Enhanced GA for Mobile Robot Path Planning Based on Links among Distributed Nodes

Dr. Mohamed Jasim Mohamed
Control and Systems Engineering Department University of Technology /Baghdad.
Email: dr.mohamed_jasim@uotechnology.edu.iq
Mustaffa waad Abbas
Control and Systems Engineering Department University of Technology /Baghdad.

Received on: 17/1/2012 & Accepted on: 15/8/2012

ABSTRACT

In this paper, we propose an Enhanced Genetic Algorithm (EGA) to find the optimal path for a mobile robot. The workspace of the mobile robot is assumed to be of known environment with many static obstacles. The space of environment is divided into equally quarters by projection of a grid of specified distance on the environment space. Each quarter represents a node in the workspace. Moreover, each node has assigned by a unique number. So, all these nodes are distributed uniformly in the workspace. Each node may link to another node by straight line unless this line crosses one or more obstacles. New operator named *Validation of Links* operator introduces here to check all valid links between any two nodes. The GA operators adjusted and enhanced to suit the path planning problem and further more we develop new other operators to increase the efficiency of the algorithm. Simulation studies are carried out to verify and validate the effectiveness of the proposed algorithm.

Keywords: Genetic Algorithm GA, Mobile Robot, Global Path Planning, Fitness Function.

الخوارزمية الوراثية المعززة لتخطيطِ طريقِ الروبوت النقالِ المستندةِ على الروبط بين العقد الموزعة

الخلاصة

في هذا البحثَ ، أقترحنا خوارزميةَ وراثيةَ معززةَ جديدة لغرض أيجاد الطريق الأمثل للروبوت النقال. تم أفتراض مجالَ عملُ الروبوت النقال معروف البيئة مَع وجودِ عدةِ عوائقِ ثابتة فيه. إن مساحةَ البيئة قد قُسمتْ الى عدة مربعات متساوية بواسطة أسقاطِ شبكةَ ذات فتحَة معينة على مساحة البيئة. يمثل كل مربع عقدةْ في مجالِ العمل والاكثر من ذلك كل عقدةْ تم تخصيصُها بواسطة رقم منفرد. لذلك وكل هذه

https://doi.org/10.30684/etj.2013.71193
2412-0758/University of Technology-Iraq, Baghdad, Iraq

العقد موزعةٌ بشكل منتظم في مجال العمل ويمكن لكل عقدة أن ترتبطُ بعقدة اخرى بواسطة خط مستقيم ما لم يتقاطعْ هذا الخط المستقيمُ مع واحد او اكثر من العوائق. تم تقديم

مشغلً جديد هنا سُمي عامل التحقق من الارتباطات لغرض فحص كل الأرتباطات المتاحة بين أي عقدتين. وان مُشغلاتَ الخوارزمية الوراثية قد تمَ تكيفُها و تعزيزها لملائمة مشكلةٍ تخطيطِ طريق الروبوت التقال والاكثر من ذلك لقد انشائنا مشغلاتٌ اخرى جديدة لزيادة كفاءةِ الخوارزمية. ولقد تمَ دراسةُ المحاكاةِ لغرضِ التحققِ والمصادقة على فعالية الخوارزمية المقترحةُ.

INTRODUCTION

Path planning is an important issue in mobile robotics. The path planning is about finding a collision free path between two specific points (start and target points) in known environment with the satisfaction of some optimization criteria like; path length, velocity, acceleration and number of turns.

The path planning problem can be classified into two major categories. The first is global path planning that requires the environment to be completely known and the obstacles should be static. The second is local path planning which means that path planning is doing while the robot is moving. In other words, the algorithm is capable of producing a new path in response to environmental changes [1].

Path planning is a wide research area, where many algorithms have been applied to find the optimal path for mobile robot such as; A* algorithm [2], D* algorithm [3], reinforcement learning [4], Q-Learning [5], global C-space methods [6], potential field methods [7], fuzzy logic [8] and neural networks [9]. Each method has its own strength over others in certain aspects.

Generally, the main difficulties for robot path-planning problems are computational complexity, local optimum and adaptability. Researchers have been seeking alternative and more efficient ways to solve the problem. In the last decade, GAs has been widely used as an alternative method to generate the optimum path by taking advantage of its strong optimization ability. Where, GA requires no derivative information or formal initial estimates of the solution. Furthermore, GA is stochastic in its nature; it is capable of searching the entire solutions space with more likelihood of finding the global optimum [10].

