
Mobile Robot Path Planning in Static Environment

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Bachelor of Technology in Computer Science & Engineering

Submitted by:

Raman Mishra

Roll no :111CS0606

Under Supervision of:

Prof. Ratnakar Dash

Department of Computer

Science & Engineering

NIT Rourkela



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA



National Institute of Technology Rourkela

Certificate

This is to certify that the thesis entitled Mobile Robot Path Planning in Static Environment submitted by Raman Mishra(111CS0606) in partial fulfilment of the requirements for the award of Bachelor of Technology Degree in Computer Science & Engineering at National Institute of Technology, Rourkela is an authentic work carried out by him under my supervision and guidance.

Prof. Ratnakar Dash
Department of Computer Science & Engineering
National Institute of Technology
Rourkela

Acknowledgement

I owe my sincere gratitude to Prof. Ratnakar Dash who guided me during the course of the project. His support and guidance was very crucial for the project. His suggestions will always be remembered which will, I believe, guide me through all walks of life.

Raman Mishra
111CS0606

Abstarct

The success of Particle Swarm Optimization (PSO) and Genetic algorithm (GA) as single objective optimizer has motivated researchers to extend the use of this bio-inspired techniques to other areas. One of them is multi-objective optimization. As a part of this review we present a classification of the approaches and identify the main approaches here. We describe useful performance measures and simulation results of conventional Genetic algorithm and PSO. We extend this to multi-objective genetic algorithm and PSO. This means that GA and PSO optimizes path based on two criteria: length and difficult. Another method that has new to this field of research is the Artificial Potential field method. In this method the entire space is supposed to contain a potential field and we calculate the net force that is acted upon the robot to reach its goal.

Keywords- *Multiple Objective Optimization, Particle Swarm Optimization, Genetic Algorithm*

Contents

Certificate	i
Acknowledgement	ii
Abstract	iii
List of Figures	v
1 Introduction	1
2 Literature Review	3
3 Proposed Work	4
3.1 A Star	5
3.1.1 A star use of heuristic	5
3.1.2 Simulation Results	6
3.2 Genetic Algorithm	8
3.2.1 Proposed Method for Path Planning	8
3.2.2 Initial Population	9
3.2.3 Path Repair Mechanism	9
3.2.4 Single Objective Function	9
3.2.5 Two Objective Function	10
3.2.6 Genetic Operator	11
3.2.7 Selection Method	11
3.2.8 Simulation Results	12
3.3 Particle Swarm Optimization	14
3.3.1 Simulation Results	15
3.4 Artificial Potential Field Method	16
3.4.1 The Attractive Potential Function	16
3.4.2 The Repulsive Potential Function	16
3.4.3 Simulation Results	17
4 Conclusion	18
References	19

List of Figures

1	Without Scanning	1
2	Path after Scanning	2
3	Path Planning	5
4	map1 Astar	6
5	map2 Astar	7
6	map1 Astar	7
7	Path Repair algorithm	10
8	Algorithm of multi-objective GA	11
9	map 1 GA	12
10	map 2 GA	12
11	map 3 GA	13
12	map 4 GA	13
13	map 1 PSO	15
14	map 1 APF	17
15	map 2 APF	17

1 Introduction

Path Planning for autonomous robots is a very challenging task. We often face a situation in real life where the robot is supposed to find the best possible paths in an environment which is full of obstacles. Some path finding algorithms that have been used to find the best possible path are Genetic, Particle Swarm, etc.

