Robot Motion Planning using genetic algorithm

BY

AKANKSHA BHARDWAJ

VINEELA MOSES

SHIVANI PAL

Introduction

The problem we are going to discuss here is the motion planning of AMR (Autonomous Mobile Robots) Robots in a static environment.

Motion planning is a problem in robotics for finding a collision free path which should be shortest, smoothest and optimal is other terms

Motion Planning

- Project has two parts : software and hardware
- We explore Evolutionary Computation for optimisation purpose.
- Implement a PID controller for cross track error minimisation and test on sample trajectories.
- Finally integrate both parts.

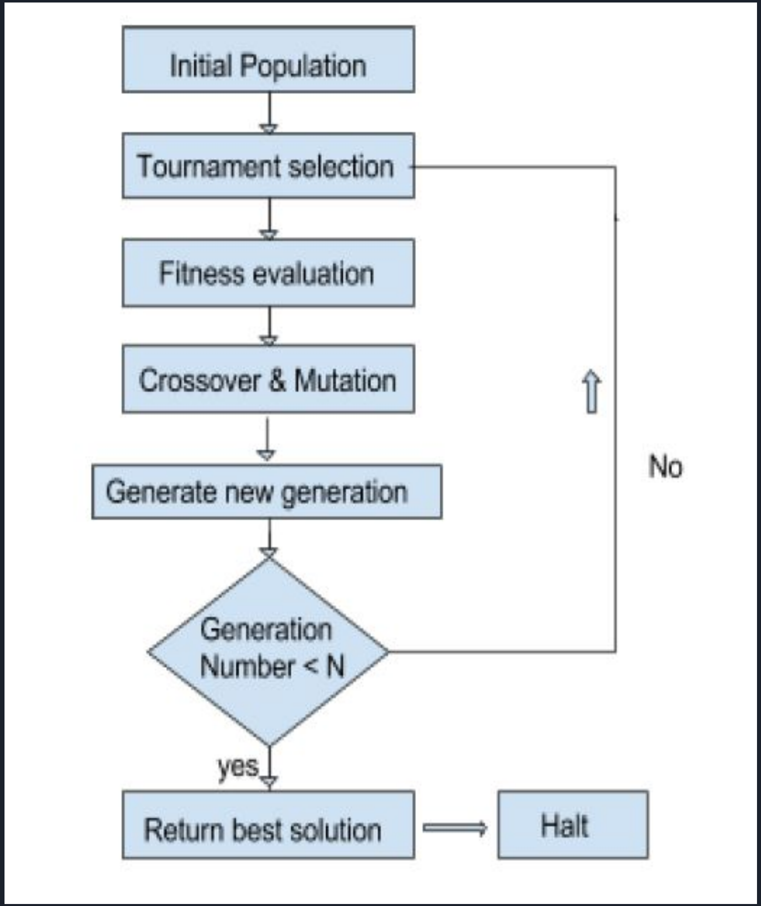
PROJECT GOALS

- Obtain collision free trajectories.
- Optimize the path between start and goal positions till Genetic algorithm converges.
- Implement a PID controller to minimize with crosstrack errors during motion.
- A smooth trajectory to address non-holonomic constraints like steering angle constraints

Genetic Algorithm

- A stochastic search technique, used to optimize an objective function over a known search space.
- We are using Real encoded GA. In general it's performance has been found better than binary enc

