# Optimal Path Selection for Mobile Robot Navigation Using Genetic Algorithm

**Tamilselvi[1], Mercy shalinie[2], Hariharasudan[3]**

**[1] Department of Computer Science and Engineering
Thiagarajar College of Engineering, Madurai 625 015, Tamilnadu, India**

**[2] Department of Computer Science and Engineering
Thiagarajar College of Engineering, Madurai 625 015, Tamilnadu, India**

**[3] Department of Computer Science and Engineering
Thiagarajar College of Engineering, Madurai 625 015, Tamilnadu, India**

**Abstract**

The proposed Navigation Strategy using GA(Genetic Algorithm) finds an optimal path in the simulated grid environment. GA forces to find a path that is connected to the robot start and target positions via predefined points. Each point in the environmental model is called genome and the path connecting Start and Target is called as Chromosome. According to the problem formulation, the length of the algorithm chromosomes (number of genomes) is dynamic. Moreover every genome is not a simple digit. In this case, every genome represents the nodes in the 2D grid environment. After implementing the cross over and mutation concepts the resultant chromosome (path) is subjected to optimization process which gives the optimal path as a result. The problem faced with is there may be chances for the loss of the fittest chromosome while performing the reproduction operations. The solution is achieved by inducing the concept of elitism thereby maintaining the population richness. The efficiency of the algorithm is analyzed with respect to execution time and path cost to reach the destination. Path planning, collision avoidance and obstacle avoidance are achieved in both static and dynamic environment.

Keywords: *Mobile Robot, Path Planning, Genetic Algorithm, Optimal Path, Navigation*

## 1. Introduction

As long as humankind exists, we strive for perfection in many areas. We want to reach a maximum degree of happiness with the least amount of effort. In our economy, profit and sales must be maximized and costs should be as low as possible. Therefore, optimization is one of the oldest of sciences which even extends into daily life [1]. Global optimization is the branch of applied mathematics and numerical analysis that focuses on, well, optimization. In 1950 – 1960's – Genetic Algorithm – Evolutionary Biologists explicitly search for model aspects of natural evolution[2]. In 1962, researchers G.E.P. Box, G.J. Friedman, W.W. Bledsoe & H.J. Bremermann – independently developed evolution-inspired algorithms for function optimization and machine learning. Early 1962, John Holland's work first to explicitly propose crossover and other recombination operators[3]. In 1965 Ingo Rechenberg from The Technical University of Berlin, introduced *evolution strategy. Without population and crossover,* one parent mutated to produce one offspring, and the better of the two was kept and became the parent for the next round of mutation[4]. 1966, when L.J. Fogel, A.J. Owens and M.J. Walsh introduced in America a technique they called *evolutionary programming*. In this method, solutions to problems were represented as simple finite-state machines; like Rechenberg's evolution strategy, their algorithm worked by randomly mutating one of these simulated machines and keeping the better of the two[8]. In 1975 , with the publication of the book *Adaptation in Natural and Artificial Systems* makes genetic algorithms on a theoretical form by introducing the notion of schemata[5]. GA is reliable and adaptive. So it is successfully used in the function optimization [6][7], job-shop scheduling problem[9;10], and other areas. A good overview of GAs can be found in [8] [9]. Genetic algorithms **(GAS)** are search algorithms and optimization techniques using the principles of natural selection inspired by Darwin's theory about evolution. GA was proposed by professor John H.Holland in Michigan State University[10]. It is a randomized search algorithm based on biological evolution in nature. It includes the steps of roulette, crossover, mutation, and so on. GA is reliable and adaptive. So it is successfully used in the function optimization, job-shop scheduling problem [11] [12] and other areas.

## 2. Mobile Robot Path Planning

Daehee Kang et al.,proposed for path plannnig using GA with two stages. First (named a minor league) checks if a chromosome can reach a goal position or not, and makes the individuals evolve; only a reaching chromosome is then transferred to the second stage (called a major league) and are then evolved. Finally, the best chromosome of individuals in the second stage survives, so that the optimal/shortest path is generated[13]. Jianping Tu proposed the chromosome has a variable length. The location target and obstacles are included to find a p ath with optimality achieved for short distance[14]. Sedighi, K.H. et al., path-planning algorithm for local obstacle avoidance (local feasible path) of a m obile robot in a given search space. The method tries to find not only a valid path but also an optimal one [15]. Ismail AL-Taharwa proposes the optimal path selection for mobile robot in which fitness function was simplified, utilizes it for path length[16]. Ashwin et al proposed the reactive control behavior using Genetic algorithm with floating point gene representation[17].

