# A Neural Network based Path Planning Algorithm for Extinguishing Forest Fires

**M.P.Sivaram Kumar[1], S.Rajasekaran[2]**

**[1] Research Scholar/ Department of CSE**
**B S Abdur Rahman University**
**Vandalur, Chennai-48**

**[2] Department of Mathematics**
**B S Abdur Rahman University**
**Vandalur, Chennai-48**

**Abstract**

In this work an algorithm for automatic detection and suppression of Forest fires is proposed. The algorithm is implemented using parallel distributed model of neural network with three activation functions to determine the next consecutive moves to the cells for the actor. The algorithm uses reinforcement learning with weights determined dynamically in each iteration. The Entire forest is decomposed into grid of square cells with initial position of the Actor is assumed to be the cell 1 and the goal cell is the cell where the fire has occurred. The neural network model uses starting cell, goal cell and number of cells in each row or column and three activation functions to determine the next consecutive cells in which the robot has to travel. It uses only three movements LEFT, DIAGONAL and UP to reach the target cell. After calculating next cell, the check will be made for presence of obstacles in that cell. If there is any obstacle in that cell, then one cell from other two cells obtained using other two movements, which is free from obstacle will be chosen for next move. Then the cell number is stored in memory. This process is repeated till the next cell computed is same as the goal cell. The Actor will begin to move from start cell and reach the goal cell using the cell numbers available in the memory to extinguish Forest fire. This algorithm is designed keeping in mind only static obstacles and hence it works well for Forest environment with static obstacles. Computer simulation results show that path has been found successfully without collision with obstacles.

***Keywords:*** Forest fires, Path Planning, Neural Networks.

## 1. Introduction

One of the most important issues to be considered in Mobile Robot Navigation is planning of path to be used, which can be done offline or on line from a start cell to a goal cell without collision with obstacles. In this work we have used offline path planning. i.e. the path planning can be done before the robot starts to move. A plethora of works are available in the literature based on path planning of mobile robots using variety of techniques such as classical approaches probabilistic approaches and others which include Genetic Algorithm, Fuzzy Logic, Ant Colony Optimisation,Neuro Fuzzy Logic and Neural Networks. In this work we have presented a parallel distributed model of neural network for path planning of the Actor (which is same as robot) to extinguish Forest fire. The remainder of the section is organized as follows: Section2 describes the related works available in the literature. In section3 we have presented the Assumptions used in developing the model. Section 4 describes Algorithm for path planning to extinguish fires. In section 5 the implementation of algorithm using Parallel distributed Neural Network Model is discussed. Section 6 exhibits the simulation results while section 7 concludes the paper.

### 2. Related works in the Literature

An artificial neural network is a system based on the operation of biological neural networks. A neural network structure is defined as a collection of parallel processors connected in the form of graph, organized in a way such that the structure itself provides solutions to the problem being considered [1]. In [2] applicability of neural network techniques to a complex control problem in automotive industry was proposed. The design of control system was achieved using dynamical neural network models. A sensor based navigation scheme which makes use of a global representation of the environment by means of self organizing or Kohonen network is presented in [3]. A neural map which offers promising alternative to distance transform and harmonic function methods for both global and local navigation is presented in [4]. A neural network model learned from human driving data is introduced to model obstacle avoidance through dense areas of obstacles is presented in [5] and is tested on different scenarios and compared using cross validation to determine optimal network structure. Danica Janglova in [6] presented a method of construction of a collision free path for moving robot among obstacles based on two neural networks. The

first neural network determines "free" space using ultrasound range finder data and second one finds a safe direction for the next robot section of the path in the workspace while avoiding nearest obstacles. A three dimensional terrain mapping and classification technique which allows the operation of mobile robots in outdoor environments using laser range finders is described in [7]. It used multi layer perceptron neural network to classify the terrain. The maps generated by this approach can be used for path planning, navigation and local obstacle avoidance. Subhrajit Bhattacharya and Siddharth Talapatra in [8] presented a model based on competitive learning for robot motion planning. It uses modified version of shunting equation describing the dynamics of the neurons. Also new algorithm for updating neural activities has been suggested. In [9] a dynamic artificial neural network based mobile robot motion and path planning system is described. The method is able to navigate a robot car on flat surface among static and moving obstacles. The motion controlling ANN is trained online with an extended back propagation through time algorithm using potential field for obstacle avoidance. In [10] self organizing spiking model neural network is introduced and applied to Mobile robot environment representation and path planning problem. The best path is found by using modified A* Algorithm. In [11] Artificial neural network an extension of back propagation through time algorithm is used for path planning and potential fields is used for obstacle avoidance. In [12] path planning of a mobile robot by using a modified Hopfield neural network is presented. There are two approaches in which a neural network can be organized to operate in parallel [13]. Artificial Neural Systems functions as parallel distributed computing networks and their basic characteristic is architecture [14]. In our work we use the approach of different networks working in parallel with each network extracting different features.

