with improved computational cost compared to pure quad-tree based single stage path planning. The GA, which was developed by John Holland (1975), is an optimization and search technique based on the principles of natural genetics and natural selection. GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the "fitness" (i.e., minimizes the cost function). It begins, like any other optimization algorithm, by defining the optimization variables, the cost function, and the cost. It ends like other optimization algorithms too, by testing for convergence.

Mobile robots imitate biological movement, and GA imitates biological survival [1]. The two topics seem to be a perfect match. Many researchers have made that connection, especially for the problem of finding optimal path, which has a great attention in the recent years. Many researchers have research in this field, for example, [9] proposes a new path planning method based on neural network and GA. The method constructs a neural network model of environmental information in the workspace for a robot, and then establishes the relationship between a collision avoidance path and the output of the model. This GA was applied to find the global optimal path in static environment.

This paper focuses on off-line path planning and uses EGA to find the best path of the mobile robot to reach its target. The terminology "best" path implies that there are many paths, which are not of equal lengths. The definition of "best" is relative to the problem at hand, its method of solution, and the tolerances allowed. Here, the required path is optimal in the sense of the shortest distance and the less effort done by the mobile robot.

2. PROBLEM DESCRIBTION

In our problem, we have a mobile robot moving in a known static environment. The environment space has many obstacles located in specific positions. The mobile robot's mission is to choose offline the shortest path to travel from a start node to a target node without colliding with the obstacles that appear on its way.

3. THE GENETIC ALGORITHM

In general, GA starts with a random initialization of individuals. The fitness for each individual is evaluated. Then the genetic operators' selection, crossover and mutation are applied. Thus new individuals are produced from this optimization process, which then results in the next population. Encoding of chromosomes is the first question to be asked when starting to solve a problem with GA. It depends on the problem heavily [10] [11]. There are many possible individuals' encodings. Binary encoding (or bit strings), is the most common way of encoding mainly because the first researchers of GA used this type of encoding and secondly owing to its relative simplicity. On the other hand, this encoding is often not natural for many problems and sometimes corrections must be made after crossover and/or mutation operations [11]. In binary encoding, every chromosome is a string of bits 0 or 1.

Crossover and mutation are two basic operators of GA. The performance of GA depends on them very much. The type and implementation of these operators depend on the encoding and also on the problem [10]. There are also many ways to perform crossover; some of these are: single point crossover, two-point crossover, uniform crossover, and arithmetic crossover.

4. PROPOSED ALGORITHM

In order to use EGA to solve the path-planning problems, a number of preparation steps are needed. These steps are;

1) Convert the search known environment to a gird map.
2) Choose the necessary and minimum number of nodes on the map to allow the mobile robot reach to each demanded place.
3) Specify an integer symbol for each node.
4) Specify the existent subpaths between nodes (i.e. the allowable node linkages between each other).
5) Calculate the coordinate values for each node.
6) Count the number of static obstacles in the known search environment.
7) Determine the starting node and target node in order to establish the required path.

5. PROBLEM FORMULATION

a) *Encoding of Chromosome*

The nodes are used to encode a path as a string of an ordered integer symbols. The first integer in the string represents the starting node while the last integer in the string is the target node. Each chromosome represents a candidate solution to the optimization problem. Thus the chromosome represents a path which consists of straight line segments; the first node indicates the starting point of first line segments while the first intermediate node is the end point. The second line segment is the line connected between the first and the second intermediate nodes. Other line segments are similarly constructed from the connection between any two neighbor intermediate nodes. The final line segment is the connection between the last intermediate node and the target node. The maximum chromosome length is equal to the number of map nodes times the gene length.

In this case, we have 16 nodes where each node is coded by one integer location. The chromosome length is equal to 16 integers. This result is straight forward but the chromosome length can be reduced to save the computation power. One can observe from map topology that the chromosome length of the shortest path between a starting node and a target node is proportional to the number of obstacles. When the number of static obstacles is equal to n, the shortest path consists of at most of (n+2) nodes or (n+1) linear segments. This relation assumes that the obstacle shapes are not complex and can be considered as mass points. When there are no obstacles in the environment n=0, the shortest path consists of a one liner segment from the start node to the target node. If there is one obstacle in the environment, the shortest path consists of two linear segments. Thus, the maximum number of chromosome length will be determined by the equation (1).