The reminder of this paper is organized as follows: in section II a brief introduction to the GA is presented. In section III, the problem description is given. In section IV, the proposed algorithm is introduced and in section V, simulation results are presented. Conclusions and discussions are included in section VI.

GENETIC ALGORITHM

GA is a part of evolutionary computing, which is a rapidly growing area of artificial intelligence. The GA is an optimization and search technique based on the principles of natural selection and natural genetics. GA allows a population composed of many individuals to evolve under specified selection rules to a state that maximizes the “fitness” (i.e., minimizes or maximize the cost function).

In a GA, a population of strings (called chromosomes), which encode candidate solutions (called individuals) to an optimization problem, evolves toward better solutions. The evolutionary process usually starts from a randomly generated population. In each generation, the fitness of each individual in the population is evaluated which is determined by a problem specific fitness function. Then, multiple individuals are stochastically selected from the current population based on their fitness, and combined by crossover operation and\or undergo mutation operation. The new offspring's population is used in the next generation of the algorithm. This generation will continue until the optimization criterion is met or the number of maximum generations is reached.

PROBLEM DESCRIPTION

We have a mobile robot moves in closed and static environment. This environment contains a set of obstacles with different shapes and specified locations. The mobile robots mission is to choose the shortest path to travel from a start node to a target node without collides with the obstacles that appear on its way.

PROPOSED ALGORITHM

The proposed algorithm consists of two parts as illustrated below;

1) Data Collection

In order to formulate the path planning problem to a form that GA can deal with it, a number of necessary preparation operations are needed. These operations are concerning the description of the underlying environment space and collecting important information about it. The acquisition data from these operations facilities the work of EGA operators and make it more efficient. Each one of these operations results a table of information about the environment and implements only one time for each environment. These operations are illustrated in details as below.

a) Determination of Obstacles

In this operation, the number and locations of all obstacles are defined in a table called obstacles table. Each obstacle is represented by two points, the lower left corner and the upper right corner of a rectangle that surround this obstacle. These rectangles are the occupied area by the obstacles and restricted for the mobile robot. The rest area of the environment space is free where the mobile robot can moves in it. The occupied area by the obstacles is shown in Figure (1).

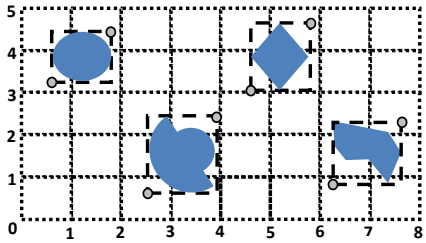


Figure (1): The occupied area by obstacles.

b)Distribution of Nodes.

In this operation**,** a grid of uniformly distributed nodes is projected on the whole environments space. Each node has defined locations, in (X , Y) coordinates space. Moreover, each node is assigned by an integer to represent that node. This integer is derived from the location of node as illustrated in the following equation:

$$N = (x + SSx * y) * 1/S \quad \ldots (1)$$

Where, N is the integer of the node, x and y are the of the coordinates values SSx is the location of node, x-axis of full length of the space, and the environment between S is the step next one. each node and the defined in These nodes are nodes a table called illustrated table. The nodes distribution is in Figure (2).

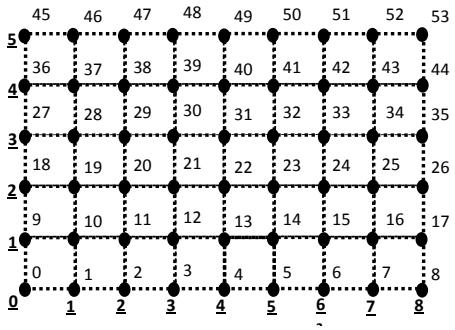


Figure (2): Nodes distribution on the space.

c) Validation of Links.