## 3. Assumptions used in Model

1. The forest domain is decomposed into M x N grid of square cells [15].
2. The forest domain decomposition for 5 x 5 grids of square cells are shown in Fig 1.
3. Each cell in the grid contains an Anchor sensor node which knows the location based on Integers.
4. The Actor (Robot) is available at the cell 1 which is always the start cell and the cell in which fire occurs is always the goal cell.
5. Once a fire occurs inside a particular cell, it will be detected by the sensor placed inside the Cell first and the sensor sends message containing coordinates of the cell to the Actor [15]. Thus Actor Knows both start and goal cells. Then it uses the algorithm implemented using Neural Network to find a path.
6. Obstacles are static and size of the obstacle is similar to the size of the cell.
7. Two adjacent cells will have obstacles either lengthwise or breadth wise but not combined.
8. Since the actor is available at the cell 1 and based on assumption 6 and 7, Only 3 movements are sufficient to navigate the entire domain. They are (i) UP denoted by 0 (ii) DIAGONAL denoted by 1 and (iii) LEFT denoted by 2 as shown in fig 2 where CPA denotes the Current Position/cell of the Actor.

| | | | | |
|---|---|---|---|---|
| **1** | **2** | **3** | **4** | **5** |
| **6** | **7** | **8** | **9** | **10** |
| **11** | **12** | **13** | **14** | **15** |
| **16** | **17** | **18** | **19** | **20** |
| **21** | **22** | **23** | **24** | **25** |

**Fig 1 Decomposition of forest using 5 x 5 grids with coordinates based on Integers**

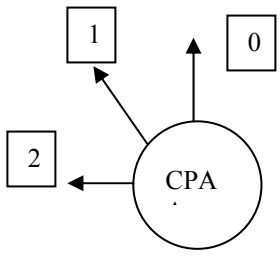



**Fig 2 Directional Movements of the Actor**

## 4. Algorithm for path planning to extinguish fires

The Actor placed in the cell whose location is 1 uses the algorithm shown below to estimate the sequence of

points which does not contain any obstacle. Then it will move through these points to reach the goal cell where the fire has occurred and start to extinguish it. The algorithm is shown below:

### 4.1 Algorithm for Estimating the path

Let s be the start position, g be the goal position and n be the number of cells in row or column.

```
location = 0
Store the start position in first location of memory path.
While (s not equal to g)
{
  location =location +1
  if ((absolute value (s-g)) mod (n+1) = 0) then
     {
       if (location s+n+1 contain obstacle) then
        {
          Check up move cell i.e. s+n and left move cell
          s+1
          If (both cell does not contain obstacle or left
              move cell contain obstacle) then
            {
              s = s+n
            }
          If (up move cell only contain obstacle) then
            {
              s = s +1
            }
        }
      else
        {
          s = s + n+1
        }
     }
  else if ((absolute value (s-g)) mod n = 0) then
          {
            if (location s+n contain obstacle) then
              {
                Check diagonal move cell i.e. s+n+1 and
                 left move cell s+1
                If (both cell does not contain obstacle or
                left move cell contain obstacle) then
                  {
                    s = s+n+1
                  }
                If (diagonal move cell only contain
                   obstacle) then
                  {
                    s = s +1
                  }
              }
            else
              {
                s = s + n
              }
          }
        else
          {
            if (location s+1 contain obstacle) then
             {
               Check up move cell i.e. s+n and diagonal
               move cell s+n+1
               If (both cell does not contain obstacle or up
               move cell contain obstacle) then
                    {
                      s = s+n+1
                    }
               If (diagonal move cell only contain
                  obstacle) then
                    {
                      s = s + n
                    }
             }
            else
             {
               s= s + 1
             }
          }

    Store the point s in location
}
```