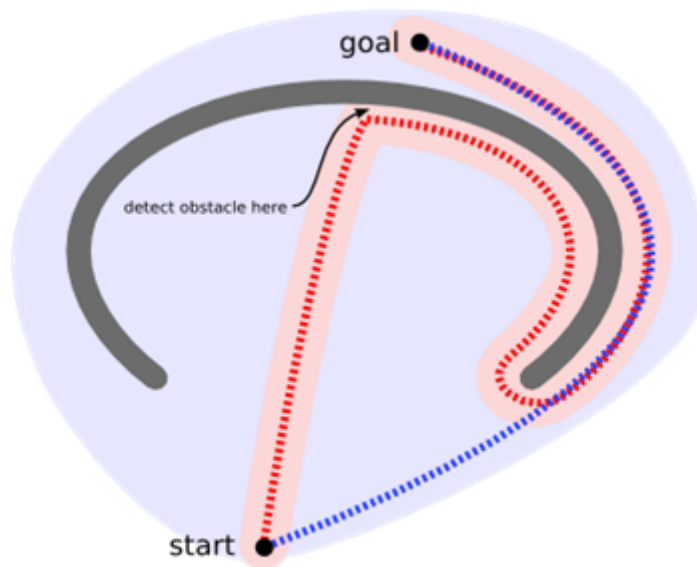


Figure 1: Without Scanning

As you can see in the above figure a unit which is at the bottom of the map want to reach the goal. Instead of scanning the area completely it keeps on moving without until it reaches a point where it realizes that it can no longer move forward. Though the goal point is within its grasp it is not able to reach the goal. So now the robot tries to find an alternate path where it first tries to come out of that concave shaped obstacle and then moves along the obstacle and then reaches its goal.

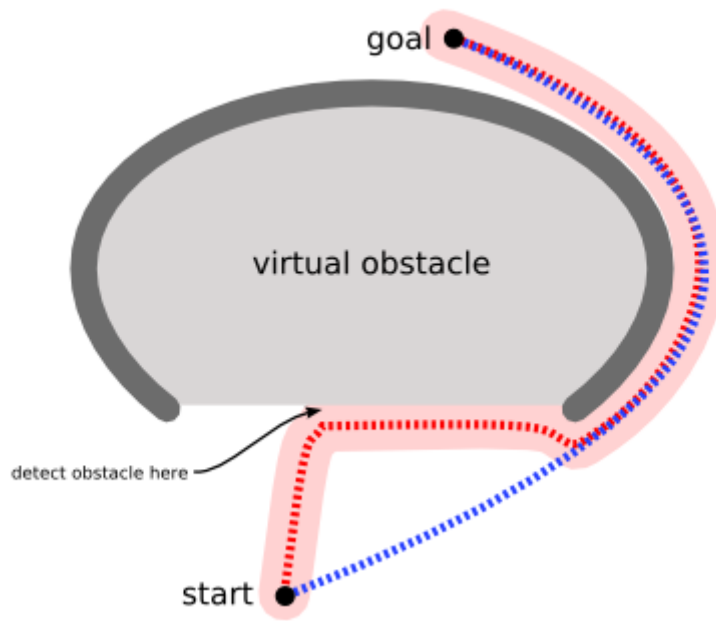


Figure 2: Path after Scanning

Now in this figure you can see the bot no longer enters into the trap. In this situation the bot has planned its path before scanning the whole area making it possible to avoid the trap. Though planning takes time but assures a safe path for the robot to travel. This type of scanning of the map is called path planning and this is done using path planning algorithms.

2 Literature Review

The first proposal of multi-objective particle swarm optimization and multi-objective Genetic Algorithm is fourteen years old but considerable algorithm have been proposed since then. Most researches that have taken place on this topic have tried to exploit this technique to solve the twin objective of finding a robust path while avoiding obstacles. In this project I have tried to implement the same and try to compare the results of Genetic algorithm with that Particle swarm Optimization. Artificial Potential Field method is also used for path planning which is an online approach. This method has been proposed in a lot of papers with different Potential functions. This method is also used in dynamic environment along with other algorithms.

3 Proposed Work

I have implemented four path finding algorithms in static environment using matlab. I have taken a map containing obstacles (black boxes, red circles) and by using the data provided i.e. source point and destination point I am able to create an optimal or sub-optimal path avoiding obstacles. The algorithms that I have implemented are as follows :-

1. A star Algorithm
2. Genetic Algorithm
3. Particle Swarm Optimization
4. Artificial Potential Field Method

3.1 A Star

A star algorithm takes the best of features from both Dijkstras Algorithm and Greedy Algorithm. Dijkstras algorithm tends to select those points which are nearer to the starting point and then it will move outwards from there. Though it is guaranteed to find the shortest path but computational complexity is very high making it difficult to use in path finding. Greedy algorithm works in a similar manner but differs in the way it chooses its points. This algorithms selects those points which are nearer to the goal than the starting point. Though this algorithm may not find the shortest path but still works faster than Greedy.