This operation is proposed to find all free collision links (straight lines) that

Connect each node with the rest. The functionality of this operation is based on the mathematical intersection between two straight lines; the first line represents the link between two nodes while the second line represents one side of the rectangle that defines the location of an obstacle. In other word, it is based on finding the solution of two straight-line equations and checking the intersection point whether lies in or not in the obstacle area. All links between each node and the other nodes are checked. If one of these links appears to be intersected with obstacles then this link is discarded. When the links checking operation is complete, all valid links for each node are stored in a table called links table.

2) THE EGA

In this section, we will describe in details the elements and the proposed operators of EGA.

a) Chromosome Encoding.

Before we get into EGA operators, we must describe the chromosomes encoding, which may be considered the key feature that all EGA operators depend on it.

In order to apply GAs to path planning problem, the path needs to be encoded into chromosome code. The distributed nodes in the space are used to encode the path. Where, the chromosome of each path is represented by a series of integers. Each integer (gene) represents specified node in the space. Any two neighbored nodes are connected by line, which is a segment of the overall path from start point to the target point.

The first node of the chromosome indicates the starting point of the first line segment while the first intermediate node represents the ending of it. The other intermediate nodes construct the intermediate line segments. The last node indicates the end of the last line segment, which should be the target. The paths may contain different number of intermediate nodes. Hence, variable-length chromosome GA is proposed. The aim is to reduce the computation time and to remove the unnecessary nodes.

The chromosome structure must have sufficient information about the entire path. Therefore, an additional information are added to the chromosome; the length of chromosome, a binary flag which gives an indication whether this path reaches the target or not (1 reach and 0 not) and the number of obstacles collisions while the mobile travels through this path from start point to target point. The chromosome structure is shown in Figure (3).

b) Initial Population

Although the initial population is generated randomly, it is submitted to some conditions to ensure the feasibility of the paths. In the whole population, the first integer (gene) and last integer of each chromosome represent the start and target points respectively. The intermediate integers represent the intermediate nodes. Each intermediate node is selected randomly from the available nodes that connect to the previous one. The available connected nodes to each node are defined in the links table. The selection of intermediate nodes will continue until the chromosome reaches a certain length chosen for this current chromosome or the chromosome reaches the target node. Additional condition is introduced; if the target node is one of the available connected nodes, then it is selected directly and stops node generating for the current chromosome.

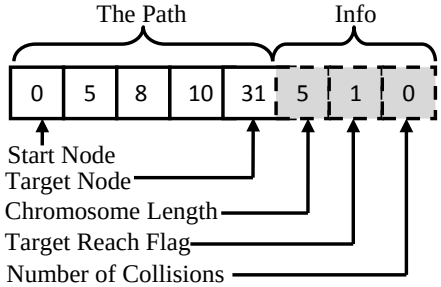


Figure (3): The chromosome structure.

This method ensures that all the initial paths will be free of collision. Moreover, it also increases the possibility of generating a path that begins at the start node and ends at the target node.

c) Trine Classifier

This operator applies to all paths (chromosomes) and classifies them into one of the following three categories;

- *Type-1*: Feasible path begins at the start node and reaches the target as well as has not obstacles collision on the way.
- *Type-2*: Infeasible path begins at the start node and reaches the target, but it has some obstacles collision on the way.
- *Type-3*: Infeasible path do not reach the target and it may has obstacles collisions.

This operator depends on links table, where it checks all nodes of each path sequentially node by node. If there is, a node in a certain path that does not belong to valid connected nodes with its previous node, this path will classify as a type-2, if it reaches the target node or type-3 if it is not.

If the all path links are valid and path reaches the target node, then it is of type-1 path. Figure (4) shows the types of paths.

b) Fitness Function

The fitness function in GAs is used to evaluate the solutions (chromosomes) according to a certain optimization criterion. In our problem, the shortest path between start node and target node is an optimal path. Thus, the fitness of the path is conversely proportional with respect to the length of the path. Therefore, the fitness function is chosen to be one over the lengths summation of the line segments that forms the overall path.

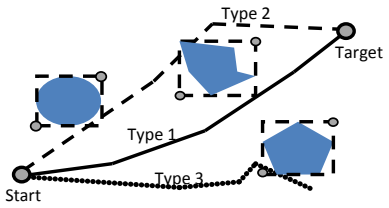


Figure (4): The types of path.