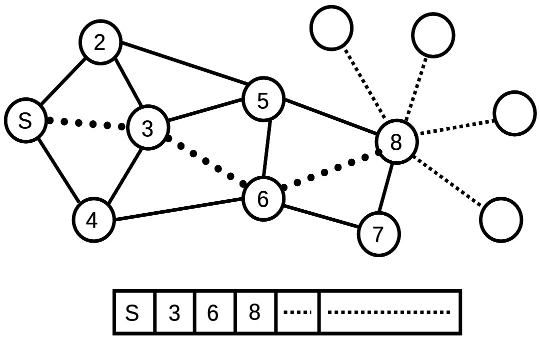
$$L = n+2 \tag{1}$$

Where, L is the number of genes in the chromosome, n number of static obstacles.

According to this relation variable length chromosome will be proposed in this paper with maximum chromosome length is equal to L and minimum length is 2.

b) *Initial Population.*

The initial population is generated in two manners. Each manner generates half the number of chromosomes of the total population. The starting node and target node are fixed in each chromosome. In the first manner, we begin from the starting node, if for example the starting node is linked to three nodes; one of these linked nodes is chosen randomly and fixed as the first intermediate node. The first line segment is the connection between these start and first intermediate nodes. Now, the path reaches the first intermediate node, in the same way, we look to the number of linked nodes to this intermediate node, and choose randomly one node to be the second intermediate node. The second line segment is the connection between these two intermediate nodes. The procedure continues till the location of last intermediate node is filled as shown in Figure (1).



Figure(1). The first manner of creation chromosome

In the second manner, we begin from the target node. The last intermediate node is chosen randomly from the number of linked nodes to the target node. The node before the last intermediate node is chosen randomly from the number of linked nodes to the last intermediates node. The procedure continues till the first intermediate node location is filled as shown in Figure (2).

We classify the chromosomes in a number of categories. So, we add one location in the chromosome for this purpose to discriminate the type of the chromosome.

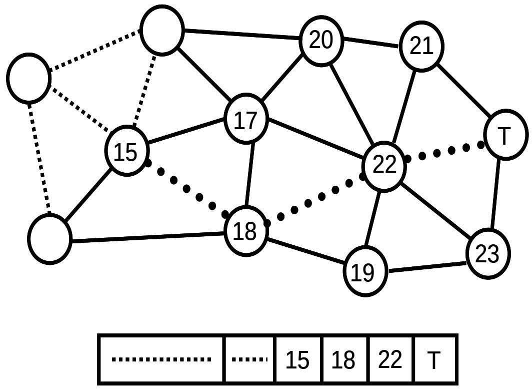


Figure (2). The second manner of creation chromosome

The chromosome generated by the first manner is of type *A*, but if the target node of this chromosome is one of the linked nodes to the last intermediate node, the chromosome is of type *C*. The chromosome generated by the second manner is of type *B*, but if the first intermediate node is one of the linked nodes to the start node the chromosome is of type *C*.

Any chromosome, which results from crossover or mutation operations that will not meet the above three types, will be of type *D*. The classification operation is applied in each generation for all chromosomes in the population.

c) *Creation Operation.*

The creation operation is unique in that it does not require any existing chromosome. It creates an entire new chromosome in the same way that a chromosome in the initial population is created. This operation is applied to create a predefined number of new chromosomes for the new population in each generation. The created chromosomes are of type *A* and *B*.

d) *Selection Operation*.

In the EGA, a tournament selection is used to create a new population. In this method, two chromosomes are chosen at random from the population. A random number r is then chosen between 0 and 1. If $r < k$ (where k is a parameter, for example 0.75), the fittest of two chromosomes is selected to be a parent chromosome, otherwise the less fit chromosome is selected. The two parental chromosomes are then returned to the original population and can be selected again. Elitist mode is also used where a certain number of best parental chromosomes of each generation are kept in the next generation, so as to improve the efficiency of the EGA.

e) *Crossover Operation.*

Two new proposed types of crossover operations are implemented on the selected chromosomes. The start node and target node are not contributed in the crossover operations, only the intermediate nodes are considered. During the crossover operation, a chromosome that has an efficient fitness value is randomly selected as a parent. The first parent is selected of type *A* or *C* chromosome while the second parent is selected from type *B* or *C* chromosomes. The two methods of crossover operations are illustrated as follows:

***Method*-1-:** If there are same intermediate nodes in the two parents and the crossover between them does not result in any child of chromosome length more than the maximum length, this node is possible to be a crossover point.