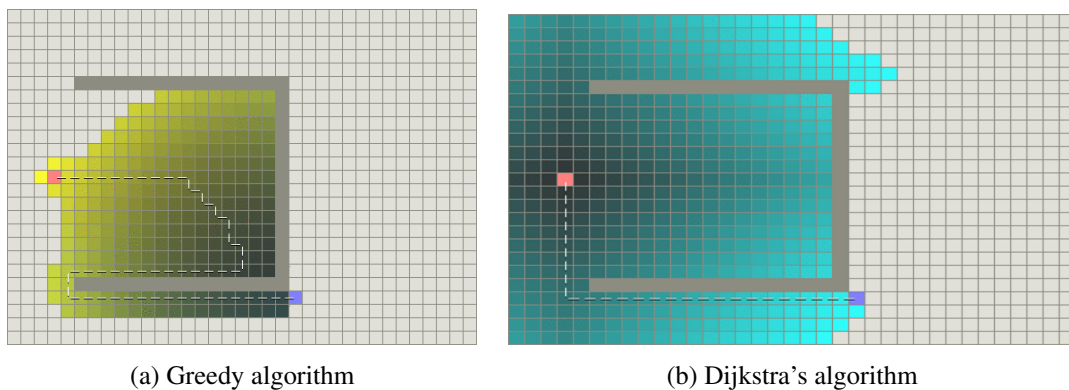


Figure 3: Path Planning

As you can see from both these figures that the path discovered using Dijkstras is the shortest path and the path found using Greedy tends to enter into the concave obstacle and then later on recovering to find its path.

A star algorithm is defined using a function $f(n) = g(n) + h(n)$ where $g(n)$ represents the exact cost of the path from the starting point to any vertex whereas $h(n)$ is the heuristic estimated cost function.

3.1.1 A star use of heuristic

The heuristic can be used to control A*'s behaviour.

- $h(n)=0$ then A star tends to perform like Dijkstras Algorithm.
- Lower the value of $h(n)$ then more closer is its behaviour to Dijkstras making it more likely for it to find the shortest path. It also expands more making it slower.

- Value of $h(n)$ is equal to the cost of moving from a vertex to a goal, then A star will find the best path. In this case it is not likely to expand into anything else making it faster but happens only in rare cases.
- If the value of $h(n)$ is greater than the cost of moving from the vertex to the goal then A star is may not find the best path but it certainly will run faster.
- If $h(n)$ has a very high value then it starts behaving like Greedy Best First Search. By suitably modifying the values of $g(n)$ and $h(n)$ we can control the behaviour and computational time for the planning of path.