Yong Zhang et al proposed in an unknown environment are proposed with grid model for path selection. Using digital potential field generated initial path population, and its optimization find the shortest path, and individual evaluation function were processed fitness function both feasible path and unfeasible path fitness function, and then by increasing the deleted and inserted operators to meet the requirement of avoiding obstacles in the path planning[18].Shijie Dai et al proposed rough set genetic algorithm (RSGA) to o ptimize the robot path planning speed and enhance the precision in complicated grid environments. In th e grid model the initial decision-making table of the robot is obtained, which can be simplified by the rough set theory to extract the minimal decision-making rules. These rules to train the initial population of the genetic algorithm (GA), and then solve the best path using GA[19]. Chen Ye & Webb proposed a reactive-based approach for navigation in complex and unknown environments called sub goal seeking. The depth point maps of the environment are analysed to extract free spaces around the robot. These spaces are then evaluated the one that is most likely to lead to the final goal is chosen as a sub goal. The robot then drives towards these sub goals, instead of the final goal until it is visible [20]. LIU Changa paper searches the global optimal path in the static obstacle environment using improved visual graph, then it searches the local optimal path in the moving obstacle environment using improved GA[21]. The proposed methodology was inspired from Jianping Tu et al in which the chromosome of v ariable length was chosen[14].

## 3. Genetic Algorithm Approach

The design of the environment is to interact with users (learners, developers). It provides a u ser friendly environment for the users to learn to implement it. Fig.1. shows the steps involved in the process of optimal path planning.

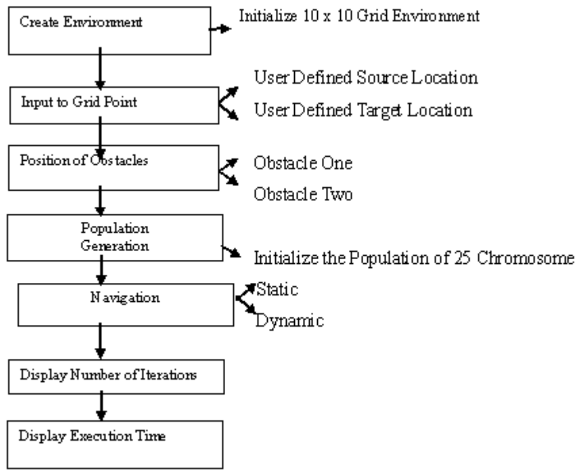

Create Environment
Initialize 10 x 10 Grid Environment
Input to Grid Point
User Defined Source Location
User Defined Target Location
Position of Obstacles
Obstacle One
Obstacle Two
Population Generation
Initialize the Population of 25 Chromosome
Navigation
Static
Dynamic
Display Number of Iterations
Display Execution Time

**Fig.1. Steps for Optimal Path Selection**

First is the generation of initial population of chromosomes.The fitness of each chromosome is evaluated.The fitness function is modified such that it avoids the collision in its path,smoothness in path.Roulette wheel selection method is used in this algorithm.Attimes we may lose the best fit chromosome in the selection process.So we propose a t echnique called as Elitism. Elitism is used to keep track of the best fit chromosome obtained during the process.So that the best fit chromosome is present in the forthcoming generations.Then the process of single point crossover and mutation is done.Thus a new population of chromosome for the next generation is obtained.For the new set of population fitness is evaluated and crossover ,mutation is performed.This process is repeated for “n” g enerations .Finally the output depicts the most shortest optimal path in less no of generations. Thus the modification applied to the existing genetic algorithm will yield the optimal path with reduced no of generations.

## 4.Genetic Algorithm for Optimal Path

The problem to select or choose an optinial/minimal length path is one of optimization problems which have constraints. The typically used path generation techniques desire very large computing power. In here, we propose one method using genetic algorithm. Genetic Algorithms are general purpose, parallel search procedures that are based upon genetic and evolutionary principles. A genetic algorithm works by repeatedly modifying a population of artificial structures through the application of genetic operators. GAS are typically black-box methods that use fitness information exclusively, that is, they are not require gradient information or other internal knowledge of the problem. The goal in optimization is to find the best possible solution or solutions to a problem, with respect to one or more criteria. In order to use a genetic algorithm, one must first choose a suitable structure for representing those solutions. In the terminology of state space search, an instance of this data structure represents **it** state or point in the search space of all possible solutions. **X** genetic algorithm's data structure consists of one or more *chrommomes* (usually one). A chromosome is typically a string of bits, so the term *strzng* is often used instead. Each chromosome (string) is a concatenation of a number of subcoiriponents called *genes.*

Genes occur at various positions or *locz* of the chromosome,and take on values called *alleles.* The biological term *genotgpe* refers to the overall genetic makeup of an individual. and In our case, a gene is a bit, a locus is its position in the string, an allele is its value (natural number), and length of a chromosome is variable.This section first define a gene and a fitness, and next presents a genetic operator.

