

Optimizing Mobile Robot Path Planning and Navigation by Use of Differential Evolution

Shilpa Samal, B.Hemalatha

PG Student Department of Instrumentation & Control Engineering, SRM University, Chennai, India
Sr. Assistant Professor, Department of Instrumentation & Control Engineering, SRM University, Chennai, India

Abstract: Path planning and navigation is essential for an autonomous robot which can move avoiding the static obstacles in a real world and to reach the specific target. Optimizing path for the robot movement gives the optimal distance from the source to the target and save precious time as well. With the development of various evolutionary algorithms, the differential evolution is taking the pace in comparison to genetic algorithm. Differential evolution has been deployed quite successfully for solving global optimization problem. Differential evolution is a very simple yet powerful metaheuristics type problem solving method. In this paper we are proposing a Differential Evolution based path navigation algorithm for mobile path navigation and analyze its efficiency with other developed approaches. The proposed algorithm optimized the robot path and navigates the robot to the proper target efficiently.

Keywords- Path navigation, differential evolution, genetic algorithm, neural network.

I. INTRODUCTION

In this century we cannot think our life without technology. To reduce our work load and utilize time we developed various technologies. Robot is one of the emerging technologies which reduce human workload significantly. Simple robot work with very little facility and efficiency but the modern age needs a smart and intelligent robot which can do very large amount of work in an efficient way. So the idea of intelligent robot came into the scenario.

An intelligent robot should reach from one source point to a target point in an efficient and collision free manner. This is call path navigation. To achieve good and efficient path navigation the robot should plan intelligently. Good utilization of time and resource with maximum efficiency is the major criteria for a good path planning technique.

Researcher broadly categorizes the path navigation problem into two categories.

a) Global Path Planning:

This method work when the environment is known. Various researches have been carried out and various techniques have been proposed for this. Numeric potential fields [1], grid visits [2], genetic algorithm [3], behavior based model [4], graphs [5-7] etc. are among the proposed and used methods. The main disadvantage of this type of algorithm is the implementation process is not suitable for real world and computation cost is also large.

b) Local Path Planning:

In the real world the reaction time is very short hence the computation time must be very short to take a decision. So in a dynamic environment when the reaction time is short the robot needs a path planning technique which will take a correct decision in a short span of time. Many researchers have proposed many techniques like fuzzy logic [8][18], certainty grid [4] [9], the bug algorithm [10], the probabilistic models [11], potential fields [12][19], Differential Evolution algorithm [13][20] etc.

In this paper we are proposing a Differential Evolution algorithm

Step 1. Initialization

Step 2. Obstacles position and sizes

Step 3. Minimum distance robot should keep from obstacle

Step 4. Run algorithm

Step 5. Object to minimize

Step 6. Calculate constraints

Step 7. Stopping criteria for DE

II. PREVIOUS METHODS

Robot can be used as a substitution of human being where human cannot reach or where we need a tedious work to do. So, a decade ago researchers started to develop intelligent robot. The main idea behind this was to develop an autonomous robot which can perform a given work in a intelligent and efficient way. Before carrying out other works a robot must navigate itself to the target in a collision free and optimized way. Because sometime situation arise where the robot must take a decision itself in a short span of time. So a proper path planning method should be taken which will react in a short span of time with a optimize path. Researcher used several technologies to achieve this goal. Artificial Neural Network (ANN), Genetic Algorithm (GA) etc. are the most popular used among them

A. *Artificial Neural Network (ANN)*

The most complex and intelligent system is our Central Nervous System (CNS). This system inspired the researcher to develop a system which will work similarly as our nervous system and will make decision like a human being. Hence they developed Artificial Neural Network (ANN). ANN is an iterative learning method i.e its learn patterns and other things just like CNS learn. So more it iterates more the efficiency increase. The core element in an ANN is named as neuron. An ANN contains different layer depending upon various system. ANN model involves three basic steps:

- a. The interconnection between neurons.
- b. The learning and updating process.
- c. The activation function.