The fitness equation is:

$$F = \frac{1}{\sum_{i=1}^{m} d(N_i, N_{i+1}) + PUNISH} \quad \ldots(2)$$

Where "*N*" is the node, "*m*" is the number of nodes in a chromosome and "*d*" is the distance function, which calculates the distance between two adjacent nodes from their coordinate's values (i.e. using nodes table) as in equation below.

$$d(N_i, N_{i+1}) = \sqrt{(x_{Ni} - x_{Ni+1})^2 + (y_{Ni} - y_{Ni+1})^2} \quad \ldots (3)$$

Punish is the punishments factor for the paths. The feasible paths (type-1) take Punish=0 and for the infeasible paths that reach the target (type-2) take PUNISH= number of path's collisions. The rest of paths that do not reach the target (type-3) take fitness value equals to zero.

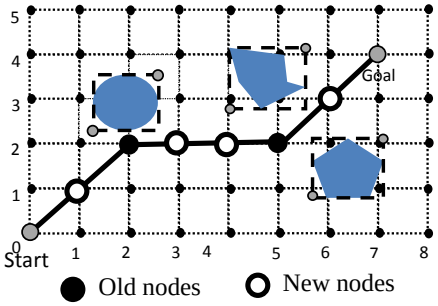
c) Sort Operation

This operator sorts the chromosomes of a population according to their classes. At first, chromosomes of type-1 are sorted in descending order according to their fitness values. Then, type-2 chromosomes are sorted firstly in descending order according to their fitness values. Secondly, if a group of chromosomes has an equal fitness values, they are again sorted, in ascending order according to the number of obstacles collisions (number of invalid links). The rest of chromosomes are of type-3. They are sorted in ascending order according to the number of obstacles collisions.

d) Line Breaker

Each path consists of a set of nodes and relevant line segments that created between any two adjacent nodes, as we mention earlier. The length of the line segment is unspecified. Therefore, it could be very long and may causes decreasing the number of nodes in the path. This situation decreases the ability of chromosome to evolve efficiently because the path cannot alter itself more freely. Moreover, all operations are applied only on those nodes.

Line breaker operator is proposed here to increase the number of nodes in the path. The additional nodes are added to the path that has a line segment exceeds a certain length without change the path length or shape. In other word, this operator will add nodes a long line segment that connects two adjacent nodes as shown in Figure (5).



Old nods path= [0, 20, 23, 43]

New nodes path= [0, 11, 20, 21, 22, 23, 32, 43]

Figure (5): Line breaker operation.

The advantages of this operation are enabling the path to avoid the obstacles collisions between these two nodes and changing the unfeasible path to feasible one.

e) Path Enhancer

This operator removes or changes the nodes that the path passes through them while travels from start node to target node.

The operator searches in the links table about another valid node to replace the node ***i+1*** that shares links with node ***i*** and node ***i+2***. Then, the operator tries to make the path distance between node ***i*** and node ***i+2*** as short as possible.

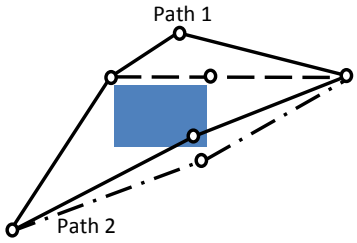


Figure (6): Path enhancer operation.

This operator applies to all the path's nodes except the nodes that have no possibility to be enhanced (the node that lies on the same straight line, which connects its previous and next nodes). Path enhancer is applied only to the new offspring and for almost five generations when the best path is not evolving to reduce the computation cost. The operation of path enhancer may validate the infeasible paths by changing or removing the invalid links. This operation is shown in Figure (6).

f) Selection Method

In applying natural genetics operations (crossover and mutation operations), the parents are selected using tournament selection method. In the tournament selection, a small subset of chromosomes (two or more) is randomly picking from the mating pool (current population). The chromosome of highest fitness in this subset is selected as a parent. The tournament is repeated for every parent needed. Tournament selection makes the selection operation fast because the population never needs to be sorted.

g) Enhanced Crossover

During the crossover operation, two chromosomes that have sufficient fitness values are randomly selected as parents based on the selection method.