### 4.1 Determination of Gene

All living organisms consist of cells. In each cell there is the same set of c**hromosomes.** Chromosomes are strings of DNA and serves as a model for the whole organism. A chromosome consists of **genes**, blocks of DNA. Each gene encodes a p articular protein. Basically can be said, that each gene encodes a **trait**, for example color of eyes. Possible settings for a trait (e.g. blue, brown) are called **alleles**. Each gene has its own position in the chromosome. This position is called **locus**. Complete set of genetic material (all chromosomes) is called **genome**. Particular set of genes in genome is called **genotype**. The genotype is with later development after birth base for the organism's **phenotype**, its physical and mental characteristics, such as eye color, intelligence etc.

Algorithm is started with a **set of solutions** (represented by **chromosomes**) called **population**. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (**offspring**) are selected according to their fitness - the more suitable they are the more chances they have to reproduce

### 4.2 Search Space

The space of all feasible solutions (it means objects among those the desired solution is) is called **search space** (also state space). Each point in the search space represents one feasible solution. Each feasible solution can be "marked" by its value or fitness for the problem. The looking for a solution is then equal to a looking for some extreme (minimum or maximum) in the search space. The problem is that the search can be very complicated. One does not know where to look for the solution and where to start. There are many methods, how to find some **suitable solution** (that is not necessarily the **best solution**), for example **hill climbing**, **tabu search**, **simulated annealing** and **genetic algorithm**. The solution found by this method is often considered as a good solution, because it is not often possible to prove what the real optimum is.

### 4.3 Reproduction

During reproduction, first occurs **recombination** (or **crossover**). Genes from parents form in some way the whole new chromosome. The new created offspring can then be mutated. **Mutation** means, that the elements of DNA are a bit changed. These changes are mainly caused by errors in copying genes from parents. The **fitness** of an organism is measured by success of the organism in its life.

### 4.4 Selection Process

Selection is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (recombination or crossover). There are several generic selection algorithms, such as tournament selection and fitness proportionate selection (also known as *roulette-wheel selection*). To selectively reproduce the population to determine the next generation we use a biased random selection procedure based on fitness. This is implemented using a roulette wheel method. A n imaginary roulette wheel is constructed with a segment for each individual in the population. The size of the segment is based on the aptness of the particular individual. A fit individual will occupy a larger slice of the roulette wheel than a weaker one. Selection is made by rotating the wheel a number of times equal to the population size. When the wheel stops the individual it p oints to is selected. This means that fitter individuals will tend to be selected more frequently than weaker ones. Other algorithms select from a restricted pool where only a cer tain percentage of the individuals are allowed, based on fitness value.

### 4.5 Crossover

Just relying on selection, a population would eventually consist only of the fittest individuals from the initial random population. To better explore potential solutions GAs employ a crossover operator, which mimics the reproduction process. Two individuals are mated by an exchange of these characteristics, and two child chromosomes are created. In the simplest form two individuals are selected for mating using a probability factor $P_x$ (0.7 say). A randomly selected set of patterns from the parents are then exchanged to form two child chromosomes

### 4.6 Mutation

A further enhancement to allow chromosomes to explore all aspects of a s earch space is mutation. The previous operations help to reproduce and improve the level of fitness of a population; however the general form of the characteristics will remain unchanged. The concept of mutation also mimics nature in that very occasionally (with $P_m$=0.001 say) a *wild-card* is played, which causes an upset to the schematic order. Stochastically, this helps to ensure that all possible solutions are potentially investigated. Typically mutation will involve swapping two randomly selected bits in a ch romosome, i.e. $\mathbf{1110100000_2}$ will become $\mathbf{1100100100_2}$ if the 3rd and 8th bits are selected.

### 4.7 Fitness

We need to ascertain some form of fitness that can be associated with a particular individual. With simple GAs this can just be the value output from a function or a computer model of the system we are trying to optimize. For more complex problems involving multi-criteria, a fitness measure associated with solution dominance can be used.