The three steps are interconnected and changed drastically if one steps change. The most significant step among these is learning process. Each and every time it percept new pattern it change weight and learning rate accordingly and trigger a neuron (activation function) as a output. But the main advantage is this model does not need a recompilation from the beginning like other learning model and mathematically the solution is optimal for a given case. But Siegelmann et al. [14] first shown that this complex model has a limitation like any other mathematical model .This model give best result when the training samples increased but the computation time also increase correspondingly.

B. *Genetic Algorithm(GA)*

Genetic algorithm is basically search heuristic mathematical [15] models which imitates the process of the natural evolution. Genetic algorithm provide a population of candidates solution to achieve an optimize solution for given problem. Genetic algorithm use different type of encoding methods i.e. 0 and 1, and others encoding technique [16]. The main advantage of this algorithm is its produce a large number of solutions for a given problem which interns give us more option to choose the best result among the best. But this algorithm suffers a major problem also. Sometimes its fails to choose the best global optimize solution. So this algorithm is not suitable for path planning where we need a accurate and moreover optimized solution.

III. PROPOSED ALGORITHM

For finding an optimal solution in today's world of science and engineering global optimization is a major problem . The global optimization is the most efficient one for finding the best parameter configuration. The global continuous optimization becomes more challenging when the situation comes to optimize many local optima. . Evolutionary algorithm (EA) has been proposed for solving the global continuous optimization problem [17]. The most popular EA is genetic algorithm but due to its time consuming process it has been overcome by differential evolution algorithm.

By applying EA which is a direct search method, a variation in a design parameter vector can be obtained from a given vector. Depending on the reduction of the objective function value the new parameter vector obtained can either be selected or rejected. This algorithm is referred as greedy algorithm, but it usually get trapped in local minima, as a result DE algorithm is being used which run several vectors simultaneously

DE algorithm is mostly used when the need of optimizing a population arises. DE is a optimization technique which converges to an optimum solution by iteratively modifying a population of candidate solution. The difference between DE and GA arises during its mutation scheme. One key aspect of DE algorithm is the mutation step size is dynamic in nature that is it adapts to the configuration of population and tends to zero when it converges. This makes DE less vulnerable to genetic drift than GA.

The problems objectives which incorporate constraints can be modeled by an objective function, minimized by using DE algorithm which is a stochastic optimization method. The algorithm mainly has three advantages; regardless of the initial parameter values it finds the true global minimum, convergence speed is fast, and uses a few control parameters. Being easy to use, fast, simple, adaptable for integer and discrete optimization, optimizes non-linear constraint which include penalty function and also optimizes multi-model search space are certain important features of DE.

Basic Mechanism Of De Algorithm

The following steps are considered in the designing of GE algorithm.

Step 1: The (N x D) population is initiated randomly within the domain described by Xmin and Xmax by all target vector(X). Each ith target vector at the gth generation of evolution can be represent as given below.

Equation: $X_i^g = [x_{(i,1)}^g, x_{(i,2)}^g, \dots, x_{(i,j)}^g, \dots, x_{(i,D)}^g]$ ----- equation 1.

X: Target vector

Step 2: Mutant vectors (M) can be produced by stochastic differential operation.

Equation: $M_i^g = [m_{(i,1)}^g, m_{(i,2)}^g, \dots, m_{(i,j)}^g, \dots, m_{(i,D)}^g]$ -----
- equation 2.

M: Mutant Vector

Step 3: The five common evolutionary strategies are

1. DE/rand/1 : $M_{(i)}^g = X_{(\alpha)}^g + F_1 * (X_{(\beta)}^g - X_{(\gamma)}^g)$ ----- equation 3.

X: Target Vector

M: Mutant Vector

F₁: Fitness Parameter

2. DE/best/1:

$M_{(i)}^g = X_{best}^g + F_1 * (X_{(\alpha)}^g - X_{(\beta)}^g)$ ----- equation 4.

X: target vector

M: Mutant Vector

F₁ : Fitness Parameter

3. DE/rand-best/1 :

$M_{(i)}^g = X_{(i)}^g + F_1 * (X_{best}^g - X_{(i)}^g) + F_2 * (X_{(\alpha)}^g - X_{(\beta)}^g)$ -----
equation 5.