All these possible crossover points are counted. Then, one of them is chosen randomly to make the crossover operation. Two parents are taken from the environment map shown in Figure (7) to illustrate the first type of crossover operation in Figure (3).

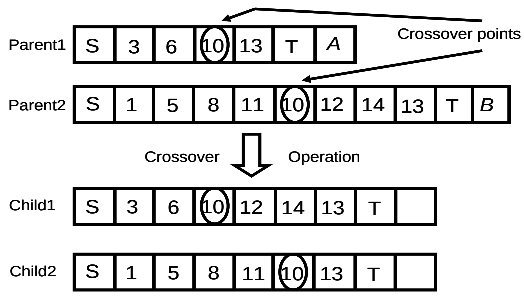
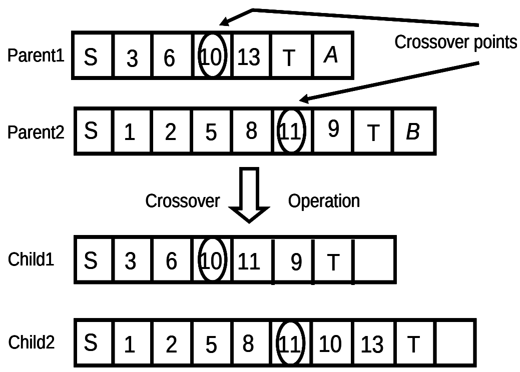


Figure (3). The first method of crossover operation.

***Method*-2-:** when there are not the same intermediate nodes in the two parents, for each node in first parent, a search is made in the second parent for a node which is one of the linked nodes to it. If the linking between them by crossover operation does not result in any child of chromosome length more than the maximum length, the crossover operation can made. All these possible states are counted. Then, one of them is chosen randomly to make the crossover operation.

Two parents are taken from the environment map shown in Figure (7) to illustrate the second type of crossover operation in Figure (4).



Figure(4). The second method of crossover operation

If the two methods of crossover operations cannot be implemented in this case, mutation is made for these two parents.

f) *Mutation Operation*

This operation is an asexual operation that operates on only one parental chromosome. The parental chromosome is selected based on fitness measure. One node is selected randomly from parental chromosome. The linked nodes to the node before this selected node are considered. Then one of these linked nodes is chosen randomly to replace the original selected node. The chosen of new linked node is accomplished using the table of linked nodes as shown in Table (1). The mutation operation is illustrated in Figure (5).

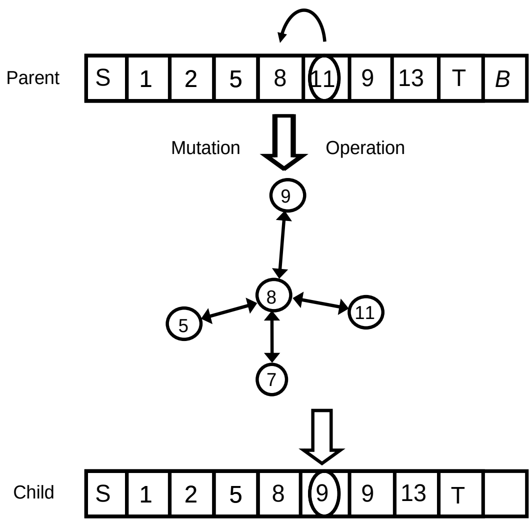


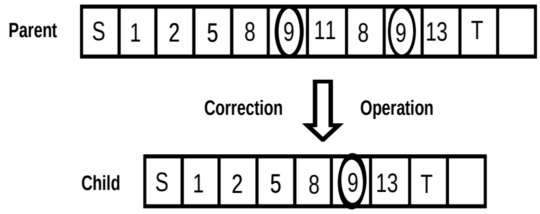
Figure (5). The proposed mutation operation

g) *Correction operation.*

This is a new proposed operation. The name "correction" has been chosen for this operation as an indication to the process of correcting a chromosome path. This operation eliminates the loops that may exist in the path during traveling from start node to target node.

The chromosomes are created or yielded from creation, crossover and mutation operations during generations. Some of these chromosomes may have repeated nodes. This means there are one or more loops in a chromosome path. The correction operation is implemented to test if the chromosome has repeated nodes.