3.1.2 Simulation Results

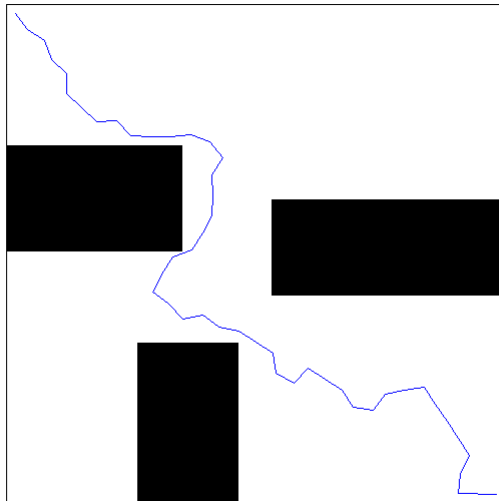


Figure 4: map1 Astar

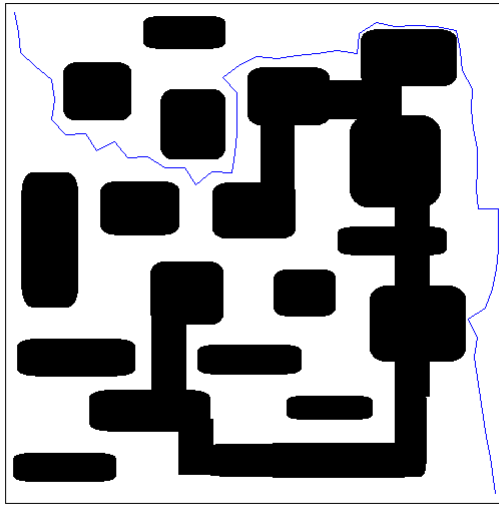


Figure 5: map2 Astar

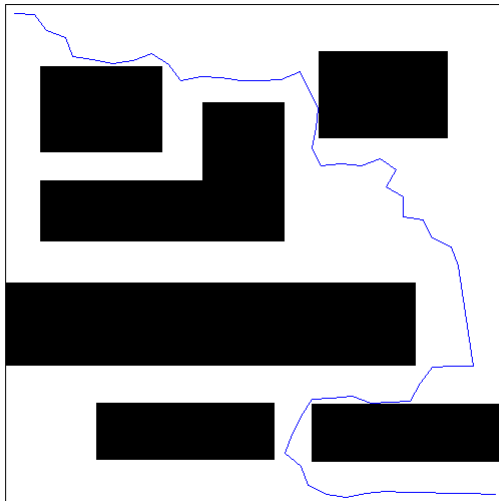


Figure 6: map1 Astar

3.2 Genetic Algorithm

Genetic Algorithm is a very efficient way of generating paths in a flat map or dangerous terrain containing obstacles. In this method unlike conventional Genetic algorithm where we are trying to optimize only one function, here we optimize under two criteria that is length and difficulty. Genetic Algorithm is an evolutionary optimization method which can solve optimization problem. The idea behind this method is that the solution to the optimization problem is to be viewed as an individual that has many parameters. These parameters are to be considered as genes that make up the individual (solution to the optimization problem).

Since Genetic Algorithm is an evolutionary method it tries to evolve p solutions or individuals from a population size of S through iteration. Each time we check the fitness of each individual by checking it with a given fitness function. Each generation produces a new set of individuals by crossover and mutation with the sole objective that the new off-springs will provide a much better solution than its previous generation. Then we select new individual for the next iteration using some selection algorithm and then replace them back into the population. This process is repeated for k generations.

3.2.1 Proposed Method for Path Planning

Input :

- We will take 500×500 map which contain obstacles in the form of black boxes.
- We will take $[0\ 0]$ as source point and $[490\ 490]$ as destination point.

Output :

The path must contain a series of points which is to be stored in a path array and it is subject to constraints and optimization criteria:

- The path must not pass through those solid black boxes.
- The path must stay inside the map.
- Path length must be optimized.

We must also establish a set of rules that Genetic algorithm will be operating under

-

- The obstacles are static in nature.

- It is assumed that we have sufficient knowledge of the map and the placement of the obstacles as this is an offline method of path planning.
- All paths produced by Genetic Algorithm are monotone in nature.
- The free traversable space will be white whereas the obstacle laden path will be black in colour.

3.2.2 Initial Population

The population size that is used will be p . Of the total population remaining, $p - 2$ of them are to be generated randomly while the remaining two present will be straight paths from s to d .

$$S = \{x_0, x_1, x_2, \dots, x_{p-2}, a, b\}$$

3.2.3 Path Repair Mechanism

The path that is obtained from Genetic Algorithm will be valid or invalid and the criteria for which a path will be considered invalid are as follows:-

- The path lies outside the map.
- The final point is not d .
- The path passes through obstacles.

3.2.4 Single Objective Function

We consider this path planning problem where we try to optimize the objective of minimizing the path length. To minimize the path length we will be using the fitness function:-

$$f(n) = n^2 - c$$

where c is the no of points in the array path.

As we had mentioned before-hand the fitness function for non-validity criteria will be

- $f(n) = 1$ when the path is out of bounds
- $f(n) = (n^2 - c)/20 * I$ where I is the number of points where the path passes through the obstacles.