## 5. Implementation Strategy

A chromosome is the complete path from the robot start position S to the target position T. A genome is a point (vertex of the obstacles in the modeled environment), which the path (chromosome) could be completely made by connecting them sequentially. According to the problem formulation, the length of the algorithm chromosomes (number of genomes) is dynamic. A fixed-length chromosome is not suitable for a complex environment, especially, in a dynamic environment. In order to reach the target, in a more obstacles environment, longer chromosome may be needed which means a fixed-length chromosome may not be enough. Therefore, **variable**-length chromosome is better suitable in a dynamic and many obstacles environment. In the proposed model, a novel approach with a v ariable-length chromosome is proposed. Using this proposed method, a mobile robot can find the shortest path in a static environment and the near-optimal obstacle-free path in a dynamic complex environment. A simple chromosome and its corresponding genomes are shown in Fig 2

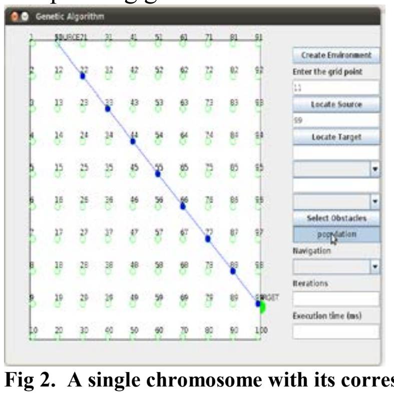


**Fig 2. A single chromosome with its corresponding genes**

The algorithm parameters and concepts are implemented as follows:

1. Generating initial Population.
2. Defining the Optimization process.
3. Defining the fitness Function..
4. Choosing the selection method
5. Defining the crossover operator.
6. Defining the mutation operator.
7. Choosing the algorithm termination condition

### 5.1 .To generate Initial Population

To generate the initial population, the following steps are implemented.

1. Initializing a 10 *10 Grid Environment with obstacles located.
2.Generation of random numbers between the prescribed boundaries i.e.) our source and destination.
3. Enriching the population by filtering it with respect to the ones index as well as value.
4. Let k = source grid point. Finding a nearer point for $a_k$ and concatenate as genome.
5. The selection of next genome is optimized using Euclidean distance.
6. Finding a nearer point for $a_k$ and concatenate as genome. Repeat till T is obtained
7. Repeat till T is obtained L et k = k + 1
8. Repeated steps 4 through 6 until the target is reached.

The above listed 8 steps is repeated for around n times equal to the population size finally resulting in our initial population.

### 5.2 . Defining Optimization Process

Optimization is enhanced in the generation of chromosomes by adopting Euclidean distance method in determining the next genome. Once the source is located, all neighboring grid points are found and it computes the distance between the target and the neighboring grid points using Euclidean distance formulae. The grid point whose distance between the target and itself is shortest is selected as the next genome. This process is repeated until the target is reached. The entire phenomena are repeated for "n" times equal to the population size. Euclidean distance formulae

$$((x_2-x_1)^2+(y_2-y_1)^2)^{1/2}$$

### 5.3. To evaluate the Fitness

Another of the more challenging aspects of using genetic algorithms is to design a fitness function. It a lso called objective function. In this 2-D workplace, a damping coefficient to each grid is assigned. If the grid has an obstacle, its damping coefficient is set to 300; if the grid is free, its damping coefficient is set to 0; if the grid is a target, its damping coefficient is set to -0.9. The fitness function is designed as

$$f=\sum_k d_j\,(1+w_j)$$

Where

$d_j$, is the distance between two adjacent genes,
$w_j$, is the damping coefficient of jth gene
K is the total grids a mobile robot will move over

d,= { 1 , if horizontal or vertical direction,
0.414 .if diagonal}

### 5.4. Selection Method

Two chromosomes are selected from the initial population. The chance of being selected is proportional to the chromosomes fitness. Selection method used here is roulette wheel selection. In this method, it d oes not guarantee that the fittest member goes through to the next generation merely that it has a very good chance of doing so. Selection method is performed in steps as listed below

1. Random number is generated
2. Fitness values are cumulatively added such that the summed fitness value exceeds the random number
3. The chromosome corresponding to the last fitness value suimmed is the chromosome to be selected.
4. Repeat the steps 1 -3 to select the next chromosome to perform reproduction operations.