4. DE/best/2 :

$M_{(i)}^g = X_{best}^g + F_1 * (X_{(\alpha)}^g - X_{(\beta)}^g) + F_2 * (X_{(\gamma)}^g - X_{(\zeta)}^g)$ -----
equation 6.

5. DE/rand/2 :

$M_{(i)}^g = X_{(\alpha)}^g + F_1 * (X_{(\beta)}^g - X_{(\gamma)}^g) + F_2 * (X_{(\zeta)}^g - X_{(\eta)}^g)$ ----- equation
7.

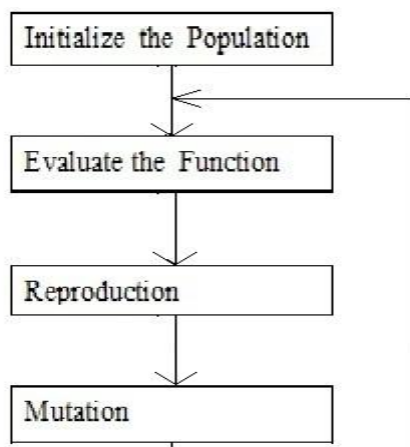
$\alpha, \beta, \gamma, \zeta, \eta, i$ are mutually exclusive integers randomly generated within the range[1,N]

Step 4: Perform crossover to obtain trail vectors (T) and reinsert weaker target vectors with better trail vectors.

$t_{(i,j)}^g = \begin{cases} m_{(i,j)}^g & \text{if } (rand_{(i,j)} \leq P_c \text{ or } j = j_{rand}) \\ x_{(i,j)}^g & \text{otherwise} \end{cases}$ ----- equation

8.

Step 5: Store the best and iterate till it terminate.



$$\begin{bmatrix} X_{11} & X_{12} & \dots & X_{1N} \\ X_{21} & X_{22} & & X_{2N} \\ \vdots & & \ddots & \vdots \\ X_{M1} & & \dots & X_{MN} \end{bmatrix} \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_M \end{bmatrix}$$

Where, M-> Population size, N-> Number of variables, X-> Deciding variables, F-> Fitness value
The best individual in each iteration is preserved. The best among such elites is the desired optimum solution.

IV. RESULT ,CONCLUSION AND DISCUSSION

To conduct the experiment regarding the implementation of our algorithm, we use MATLAB tool. The experiment has been conducted on window home basic with INTEL CORE™ i3 processor. The system requirement include 4GB RAM with 2GB Graphics. We have to use Mat lab (R2010a) and conducted test on various scenario.

A. Scenariol

The experiment is conducted for three different scenarios. The first scenario is based on fixed input and fixed output location. The experiment is conducted in a (500,500) co-ordinate. The scenario is static nature in which number of obstacle are fixed. Figure 1and table 1 shows the simulation result.

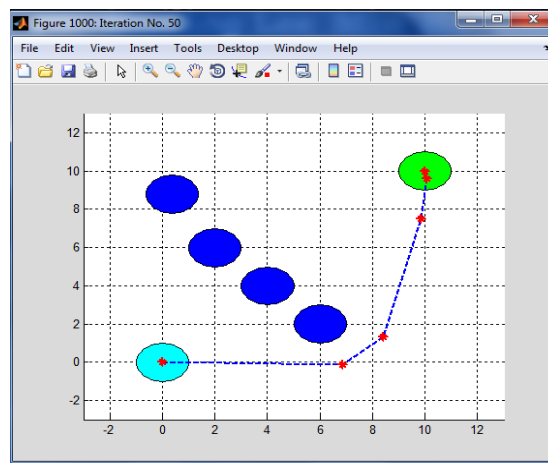


Figure 1: Avoiding Four Static Obstacles

Turn to angle in bent1(in degree)	Move forward distance bent1(in unit distance)	Turn to angle in bent2(in degree)	Move forward distance bent2(in unit distance)	Turn to angle in bent3(in degree)	Move forward distance bent3(in unit distance)	Turn to angle in bent4(in degree)	Move forward distance bent4(in unit distance)
-1.076	6.879	45.548	2.100	80.365	6.368	88.086	2.104

Table1: Distance and Angle For Each Bent

Optimal X	6.878	8.402	9.862	10.078
Optimal Y	-0.123	1.322	7.521	9.614

Table2: Best Path

B. Scenariol2

This scenario is conducted in the same (500,500) co-ordinate. There are certain static geometric shape obstacles in the given environment .The scenario has user input for the robot starting location but a fixed target location so that the robot, if given any location point can reach the target by avoiding the obstacles which come in its way. The navigation of the robot for finding an optimal path by avoiding the obstacles has been successfully conducted. The output mostly provides optimal x and optimal y value. Figure 2 and table 3 ,4 shows result.