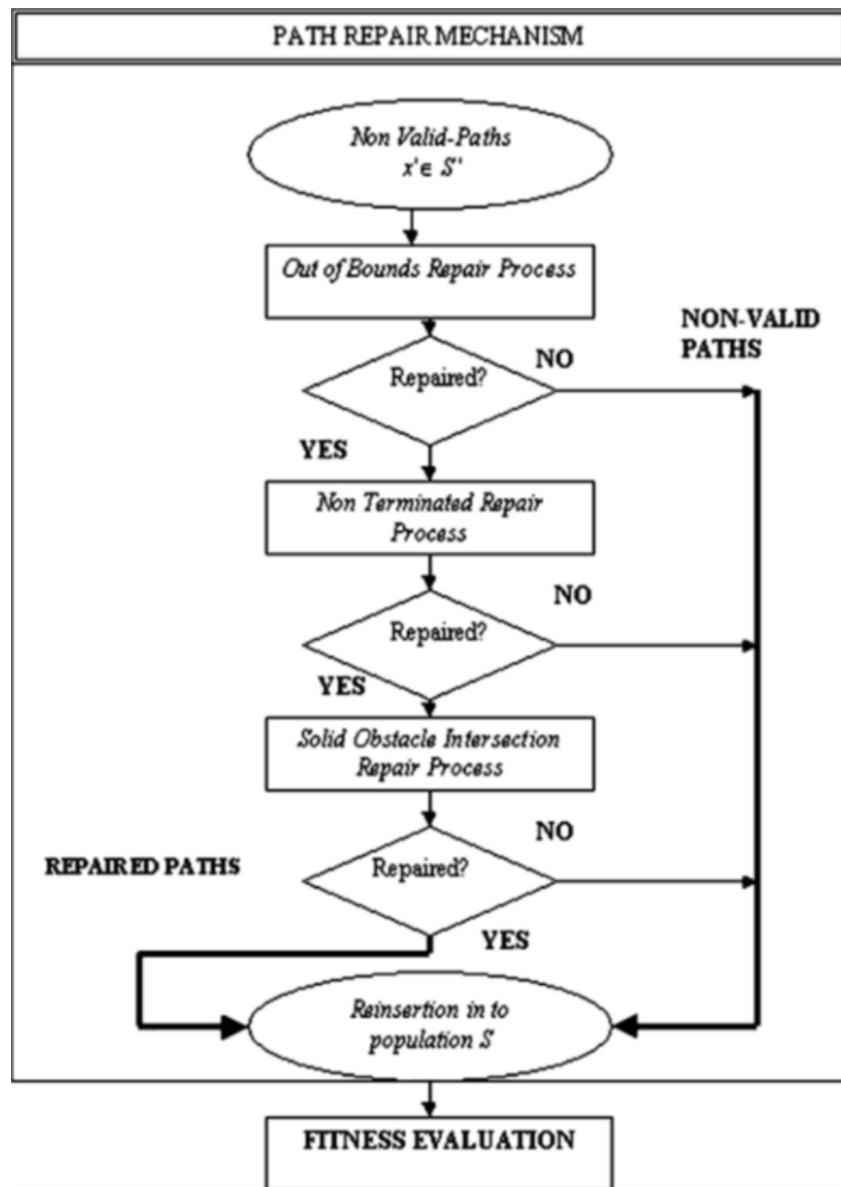


Figure 7: Path Repair algorithm

3.2.5 Two Objective Function

Another fitness function will be defined which will measure the path difficulty. It is calculated by the following fitness function:

$$f(n) = n^2 - wd$$

where sum of wd for each cell in a given path x . With this we are forced to use Pareto optimality for a rank based system for individuals in population S .

3.2.6 Genetic Operator

We have used single point crossover and one bit mutation as our genetic operator to create the next generation.

3.2.7 Selection Method

We used the Roulette selection method and for termination we fixed an upper criteria for k .