When there are loops in the path, these loops are removed in order to shorten the length of the chromosome to minimum value. The correction operation is illustrated in the Figure (6).



Figure(6). The proposed correction operation

This operation is applied for all chromosomes of the population in each generation.

h) *Classification Operation.*

The new chromosomes resultant from crossover and mutation operations have an unknown type of chromosomes because they are constructed from parts of their parents.

So, this operation is used to classify each chromosome of the new population in order to use them correctly in the next generation.

i) *Fitness and Evaluation.*

The fitness can be considered the most difficult and important part in EGA. The success of EGA to find the optimal chromosome depends mainly on the selection of appropriate fitness measures.

The fitness is a number assigned to a chromosome representing a measure of goodness. In our problem the shortest path between start node and target node is an optimal path. Thus, the fitness measure must be able to guide the EGA to find this shortest path. Since each path consists of many segments of straight lines, the distance can be calculated by equation (2):

$$\text{Distance} = \sum_{i=1}^{n+1} \sqrt{(X_{i+1} - X_i)^2 + (Y_{i+1} - Y_i)^2} \quad ..(2)$$

Since there are a number of types of chromosome, then any chromosome results in an unfeasible path is punished. The chromosomes of type *D* are getting punishment in order to have very small fitness value. The chromosomes of types *A* and *B* are getting relatively small punishment. While the chromosomes of type *C* are not getting any punishment because they are feasible chromosomes. The fitness function is illustrated by equation(3):

$$\text{Fitness} = \frac{1}{\text{Distance} + \text{Punishment}} \quad ..(3)$$

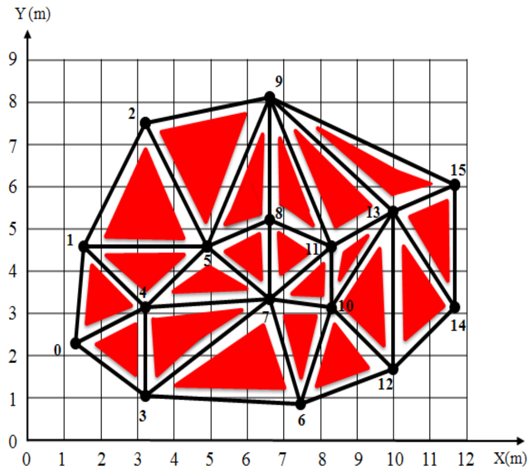
6. FINDING THE OPTIMAL PATH

In this paper, two examples are suggested and expressed in two maps of static environments with many obstacles. We assume the mobile robot works in these two known environments and can reach any point in the environment without colliding with the existent obstacle.

Example-1: The map of the first example is shown in Figure (7).

When we apply the steps of the proposed algorithm on this map, we find the minimum number of required nodes is 16 and the number of obstacles in the environment is 21.

The coordinate values of any node and the nodes linking to it through the subpaths are shown in Table (1).



Figure(7).Environment map of example-1

Table(1). Nodes linking and their coordinates values for the map of example-1.

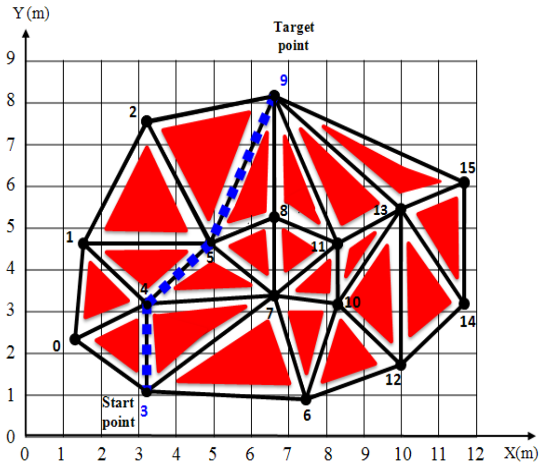
Node No.	Node code	X value (m)	Y value (m)	Linked Nodes
0	0	1.3	2.3	1,3,4
1	1	1.5	4.6	0,2,4,5
2	2	3.2	7.6	1,5,9
3	3	3.2	1	0,4,6,7
4	4	3.2	3.1	0,1,3,5,7
5	5	4.9	4.6	1,2,4,7,8,9
6	6	7.5	0.8	3,7,10,12
7	7	6.6	3.3	3,4,5,6,8,10,11
8	8	6.6	5.2	5,7,9,11
9	9	6.6	8.2	2,5,8,11
10	10	8.3	3.1	6,7,11,12,13
11	11	8.3	4.6	7,8,9,10,13
12	12	10	1.6	6,10,13,14
13	13	10	5.5	9,10,11,12,14, 15
14	14	11.7	3.1	12,13,15
15	15	11.7	6	9,13,14