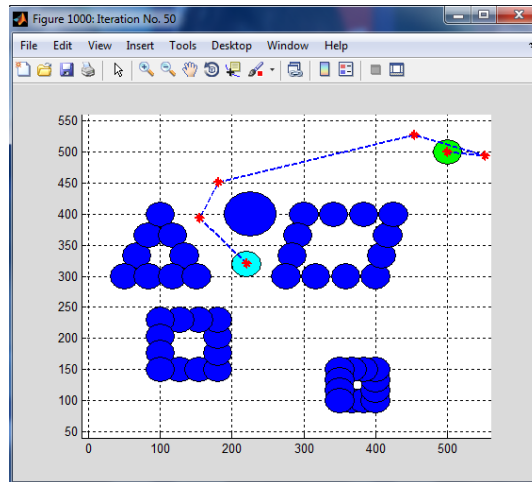


Figure2: Avoiding Sixty-one Static Obstacles

Turn to angle in bent1(in degree)	Move forward distance bent1(in unit distance)	Turn to angle in bent2(in degree)	Move forward distance bent2(in unit distance)	Turn to angle in bent3(in degree)	Move forward distance bent3(in unit distance)	Turn to angle in bent4(in degree)	Move forward distance bent4(in unit distance)
137.92	97.936	68.709	63.874	16.2512	283.819	-19.425	103.21

Table3: Distance And Angle For Each Bent

Optimal X	154.844	181.217	454.689	553.544
Optimal Y	393.118	451.293	527.230	494.394

Table4: Best Path

C. Scenario3

The third scenario is random by nature in the (500,500) co-ordinate. The “Rand” and “Randi” mat lab functions have been used for this scenario.

The program does not only work for this three scenario but it will be working for any given scenario in the same (500, 500) co-ordinate with 61 number of obstacles. Figure 3 and table 4,5 shows the result.

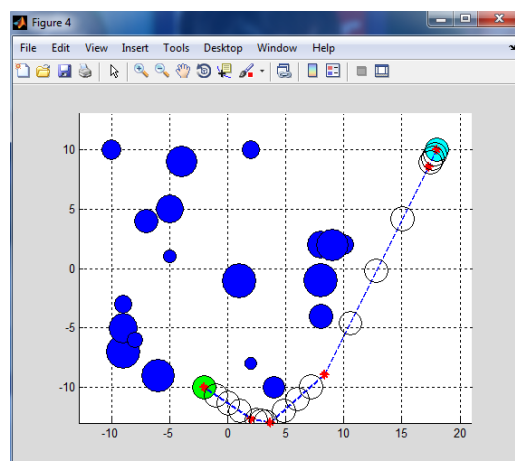


Figure3: Avoiding Twenty-One Static Obstacles

Turn to angle in bent1(in degree)	Move forward distance bent1(in unit distance)	Turn to angle in bent2(in degree)	Move forward distance bent2(in unit distance)	Turn to angle in bent3(in degree)	Move forward distance bent3(in unit distance)	Turn to angle in bent4(in degree)	Move forward distance bent4(in unit distance)
-121.55	1.618	-122.70	19.671	-145.35	6.116	176.92	1.673

Table4: Distance And Angle For Each Bent

Optimal X	17.288	8.302	3.700	2.057
Optimal Y	8.534	-8.953	-12.981	-12.661

Table5: Best Path

This paper discuss the path navigation of the robot in different scenario and finding the optimal path value both in x and y axis. The proposed DE algorithm has been successfully implemented and the simulation result has been shown above. The program does not only work for this three scenario but it will be working for any given scenario in the same (500, 500) co-ordinate with 61 number of obstacles. A brief survey of both DE algorithm and GA algorithm has been performed. Advantages and disadvantages of these methods are also projected out.