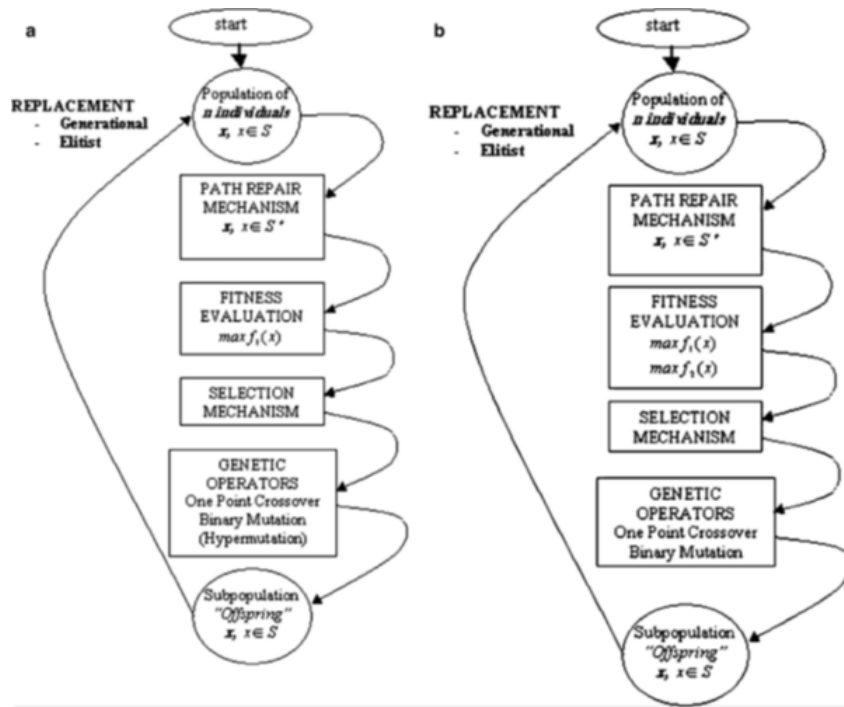


Figure 8: Algorithm of multi-objective GA

3.2.8 Simulation Results

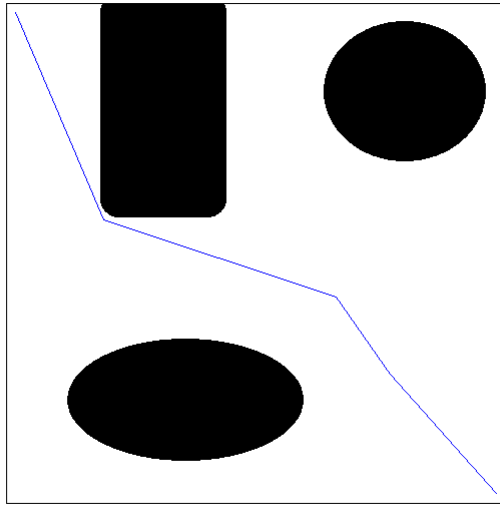


Figure 9: map 1 GA

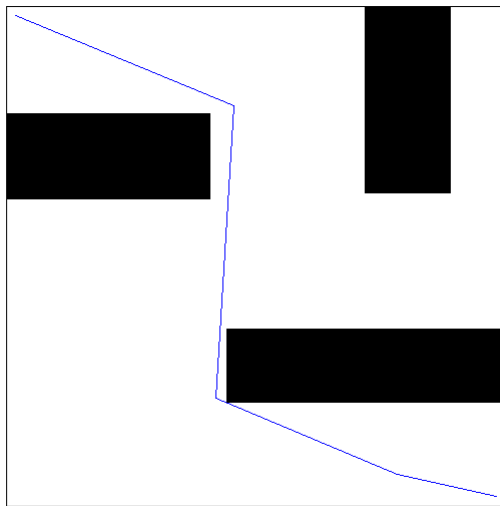


Figure 10: map 2 GA

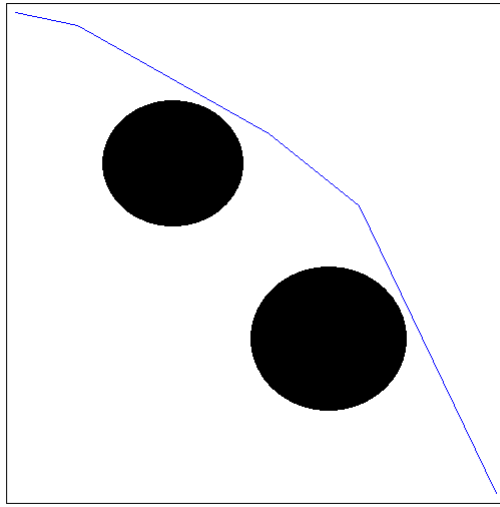


Figure 11: map 3 GA

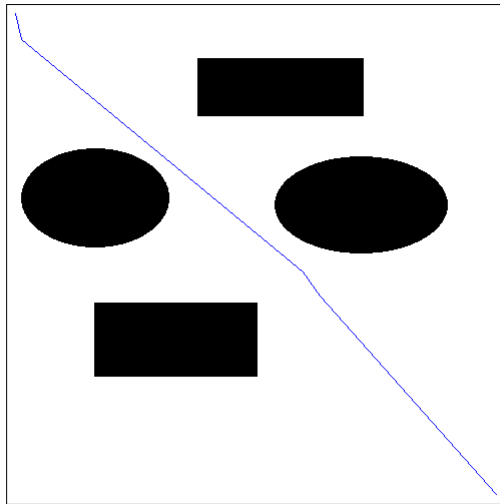


Figure 12: map 4 GA

3.3 Particle Swarm Optimization

In this method I am taking an image initially. For each agent I am initializing a random position. Then I am calculating the fitness values of each point. Then initializing random velocities as well. My fitness function in this is:

$$H = \sqrt{(x - 490)^2 + (y - 490)^2}$$

This is my fitness function. This is what I am trying to minimize while trying to evaluate new points.

After I am getting a new set of points I again tried to find the best point by trying to use the fitness function. The best points are plotted on the map. This works fine unless there is an obstacle. With obstacles the path will traverse straight through the obstacles. This is where the obstacle detection method comes into the picture. I have tried two methods which though have been unsuccessful needs to be mentioned.

In the first method I tried imposing penalty. In this method whenever a point after being minimized and passing the fitness function criteria would lie on an obstacle I would try to add a value (penalty) to its fitness which would make it unfit for being plotted. Then a new point would be found which if fit would be plotted on the map. Then this came with its own set of