In the proposed EGA the chromosome is represented as a string of integers, each node represented by one integer, the chromosome length is a variable with maximum value 16 and minimum value 2. Tournament selection with elitism strategy is used (retain two best individuals), Creation rate $\mathbf{P_r}$ = 0.2, Crossover rate $\mathbf{P_c}$=0.65, Mutation rate, $\mathbf{P_m}$=0.15, Population size=100, Maximum generation 1000 for each run.

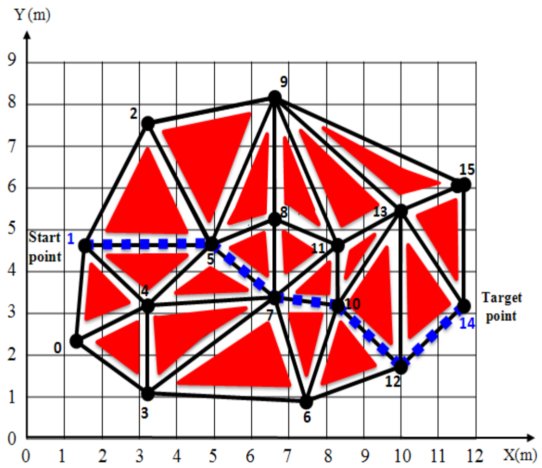
The results of some optimal paths between certain start nodes and certain target nodes are shown in Table (2). Figures (8,9,10) show the optimal paths on the map of the environment.

Table(2).Some optimal paths for example-1.

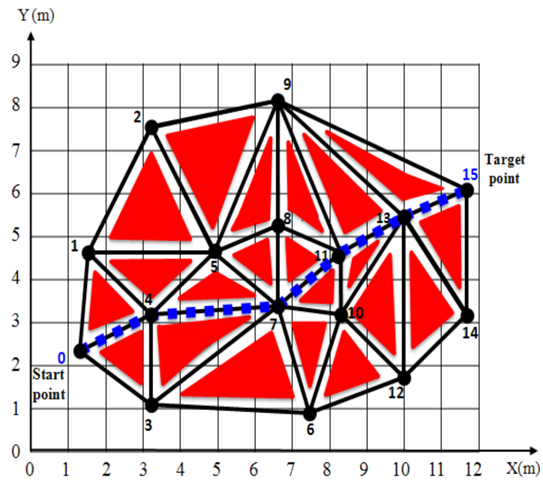
Start node	Intermediate nodes or optimal path	Target node	The distance of path
3 →	4 → 5	→ 9	**8.34836**
1 →	5 → 7 → 10 → 12	→ 14	**11.7861**
0 →	4 → 7 → 11 → 13	→ 15	**11.3030**



Figure(8). The optimal path from node 3 to node 9 for example-1

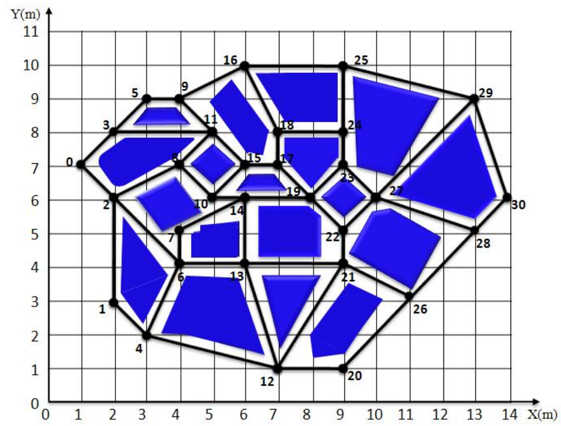


Figure(9). The optimal path from node 1 to node 14 for example-1



Figure(10). The optimal path from node 0 to node 15 for example-1

Example -2: The map of the second example is shown in Figure (11).



Figure(11).Environment map of example-2

When we apply the steps of the proposed algorithm on this map, we find that the minimum number of required nodes is 31 and the number of obstacles in the environment is 18. The coordinate values of each node and the nodes linking to it through subpaths are shown in Table (3).