**5.5. Defining the Elitism**

The steps involved in incorporating the concept of elitism in Genetic Algorithm is as follows

1. Generate an initial population P of size M and calculate the fitness of each string S of P.
2. Find the best string Sbest of P. If the best strings are not unique, then call anyone
   Of the best string in P as Sbest
3. Construct the mating pool. Perform crossover and mutation operation on the strings of the mating pool and obtain a population Ptmp.
4. Compare the fitness of each string S of Ptmp with Sbest.
5. Replace the worst string of Ptmp with Sbest if the fitness of each string in Ptmp is less than the fitness of best Otherwise no replacement takes place in Ptmp. Rename Ptmp as P.
6. Go to step 3.

This process is done to maintain the consistency in population richness

**5.6. Defining the Crossover Concept**

Once the chromosomes are selected by employing the above selection process, a random number is generated to determine the position where the crossover is to be performed. The crossover techniques used here is single point crossover and the crossover probability is 0.9.Two selected chromosomes are referred to as the parents. Children are generated by concatenating the first half of the first parent and second half of the second parent and vice versa. Thereby two off springs are generated. After performing crossover, the offspring produced forms the new generation.

**5.7. Defining the Mutation operator**

The next reproduction operation to be performed is mutation where a particular gene in the chromosome gets changed. Mutation probability is 0.02.As in the crossover a random number is generated to determine the position where the mutation has to be performed.

**6. Genetic Algorithm under various Constraints**

**6.1. Working of GA in Obstacle free environment**

The source and destination location is obtained from the user and the corresponding nodes are located in the 2D grid environment. The initial population which is the possible paths between the given source and destination is generated and displayed in the environment. For each chromosome the fitness function is evaluated and the fitness values are sorted in ascending order. Then each time two chromosomes from the initial population are selected as parents and crossover is performed to produce two off springs. This process is repeated for all the chromosome in the initial population .Then mutation is performed for each chromosome. Thereby, we obtain new population of chromosome for the next generation. .For the next generation, fitness is computed and also crossover, mutation process is done the process is repeated for “n” generations .Finally, the shortest optimal path is obtained for the user defined source and user defined goal.

**6.2. Working of GA in Obstacle (Static environment)**

The source and destination location is obtained from the user and the corresponding nodes are located in the 2D grid environment. The obstacle location is obtained for the user and the location is painted in the 2D grid environment. While generating the initial population (which is the possible paths between the given source and destination), we detect the presence of obstacle .For each chromosome the fitness function is evaluated and the fitness values are sorted in ascending order. The path with the obstacle will have the highest fitness value. Then each time two chromosomes from the initial population are selected as parents and crossover is performed to produce two off springs. This process is repeated for all the chromosomes in the initial population .Then mutation are performed for each chromosome. Thereby, we obtain new population of chromosome for the next generation. For the next generation, fitness is computed and also crossover, mutation process is done. The path with has the highest fitness value gets eliminated in the next forthcoming generation. The process is repeated for “n” generation’s .Finally, the shortest optimal path is obtained for the user defined source and user defined goal in the presence of static obstacle.

**6.3. Working of GA in Obstacle (Dynamic environment)**

The source and destination location is obtained from the user and the corresponding nodes are located in the 2D grid environment. The obstacle location is obtained for the user and the location is painted in the 2D grid environment. While generating the initial population (which is the possible paths between the given source and destination), we detect the presence of obstacle .For each chromosome the fitness function is evaluated and the fitness values are sorted in ascending order. The path with the obstacle will have the highest fitness value. Then each time two chromosomes from the initial population are selected as parents and crossover is performed to produce two off springs. This process is repeated for all the chromosomes in the initial population .Then mutation are performed for each chromosome. Thereby, we obtain new population of chromosome for the next generation. For the next generation, fitness is computed and also crossover, mutation process is done. The path with has the highest fitness value gets eliminated in the next forthcoming generation. After each five generations, the obstacles will move a single grid cell. The process is repeated for "n" generation's .Finally, the shortest optimal path is obtained for the user defined source and user defined goals in the presence of dynamic moment of obstacles.

## 7. Experimental Results – Simulation

Fig.3. shows that the user has given the source location as 1 and the target location as 100.Those user defined source and target is painted in the grid environment with source in Blue Colour and Target in Green Colour. Various possible path to reach the target from the source in the grid environment were shown. The initial population is generated in the absence of obstacles.