problems. Suppose two points, the one plotted before and one plotted after imposing penalty, lied above and below an object the spline would pass straight through the object which was inconvenient as it would make the robots path impossible. Now we have to look up for a second method.

In this method the moment a point is plotted on an obstacle the robot tries to plan a new path by finding a new point by changing the angles with the vertical. The angles need to be varied from 30° to 180° . The angle at which it finds a valid point it plots that point moves on with its plot. Here too we have less control over the spline which makes it inconvenient for us to draw a new path.

3.3.1 Simulation Results

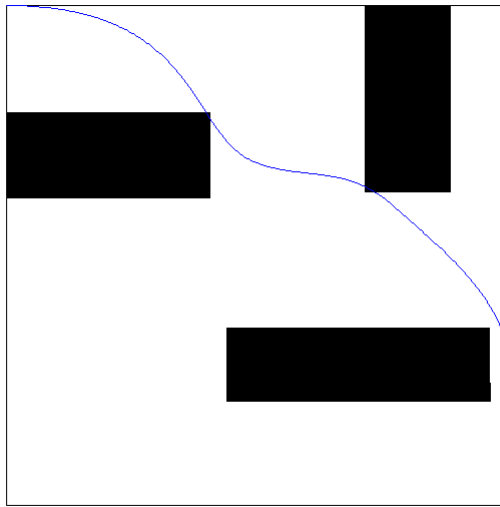


Figure 13: map 1 PSO

3.4 Artificial Potential Field Method

The potential field method is very successfully implemented in stationary environment. There are virtual forces that act upon the robot and guide it towards the goal. Here the robot is considered to be a particle in the space and it is assumed to be present in an artificial potential field. The path planning method is iterative in nature. At any given position we calculate the net force that is acting upon the robot. When the robot reaches its goal then the iterative process is stopped.

3.4.1 The Attractive Potential Function

Here we have taken the attractive potential function to be

$$U_{att} = \frac{1}{2}k_{att}\rho_g^2 \quad (3.4.1)$$

Here k_{att} is a constant scaling factor and $\rho_g = \|q - q_g\|$ is the Euclidean distance between the point and the goal.