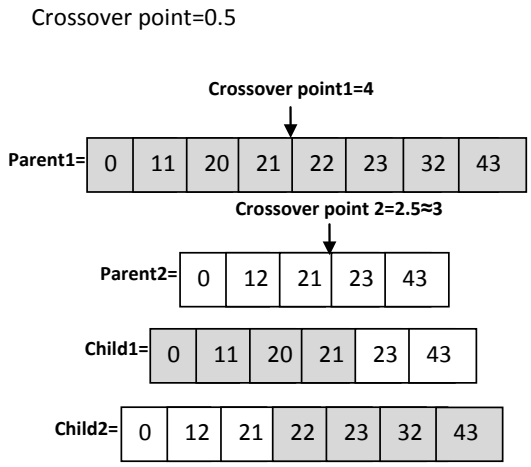


Figure (7): Crossover operation.

In this paper, single point crossover is used. Each parent has its chromosome length. The crossover point is randomly chosen as flow; a number between zero and one is chosen randomly and multiply it by chromosomes length of each parent (proximate the result to the nearest integer if required). For example, if parents one and two have chromosomes lengths of 10 and 8 respectively and the random number is 0.5. The location of crossover point for parent one and two will be 5 and 4 respectively. Figure (7) shows an example of crossover operation.

Crossover operation swaps the two parents around crossover points. This operation results feasible paths, if and only if the node before crossover point in the first parent and the node after crossover point in the second parent and in opposite, have at least

one valid link. If not, the crossover operation will be done randomly. This will result in an infeasible path.

j)Enhanced Mutation Operation.

To ensure that the mutation operation will not create an infeasible path, enhanced mutation operation is proposed. The parental chromosome is chosen randomly according to the used selection method. The parent's start and target nodes are not mutated. The mutation operation is done by selecting an intermediate node in the parent according to mutation probability. Then, count all the nodes that can replace this mutated node (these nodes have also links to previous and next nodes of the mutated node). One of these nodes is chosen randomly to replace the mutated node.

SIMULATION RESULTS

The proposed algorithm is applied on the following three environments spaces using computer with properties of 2.1 GHz Core 2 Due CPU and the software written in C++ programming language.

***Example-1*:** In this example, an environment space of 30×30 which contains 28 fixed obstacles is illustrates in Figure (8) and Figure (9). For this example, the EGA parameters are, population size is 300, the crossover rate is 0.85, the mutation is 0.05, the maximum generation is 5000 and the maximum length of chromosomes is 30.

The proposed algorithm is applied to find two optimal paths. The results are shown in Table (1) and the graphs of these optimal paths are shown in Figure (8) and Figure (9) respectively.

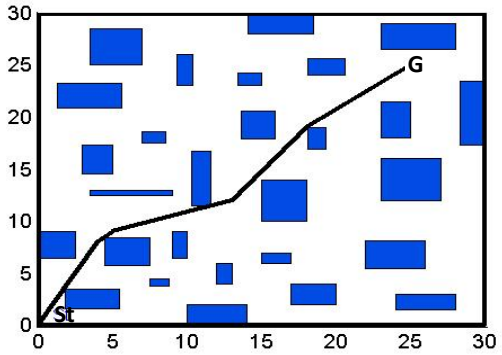
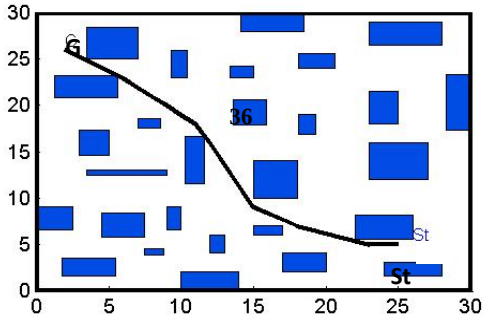


Figure (8): Optimal path of Example 1.A.



Deleted: ¶
<object>¶

	No. of Runs	Start Point	Target Point	Max Generations	Min Generations	Elapsed Time for all runs in sec	Failed Runs	Max Error	Best path length
Example 1.A	100	(0,0)	(25,25)	137	101	15.28	0	0	36.7242
Example 1.B	100	(25,5)	(2,26)	209	105	31.29	0	0	32.9131
Example 2.A	10	(0,0)	(90,52)	177	136	443.52	2	0.05	106.46
Example 2.B	10	(80,10)	(30,40)	144	136	306.229	0	0	61.274
Example 2.C	10	(10,50)	(90,20)	159	126	354.842	3	0.07	87.95
Example 3	10	(1,14)	(14,1)	8	7	1.9	0	0	20.54

The maximum error is calculated by subtracting the optimal path length from each failed run path length and takes the maximum result. Max generation and min

generation is the maximum and minimum number of generations of all the runs of each example that the EGA needs to reach the optimal path.