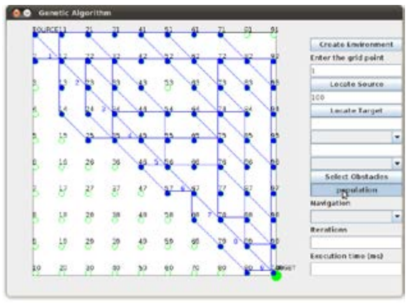


**Fig.3. Initial Population for Obstacle free environment**

Fig.4 shows the most shortest optimal path for the robot to reach the defined target .The optimal path is obtained by calculating fitness for the initial population. Then after performing crossover and mutation ,the optimal path is obtained. Fig.5 shows the grid environment with the navigation in a static environment. In the grid environment ,the user defined source location, target location ,obstacle location are painted in the grid environment.

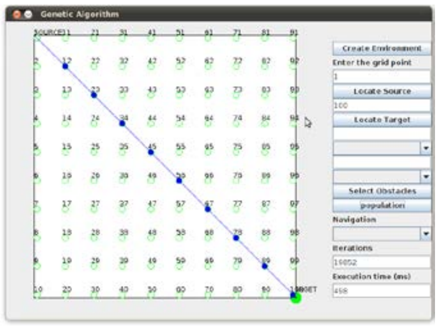


**Fig.4 Optimal Path**

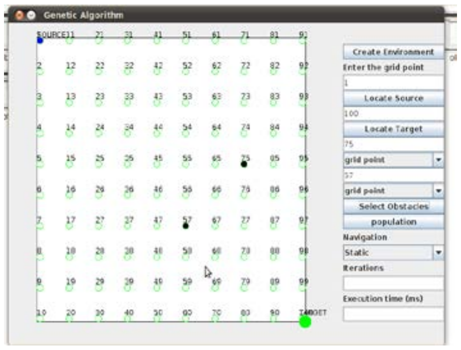


**Fig.5 Static Obstacle environment**

Fig.6 shows the various possible path to reach the target from the source in the grid environment. The initial population is generated by detecting the presence of static obstacles. Fig.7 shows the most shortest optimal path for the robot to reach the defined target in the presence of static obstacle .While generating the initial population ,the obstacle locations are detected.The optimal path is obtained by calculating fitness for the initial population. Then after performing crossover and mutation ,the optimal path is obtained.

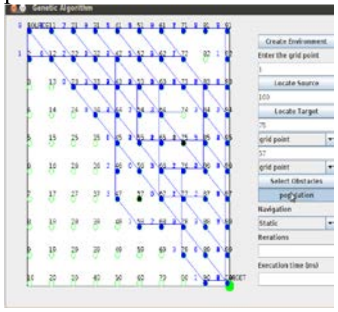


**Fig.6 Initial Population in Static Obstacle environment**

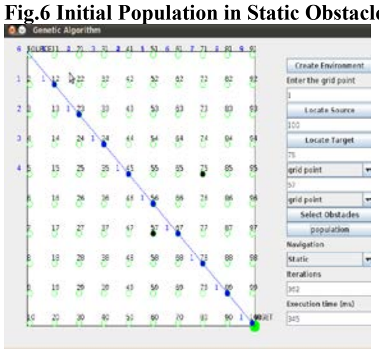
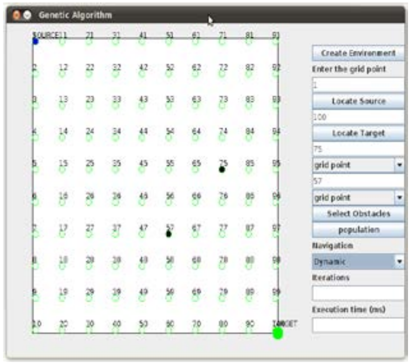


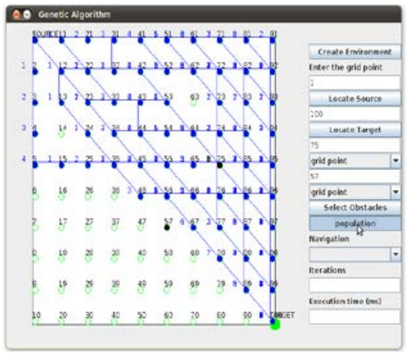
**Fig.7 Optimal Path for Static obstacle environment**

Fig.8 shows the grid environment with the navigation type as Dynamic .In the grid environment ,the user defined source location, target location ,obstacle location are painted in the grid environment. Fig.9 shows the various possible paths to reach the target from the source in the grid environment. The initial population is generated by detecting the presence of Dynamic obstacles.