x1	y1	x2	y2	x3	y3	x4	y4	x5	y5



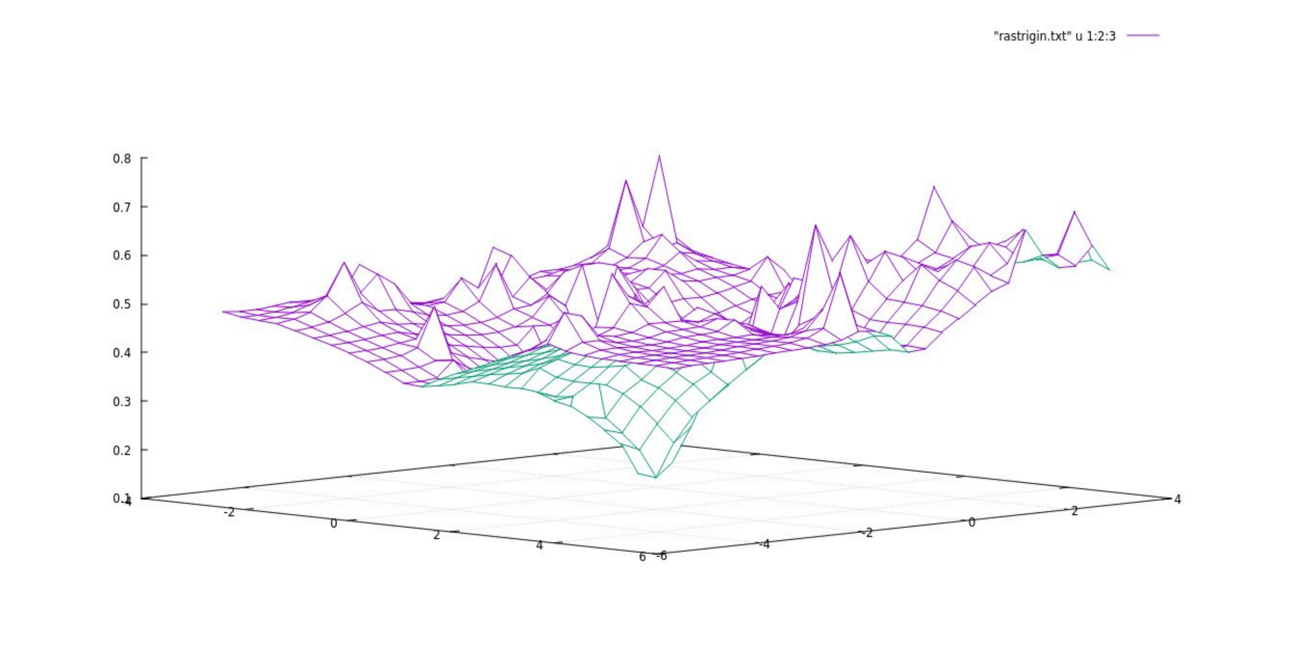
Flow of GA

Population is initialized with random but feasible trajectories.

PARAMETER VALUES FOR GA

Parameter	Values
Population size	40
Individual size	10
Crossover rate	0.8
Mutation rate	0.1
Tournament size	2
Uniform mutation width	0.5
Maximum generations	300
Minimum generations Maximum steady generations	100 50

PLOT OF RASTRIGIN TEST FUNCTION



RESULT

f(x1, x2) = 0.0000059571967

x1 =0.0000038869627

x2 = 0.000001002786

OBJECTIVE FUNCTION

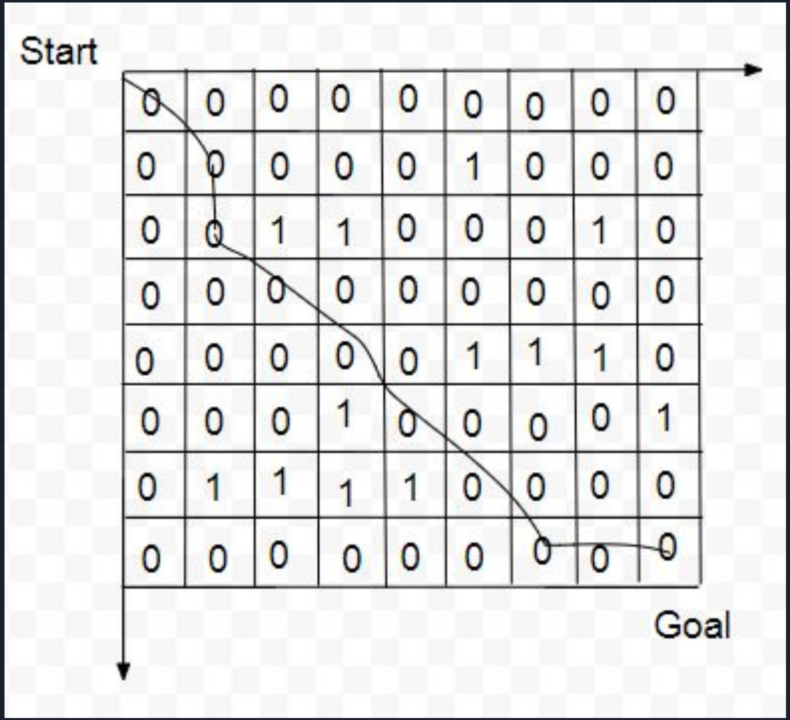
$$PL = \sqrt{(x_S - x_1)^2 + (y_S - y_1)^2} + \sum_{i=1}^{n} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} + \sqrt{(x_G - x_n)^2 + (y_G - y_n)^2}$$

$$F = PL + alpha * (points\ in\ obstacle)$$

A minimisation problem

x = (1 - T) * x1 + T * x2;

y = (1 - T) * y1 + T * y2;
Map[x][y] = 1 ? (obstacle)