Table(3).Nodes linking and their coordinates values for the map of example-2

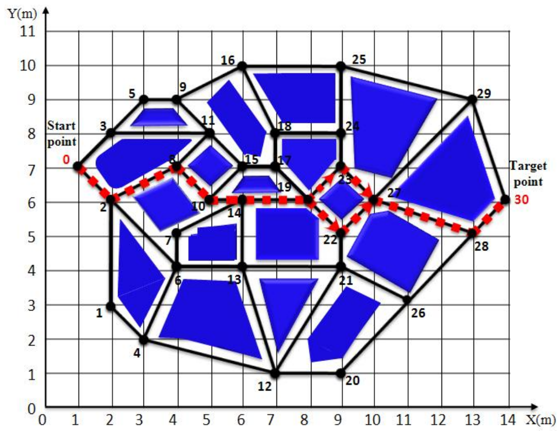
Node number	Node code	X Value (m)	Y Value (m)	Linked Nodes
0	0	1	7	2,3
1	1	2	3	4,2
2	2	2	6	0,1,6,8
3	3	2	8	0,5,11
4	4	3	2	1,6,12
5	5	3	9	3,9
6	6	4	4	2,4,7,13
7	7	4	5	6,14
8	8	4	7	2,10,11
9	9	4	9	5,11,16
10	10	5	6	8,14,15
11	11	5	8	3,8,9,15
12	12	7	1	4,13,20,21
13	13	6	4	6,12,14,21
14	14	6	6	7,10,13,19
15	15	6	7	10,11,17
16	16	6	10	9,18,25
17	17	7	7	15,18,19
18	18	7	8	16,17,24
19	19	8	6	14,17,22,23
20	20	9	1	12,26
21	21	9	4	12,13,22,26
22	22	9	5	19,21,27
23	23	9	7	19,24,27
24	24	9	8	18,23,25
25	25	9	10	16,24,29
26	26	11	3	20,21,28
27	27	10	6	22,23,28,29
28	28	13	5	26,27,30
29	29	13	9	25,27,30
30	30	14	6	28,29

The chromosome of proposed EGA is represented by a string of integers, each node represented by one integer, the chromosome length is a variable with maximum value 20 and minimum value 2. Tournament selection with elitism strategy are used (retain two best individuals), Creation rate **P_r**=0.2, Crossover rate, **P_c**=0.65, Mutation rate, **P_m**=0.15, Population size=100, Maximum generation is 1000 for each run. The results of some optimal paths between some start nodes and target nodes are shown Table (4).

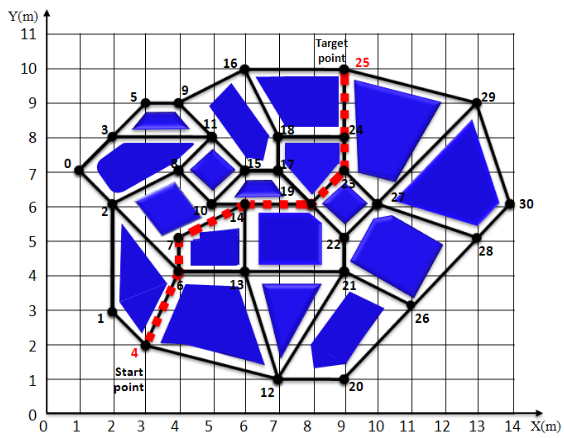
Table (4). Some optimal paths for example-2.

Start node	Intermediate nodes or optimal path	Target node	The distance of path
0 → 0 →	2,8,10,14,19,22,27,28 2,8,10,14,19,23,27,28	→ 30 → 30	15.46941
4 →	6,7,14,19,23,24	→ 25	11.88634
9 → 9 →	11,8,10,14,13,12 11,15,10,14,13,12	→ 20 → 20	12.40491

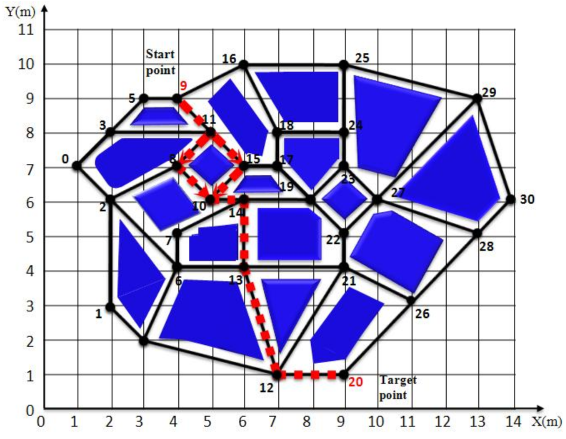
Figures (12,13,14) show the optimal paths on the map of the environment.