***Example-2*:** In this example, an environment space of 100×60 which contains 146 fixed obstacles is shown in Figures (10, 11, and 12). For this example, the EGA parameters are; the population size is 800, the crossover rate is 0.86, the mutation rate is 0.05, the maximum generation is 5000, and the maximum length of chromosome is 100. The proposed algorithm is applied to find three optimal paths. The results are shown in Table (1) and the optimal paths are shown in Figure (10), Figure (11) and Figure (12) respectively.

From the results shown in Table (1), we can notice that the proposed algorithm has a very good ability to find different global optimal paths for simple and complex environments spaces with small and huge number of obstacles.

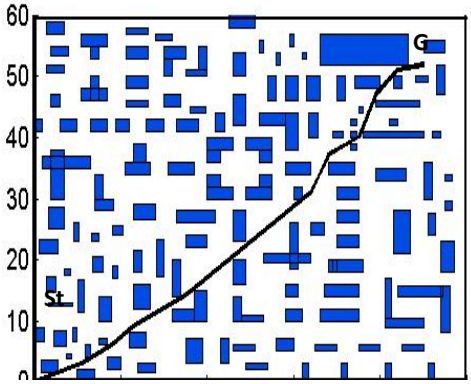


Figure (10): Optimal path of Example 2.A.

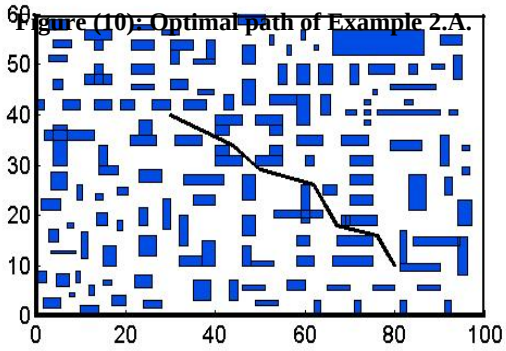


Figure (11): Optimal path of Example 2.B.

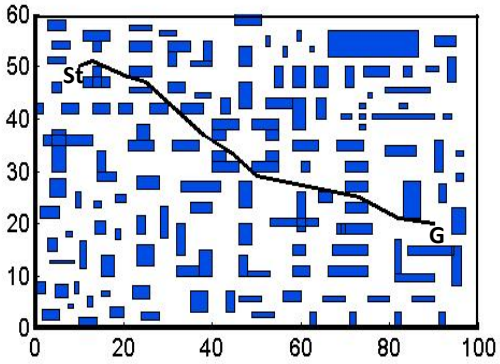


Figure (12): Optimal path of Example 2.C.

***Example-3*:** In this example, a comparison is done between our algorithm and the improved chaotic genetic algorithm (CGA) which is proposed in [11]. The CGA is applied on an environment of 15×15, the results are shown in Figure (13).

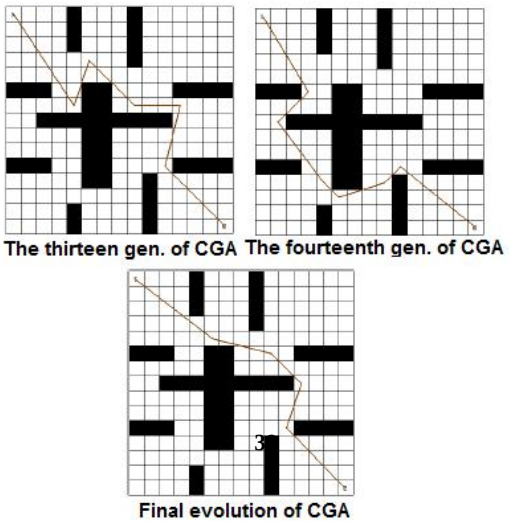


Figure (13): CGA results for 15×15 environment [11].