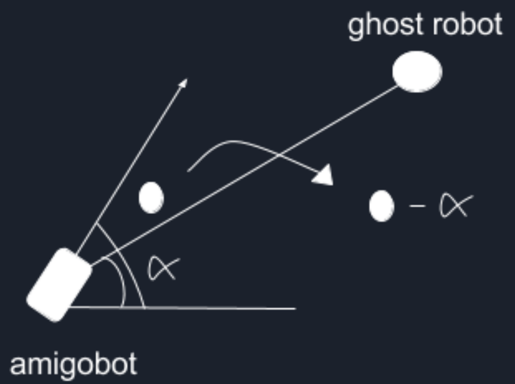


Map is converted into a grid

form where 1s denote obstacles and 0s empty space.

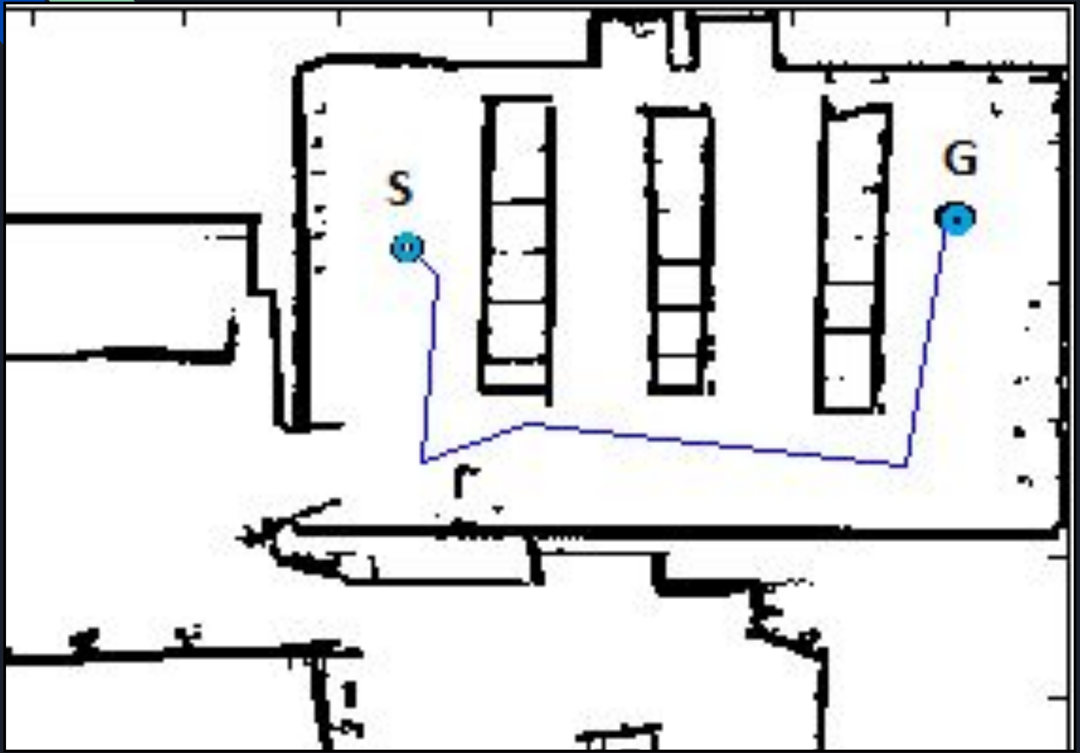
PID control and Cross track error calculation

distance	sqrt((cx-px)2+(cy-py)2)/100
alpha	((atan2(py-cy,px-cx)*180/PI)-theta
Frx	cos(alpha*PI/180)*dist
Fry	sin(alpha*PI/180)*dist
F	(Frx*Frx)+(Fry*Fry)
alphaf	atan2(Fry,Frx)*180/PI;
Linear velocity	abs(F)*Lc
Rotational velocity	alphaf*Rc;



The Development Kit

- AmigoBot a mobile robot with differential wheel drive
- ARIA Advanced Robot Interface for Applications (ARIA) is a C++ library (software development toolkit or SDK) for all MobileRobots
- MobileSim for simulating environments and MobileRobots
- SONARNL & Mapper3 sonar sensors are used for figuring the location using sensor data. And then into map.
- EO (Evolving Objects) an open source , ANSI-C++ library for genetic algorithm.