Enter the grid point: 1
100
75
grid point: 57
grid point
Navigation: Dynamic
Iterations
Execution time (ms)

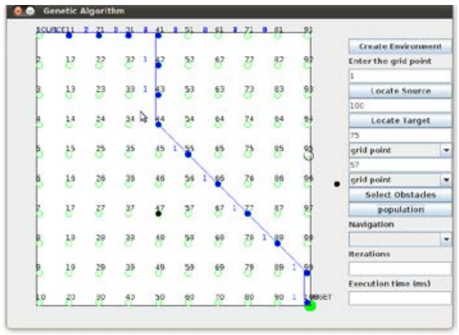
**Fig.8 Dynamic Obstacle Environment**



Enter the grid point: 1
100
75
grid point: 57
grid point
Navigation
Iterations
Execution time (ms)

**Fig.9 Initial Population in Dynamic Obstacle Environment**

Fig.10 shows the most shortest optimal path for the robot to reach the defined target in the presence of dynamic obstacle .While generating the initial population, the obstacle locations are detected. The optimal path is obtained by calculating fitness for the initial population. Initially the obstacle were int the positions 75,57. Then after performing crossover and mutation ,the p ath is obtained. After every five generation the obstacle moves a single grid in the environment. The obstacles is moved to the location 47,outside the grid. Finally the optimal path is obtained.



Enter the grid point: 1
100
75
grid point: 57
grid point
Navigation
Iterations
Execution time (ms)

**Fig.10 Optimal Path for Dynamic obstacle environment**

## 8. Conclusion & Future Work

We have proposed genetic algorithm for path planning of a mobile robot. The genetic algorithm incorporates the domain knowledge into its problem specific genetic operators. The developed GA also features its efficient evaluation method that is greatly beneficial for evolving good solutions from infeasible solutions. The most important modification applied to the algorithm is the fitness evaluation. Due to the modification, the optimal path is obtained in the 50 generations where the optimal path is obtained in 283 generation.

In the future work, it can be extended to have other types of environment like maze-type, grid points etc. Enhancements in terms of obstacles like including the different shapes of obstacles. The different shapes like square, rectangle, upward U, inverted U, upward V, and inverted V can be included in the environment. These different shapes obstacle is used to detect the collision free path where the dynamic obstacle cannot collide with each other. The dynamic obstacles with different shapes will be embedded with the capability to sense along the robots. Thus one of the two or both can stop to avoid collision. Determining the collision free path in the dynamic environment with different shapes will be the future work. This project can also be implemented in the Fuzzy environment. The future step is to implement the algorithm like Genetic algorithm in the hardware named Fire bird kit. Thereby, we will be able to make software stimulation in the tool into real time by making the grid as environment. By burning the algorithm in firebird kit using embedded c as language ,we will be able to move the firebird kit by the logic of the corresponding algorithms.

## 9. Acknowledgments

This work i s a p art of Minor Research Grant Project Sponsored by Computer Society of India titled" Intelligent Algorithmic Approach for Mobile Navigation in an Indoor Environment" , File No. 1-14/2010-03 dated 29.03.2010. Corresponding Author is the Project Invigilator for the above sponsored project.

## References

[1] Arnold Neumaier. Global optimization and constraint satisfaction. In I. Bomze, I. Emiris, Arnold Neumaier, and L. Wolsey, editors, Proceedings of GICOLAG workshop, 2006. Online available at http://www.mat.univie.ac.at/~neum/glopt.html

[2] Mitchell, Melanie," An Introduction to Genetic Algorithms" MIT Press, 1996

[3] Holland, John. "Genetic algorithms." Scientific American, July 1992, p. 66-72.

[4] Haupt, Randy and Sue Ellen Haupt. Practical Genetic Algorithms. John Wiley & Sons, pp.146-147, 1998.

[5] John Holland's ,"Adaptation in Natural and Artificial Systems" MIT Press,1992

[6] Fanlin Meng, Shunxiang Wu," Research of genetic algorithm in function optimization based on HCI", Proc. of IEEE International Symposium on IT in Medicine and Education, Xiamen, China, pp.1049-1052, 2008

[7] Nedim Tutkun, “Optimization of multimodal continuous functions using a new crossover for the real-coded genetic algorithms", Expert Systems with Applications, Vol.36, No.4,pp.8172{8177, 2009

[8] David E.Goldberg, “ Genetic Algorithms in Search, Optimizations and Machine Learning” Addison-Wesley Longman Publishers, ISBN: 0201157675, 1989

[9] DeJong, K. A. (1975). An Analysis of the Behavior of a class of Genetic Adaptive Systems. PhD thesis, University of Michigan, Ann Arbour. Department of Computer and Communication Sciences.