The attractive force is obtained by differentiating the attractive potential function and we get

$$F_{att}(q) = -\nabla U_{att}(q) = -k_{att}(q - q_g) \quad (3.4.2)$$

3.4.2 The Repulsive Potential Function

Here we have taken the repulsive potential function to be

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)^2 & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases} \quad (3.4.3)$$

Here k_{rep} is the constant scaling factor, $\rho(q)$ is the distance of the robot from the obstacle and ρ_0 is the distance from the obstacle upto which the repulsive force is effective.

The repulsive force can be obtained as

$$F_{rep}(q) = -\nabla u_{rep}(q) = \begin{cases} k_{rep}\left(\frac{1}{\rho(q)} - \frac{1}{\rho_0}\right)\left(\frac{1}{\rho^2(q)}\right)\frac{q - q_c}{\|q - q_c\|} & \text{if } \rho(q) \leq \rho_0 \\ 0 & \text{if } \rho(q) \geq \rho_0 \end{cases} \quad (3.4.4)$$

3.4.3 Simulation Results

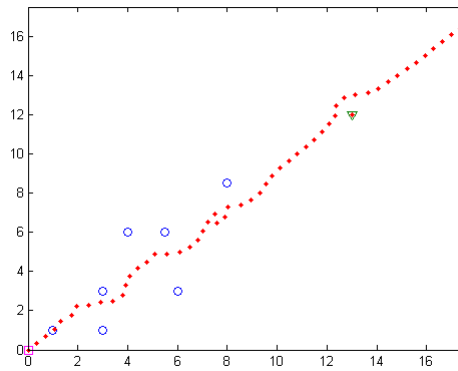


Figure 14: map 1 APF

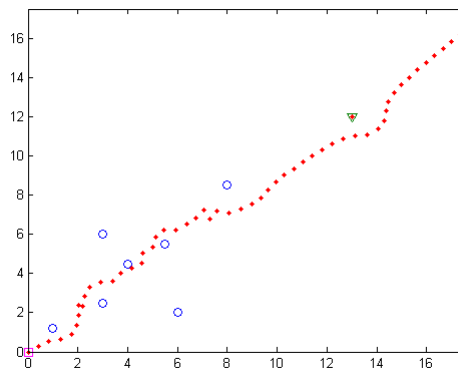


Figure 15: map 2 APF

4 Conclusion

The thesis has tried to solve the problem of finding an optimized path in a map using some evolutionary algorithms like Genetic Algorithm and Particle Swarm Optimization. It is showed here that these algorithms coupled with efficient obstacle detection algorithm can help robotic path planning. Here we have tried to optimize on two criteria: length and difficulty. Here we have described useful performance results and showed some simulation results on different maps. Then finally path planning using artificial potential field method has been implemented using the classical potential functions to get results.

References

- [1] Margarita Reyes-Sierra and CA Coello Coello. Multi-objective particle swarm optimizers: A survey of the state-of-the-art. *International journal of computational intelligence research*, 2(3):287–308, 2006.
- [2] Oscar Castillo, Leonardo Trujillo, and Patricia Melin. Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots. *Soft Computing*, 11(3):269–279, 2007.
- [3] Lu Yin, Yixin Yin, and Cheng-Jian Lin. A new potential field method for mobile robot path planning in the dynamic environments. *Asian Journal of Control*, 11(2):214–225, 2009.
- [4] AL-Taharwa Ismail, Alaa Sheta, and Mohammed Al-Weshah. A mobile robot path planning using genetic algorithm in static environment. *Journal of Computer Science*, 4(4):341, 2008.
- [5] Petr Švestka and Mark H Overmars. Coordinated path planning for multiple robots. *Robotics and autonomous systems*, 23(3):125–152, 1998.
- [6] Jianping Tu and Simon X Yang. Genetic algorithm based path planning for a mobile robot. In *Robotics and Automation, 2003. Proceedings. ICRA'03. IEEE International Conference on*, volume 1, pages 1221–1226. IEEE, 2003.
- [7] Yunfeng Wang and Gregory S Chirikjian. A new potential field method for robot path planning. In *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, volume 2, pages 977–982. IEEE, 2000.