Scenario 1

Best Fitness Value

351.656

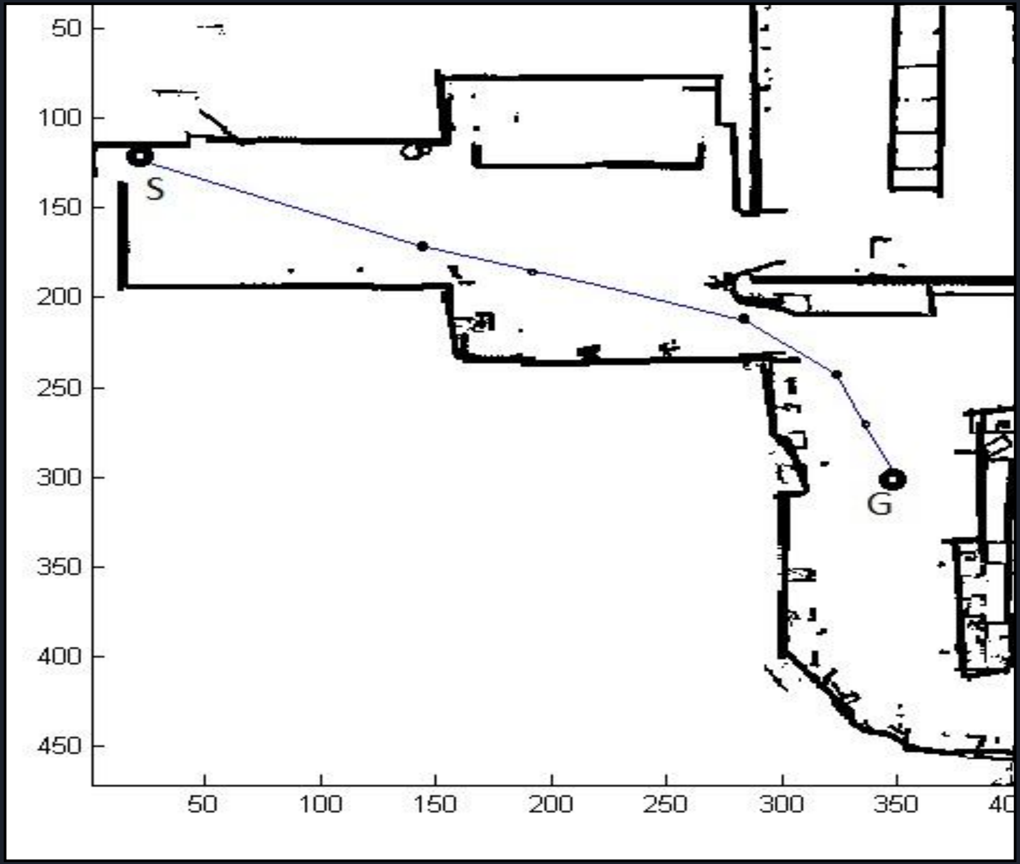
Last generation number
50

Scenario 2

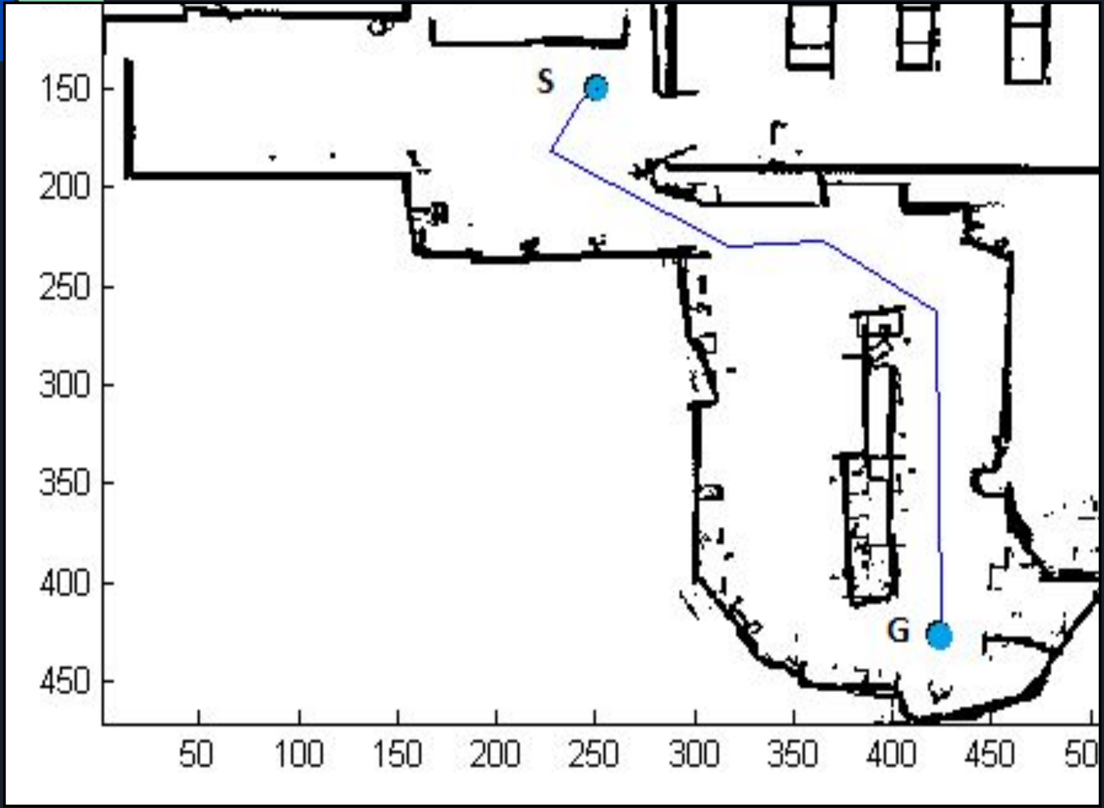
Best Fitness Value

434.59