[10] J. H. Holland, "Adaptation in Natural and Artificial Systems", Ann Arbor, MI, The University of Michigan Press, 1975

[11] F. Pezzella, G. Morganti, G. Ciaschetti, “A genetic algorithm for the flexible job-shop scheduling problem", Computers and Operations Research, Vol.35, No.10, p.3202-3212, 2008

[12] Jason Chaohsien Pan, Hanchiang Huang, A hybrid genetic algorithm for no-wait job shop scheduling problems", Expert Systems with Applications, Vol.36, No.3, pp.5800-5806, 2009.

[13] Daehee Kang, Hashimoto, H. Harashima, F.,” Path generation for mobile robot navigation using genetic algorithm” 21st International Conference on Industrial Electronics, Control, and Instrumentation VO.1.pp 167- 172,1995

[14] Jianping Tu, Yang, S.X.,” Genetic algorithm based path planning for a mobile robot” IEEE International Conference on Robotics and Automation,Vol.1.pp.1221-1226, 2003

[15] Sedighi, K.H., Ashenayi, K., Manikas, T.W.,Wainwright, R.L., Heng-Ming Tai,” Autonomous local path planning for a mobile robot using a genetic algorithm” Congress on Evolutionary Computation,Vol.2, pp.1338-1452, 2004

[16] Ismail AL-Taharwa, Alaa Sheta and Mohammed Al-Weshah,” A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment” Journal of Computer Science 4 (4): 341-344, 2008

[17] Ashwin, Arkin, “Using Genetic Algorithm to learn Reactive Control Parameters for Autonomous Mobile Robot Navigation” Adaptive Behaviour, Vol.2, Issue 3, pp.277-304,1994

[18] Yong Zhang, Lin Zhang, Xiaohua Zhang, "Mobile Robot Path Planning Base on the Hybrid Genetic Algorithm in Unknown Environment," isda, vol. 2, pp.661-665, 2008 Eighth International Conference on Intelligent Systems Design and Applications, 2008

[19] Shijie Dai, He Huang, Fang Wu, Shumei Xiao, Ting Zhang, "Path Planning for Mobile Robot Based on Rough Set Genetic Algorithm," icinis, pp.278-281, 2009 Second International Conference on Intelligent Networks and Intelligent Systems, 2009

[20] C. Ye and P. Webb, “A Sub Goal Seeking Approach for Reactive Navigation in Complex Unknown Environments”, Robotics and Autonomous Systems, Vol. 57, Issue 9, 30 September 2009, pp 877-888.

[21] LIU Changan, YAN Xiaohu, LIU Chunyang and LI Guodong, “Dynamic Path Planning for Mobile Robot Based on Improved Genetic Algorithm” Chinese Journal of Electronics, Vol.19, No.2,pp.2010-2014, Apr. 2010

**D.Tamilselvi** received B.E., from Thiagarajar College of Engineering, Madurai, M.E., in National Engineering College. Currently Pursuing Research in AI based Mobile Robot Navigation. She is working as Assistant Professor in Thiagarajar College of Engineering. She is the Project coordinator for the CSI project grant and this work is the part of the project. She published the book chapter in Intech “Advances in Robot Navigation” ISBN: 978-953-307-346-0,2011. Her research areas include Artificial Intelligence, Agents, Mobile Robot Path Planning.

**Dr.S.Mercy Shalinie** is a Professor and Head of the Department of Computer Science and Engineering, Thiagarajar College of Engineering, Madurai. She has published more than 30 papers in both national and international Journals.. She published the book chapter in Intech “Advances in Robot Navigation” ISBN: 978-953-307-346-0,2011. Her research areas include Artificial Intelligence, Neural Networks, Machine Learning and Image Processing etc

**M.Hariharasudan** is a Final Year Bachelor Student of Computer Science and Engineering Department. He presented his research work in All India Seminar conducted by IE for the innovations. He published the book chapter in Intech “Advances in Robot Navigation” ISBN: 978-953-307-346-0,2011. His research areas include Artificial Intelligence, Mobile Robot Navigation.