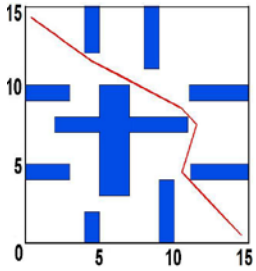


Figure (14): EGA result for 15×15 environment [11].

The proposed EGA in this paper is also applied to the same environment of 15×15. The results show that the CGA find an optimal path with length equal to 22.17 while EGA find an optimal path with length equal to 20.54. In addition, the EGA finds the optimal path with only 7 or 8 generations while the CGA needs more generations to reach its optimal path. Thus, we can say that EGA is better than CGA because it can find shorter path and takes less time to find it.

CONCLUSIONS

We developed optimal path finder algorithm for mobile robot. In fact, this algorithm works in two phases, the off-line phase that is called the data collection part and the online phase that is the EGA part. Data collection part gathers all the information needed from the environment to increase the efficiency of EGA before the searching process begins. Moreover, EGA consists of two types of operations; basic EGA operations or genetic operations and auxiliary operations or non-genetic operations. Auxiliary operations developed to reduce the number of generations needed

to reach the optimum path by making the work of EGA basic operations more effective.

The proposed EGA in this paper is compared with CGA proposed in [11]. The results show that the EGA is better because EGA found shorter path with less generations than CGA. However, the differences between the two algorithms become more effective and obvious when they applied on more complex environment.

Finally, the proposed algorithm proved that is it very powerful, reliable and effective algorithm to find an accurate optimal global path for mobile robot even though the environment space has huge number of obstacles.

REFERENCES

[1]. Kamran H. Sedighi, Kaveh Ashenayi, Theodore W. Manikas, Roger L. Wainwright, Heng-Ming Tai, "Autonomous Local Path Planning for a Mobile Robot Using a Genetic Algorithm", 19-23 June 2004 IEEE On page(s): 1338 - 1345 Vol.2.

[2]. Rahul Kala, Anupam Shukla and Ritu Tiwari, "Fusion of probabilistic A* algorithm and fuzzy inference system for robotic path planning", Springer DOI, Volume 33, Number 4, Pages 307-327, 2010.

[3]. Anthony Stentz, "Optimal and Efficient Path Planning for Partially-Known Environments" Carnegie Mellon University Robotics Institute, In Proceedings IEEE International Conference on Robotics and Automation, May 1994.

[4]. William Donald Smart and Leslie Pack Kiebling, "Effective Reinforcement Learning for Mobile Robots", Proceedings of the 2002. IEEE International Conference on Robotics & Automation, Washington DC, USA, Pages 3404 – 3410, 2002.

[5]. Velappa Ganapathy, Soh Chin Yun, Wen Lik Dennis Lui, "Utilization of Webots and Khepera II as a Platform for Neural Q-Learning Controllers", IEEE Symposium on Industrial Electronics and Applications (ISIEA 2009), Kuala Lumpur, Malaysia, Pages783–788, 4-6 October 2009.

[6]. Tomas Lozano-Perez, "Spatial Planning: A Configuration Space Approach", IEEE Transactions on Computers, Vol. C32, No. 2, Pages 108-120, 1983.

[7]. Elon Rimon, Daniel E. Koditschek, "Exact Robot Navigation Using Artificial Potential Functions", IEEE Transactions on Robotics and Automation, Vol. 8, No. 5, Pages 501-518, 1992.

[8]. Homayoun Seraji and Ayanna Howard, "Behavior-Based Robot Navigation on Challenging Terrain: A Fuzzy Logic Approach", IEEE Transactions on Robotics and Automation, Vol. 18, No. 3, Pages 208 – 321, 2002.

[9]. Gordon DeMuth and Steve Springsteen, "Obstacle Avoidance Using Neural Networks ", Autonomous Underwater Vehicle Technology, Proceedings of the (1990) Symposium, Pages 213 – 215, 1990.

[10]. Soh Chin Yun , Veleppa Ganapathy and Lim Ooi Chong," Improved Genetic Algorithms based Optimum Path Planning for Mobile Robot", 7-10 Dec. 2010 IEEE, On page(s): 1565–1570.

[11]. Gao Ye and Zheng Tao, "Improved genetic algorithms based on chaotic mutation operation and its application", IEEE DOI, 2010.