Last generation number 100



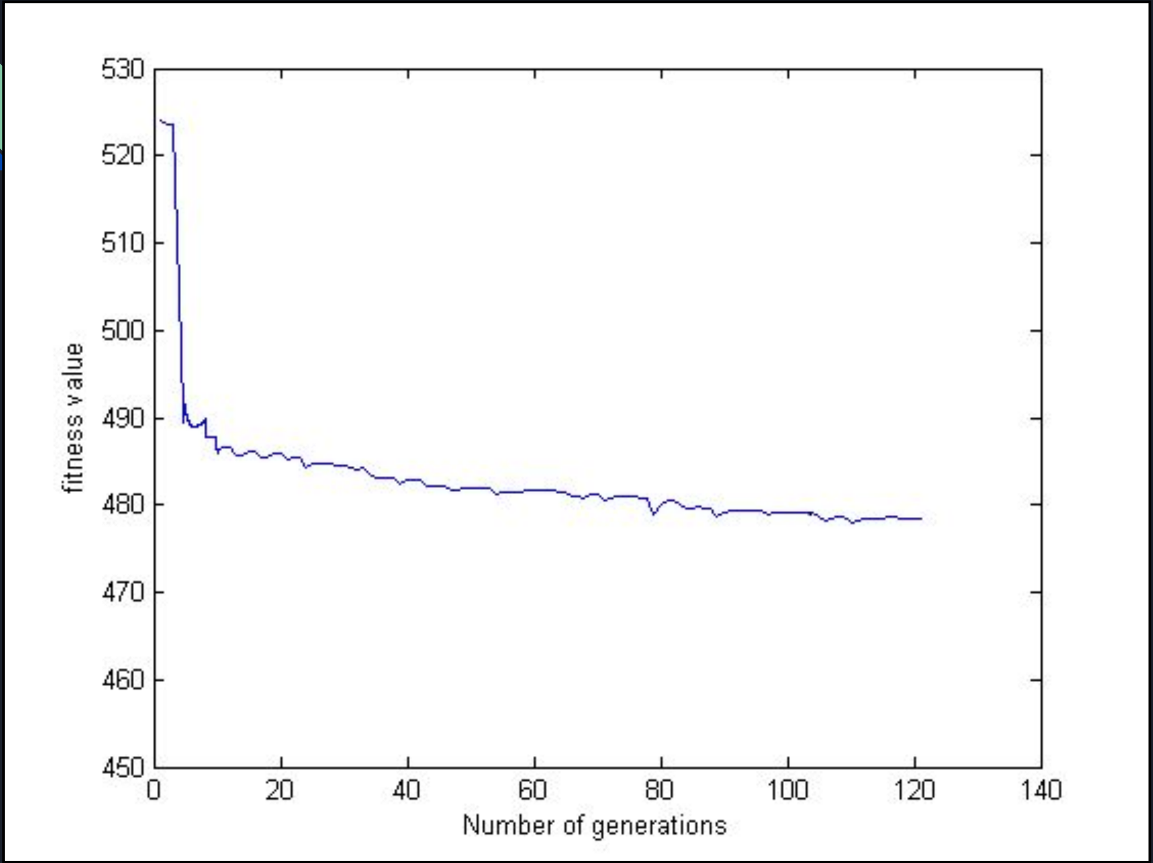
Scenario 3



Best Fitness Value

478.47

Last generation number
120



PLOT

NUMBER Of GENERATIONS Vs

FITNESS VALUES

For SCENARIO 3