Figure(12). The optimal path from node 0 to node 30 for example-2



Figure(13). The optimal path from node 4 to node 25 for example-2



Figure(14). The optimal path from node 9 to node 20 for example-2

7. CONCLUSIONS & DISCUSSION

In this paper, a new path planning method based on EGA is introduced. The EGA is proposed to find the optimal path in two dimensional known environments offline. The grid is used to establish the working environment of a mobile robot. The nodes labels are used to encode the chromosome as a string of integers. The EGA is enhanced by the new method of creating initial population, two types of crossover operations, mutation operations and new additional operations which are called correction operation to shorten the length of the chromosome, and classification operation to classify the chromosomes. The *statistic* results of the two examples are shown in Table (5).

In order to ensure the ability of the proposed EGA to solve path planning problem, different optimal paths with different start nodes and target nodes are taken for two environment maps. Moreover 10 runs are taken for each case. The results show that the EGA can find the optimal path with probability 100% because all runs succeeded to find the optimal path in all cases as shown in the Table (5).

Table(5).The statistic results of two examples

	Start node	Target node	Number of runs	Number of Succeeded run	The average of generations for Succeeded runs
Exp -1	**3**	**9**	**10**	**10**	**1.1**
	1	**14**	**10**	**10**	**5.7**
	0	**15**	**10**	**10**	**4.8**
Exp -2	**0**	**30**	**10**	**10**	**45.5**
	4	**25**	**10**	**10**	**12.6**
	9	**20**	**10**	**10**	**5.3**

The average of generations for success runs is relatively small and so it takes a reasonable time. Therefore, the EGA can be implemented online to find the optimal path for the mobile robot. Finally the simulation results prove that the EGA is very reliable, accurate, efficient and fast to find the best global optimal path planning for the mobile robot in different known environments.

REFERENCES

[1] Randy L. Haupt & Sue Ellen Haupt, "Practical Genetic Algorithms", 2nd edition, John Wiley & Sons, Inc., publication, 2004.

[2] Sugihara, K. and Smith, J., "Genetic Algorithms for Adaptive Motion Planning of an Autonomous Mobile Robot", Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, CA, PP.138-146, 1997.

[3] Hurezeanu V., "Path Planning Software and Graphics Interface for an Autonomous Vehicle, Accounting for Tarrain Features", M.Sc., Graduate School of the University of Florida, 2000.

[4] O. Hachour, “Path planning of Autonomous Mobile robot”, international journal of systems applications, engineering & development**,** Issue 4, Volume 2, 2008.

[5] Kolushev F.A, & Bogdanov A.A., “Multi-Agent Optimal Path Planning for Mobile Robots in Environment with Obstacles”, PSI’99, lncs1755, pp.503-510, 2000.

[6] Meijuan Gao, Jin Xu, Jingwen Tian, Hao Wu, "Path Planning for Mobile Robot Based on Chaos Genetic Algorithm," icnc, vol. 4, pp.409-413, 2008 Fourth International Conference on Natural Computation, 2008.

[7] Meijuan Gao, Jin Xu & Jingwen Tian, Hao Wu, "Path Planning for Mobile Robot Based on Chaos Genetic Algorithm," icnc, vol. 4, pp.409-413, 2008 Fourth International Conference on Natural Computation, 2008.

[8] Kambhampati S. & Davis L.S., “Multi resolution Path Planning for Mobile Robots”, IEEE Journal of Robotics and Automation, Vol.RA-2, No.3, September 1986.

[9] DU Xin, CHEN Hua-hua & GU Wei-kang, “Neural network and genetic algorithm based global path planning in a static environment” Du et al. / J Zhejiang University SCI pp.549-554, 2005.

[10] Mitchell M.; "An Introduction to Genetic Algorithms ", 1st MIT press paperback edition, Cambridge, Mass, 1998.

[11] Goldberg D. E. ;"Genetic Algorithms in Search, Optimization and Machine Learning", Addison-Wesley, Reading Mass 1989.