This work could be further extend for real world application where we can develop this work using hardware and image processing could be used as method for object detection and classification. This needs a through and massive analysis and research.

ACKNOWLEDGMENT

The authors would like to thanks Dr. A.Vimala Juliet (HOD Electronics and Control Engineering) for her kind support. We would also especially grateful to SRM University for its infrastructure.

REFERENCES

- [1] Barraquand, Jerome, Bruno Langlois, and J-C. Latombe. "Numerical potential field techniques for robot path planning." *Systems, Man and Cybernetics, IEEE Transactions on* 22.2 (1992): 224-241.
- [2] Balch, Tucker. "Avoiding the past: A simple but effective strategy for reactive navigation." *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993.
- [3] Shibata, Takanori, and Toshio Fukuda. "Coordinative behavior by genetic algorithm and fuzzy inevoluntary multi-agent system." *Robotics and Automation, 1993. Proceedings., 1993 IEEE International Conference on*. IEEE, 1993.
- [4] Lim, Jong Hwan, and Dong Woo Cho. "Experimental investigation of mapping and navigation based on certainty grids using sonar sensors." *Robotica* 11.01 (1993): 7-17.
- [5] Liu, Y-H., and Suguru Arimoto. "Proposal of tangent graph and extended tangent graph for path planning of mobile robots." *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991.
- [6] Garcia, F., and Roger Mampey. "Mobile robot motion planning by reasoning both at itinerary and path levels." *Advanced Robotics, 1991. Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*. IEEE, 1991.
- [7] Iyengar, S. S., et al. "Robot navigation algorithms using learned spatial graphs." *Robotica* 4.2 (1986): 93-100.
- [8] Beaufre, Bertrand, and Said Zeghloul. "A mobile robot navigation method using a fuzzy logic approach." *Robotica* 13.05 (1995): 437-448.
- [9] Thrun, Sebastian. "Learning metric-topological maps for indoor mobile robot navigation." *Artificial Intelligence* 99.1 (1998): 21-71.
- [10] Skewis, T., and V. Lumelsky. "Experiments with a mobile robot operating in a cluttered unknown environment." *Journal of robotic systems* 11.4 (2007): 281-300.
- [11] Borenstein, Johann, and Yoram Koren. "The vector field histogram-fast obstacle avoidance for mobile robots." *Robotics and Automation, IEEE Transactions on* 7.3 (1991): 278-288.
- [12] Tianmiao, Wang, and Zhang Bo. "Time-varying potential field based perception-action behaviors of mobile robot." *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*. IEEE, 1992.
- [13] Deyun, Feng Qi Zhou. "Time Optimal Path Planning Based on Differential Evolution Algorithm [J]." *Computer Engineering and Applications* 12 (2005): 021.
- [14] Siegelmann, Hava T., and Eduardo D. Sontag. "On the computational power of neural nets." *Journal of computer and system sciences* 50.1 (1995): 132-150.
- [15] Mitchell, Melanie (1996). *An Introduction to Genetic Algorithms*. Cambridge, MA: MIT Press.
- [16] Whitley, Darrell (1994). "A genetic algorithm tutorial". *Statistics and Computing* 4 (2):pp. 65-85
- [17] GA, Dervis Karabo, and Selcuk Okdem. "A simple and global optimization algorithm for engineering problems: differential evolution algorithm." *Turk J Elec Engin* 12.1 (2004).
- [18] Al-Khatib, Mohannad, and Jean J. Saade. "An efficient data-driven fuzzy approach to the motion planning problem of a mobile robot." *Fuzzy Sets and Systems* 134.1 (2003): 65-82.
- [19] Koren, Yoram, and Johann Borenstein. "Potential field methods and their inherent limitations for mobile robot navigation." *Robotics and Automation, 1991. Proceedings., 1991 IEEE International Conference on*. IEEE, 1991.
- [20] Liu, Bo, Ling Wang, and Yi-hui Jin. "Advances in differential evolution." *Control and Decision* 22.7 (2007): 721.