## 5. Implementation of Algorithm using Parallel distributed Neural Network Model

The parallel distributed neural network model is shown in Fig 2. It uses 3 neurons which takes two inputs cell s multiplied by weight and bias and sums it. It uses reinforcement learning i.e. network is designed in such a way that each time best next move will be selected out of three possible moves. The move selected means it is rewarded and moves not selected means they are punished. The weights used in the model are binary weights and takes the value either 0 or 1.The weight will be calculated for each neuron separately using the formula shown below:

$$w1 = \begin{cases} 1 & \text{if ((abs(s-g)) mod (n+1) = 0)} \\ 0 & \text{other wise} \end{cases}$$

$$w2 = \begin{cases} 1 & \text{if ((abs(s-g)) mod n = 0)} \\ 0 & \text{other wise} \end{cases}$$

$$w3 = \begin{cases} 1 & \text{if (((abs(s-g)) mod (n+1)! = 0) \&\& ((abs(s-g)) mod n! = 0))} \\ 0 & \text{other wise} \end{cases}$$

abs(x) is a function which takes an integer argument x which can be either positive or negative and returns a positive value of the argument

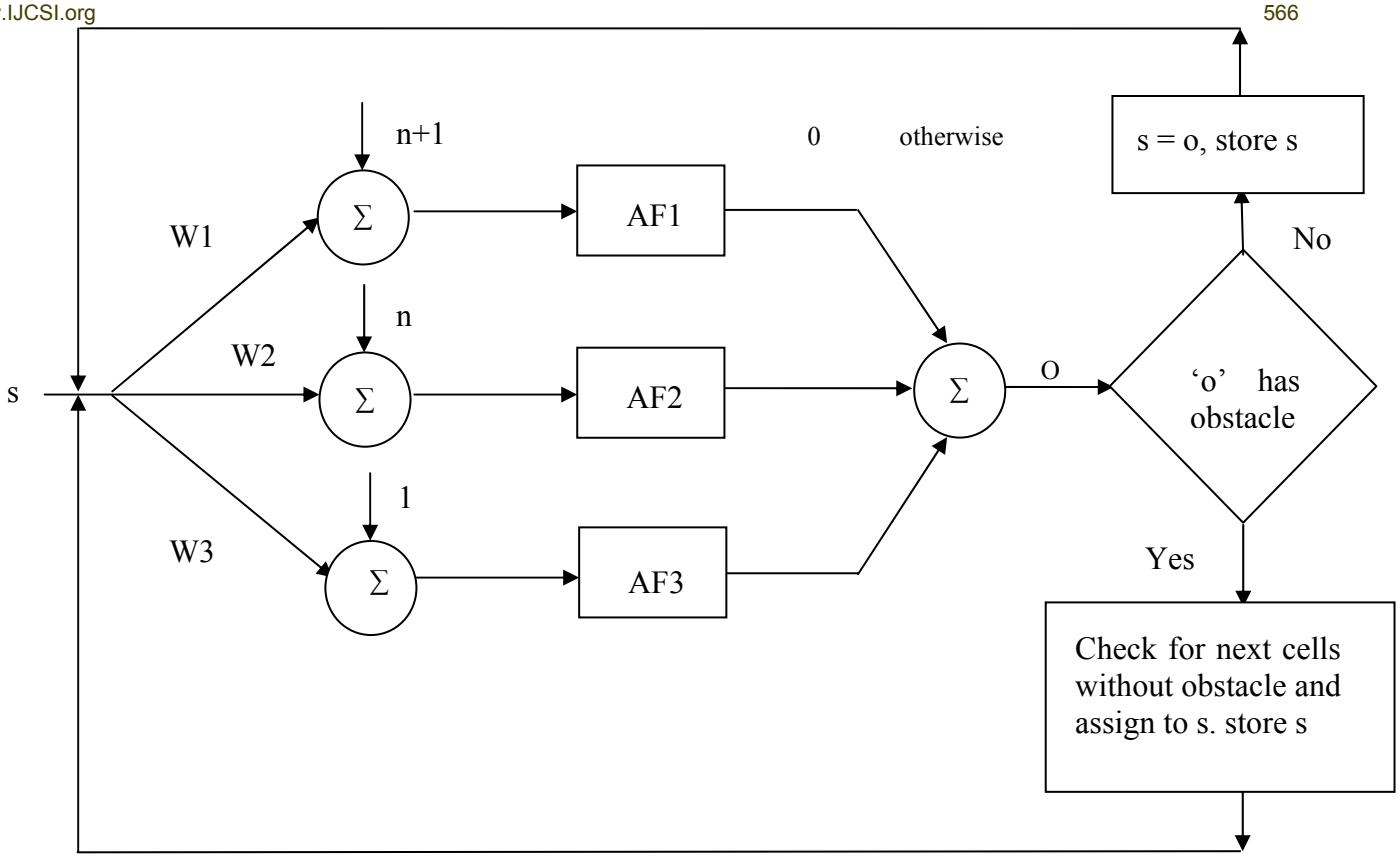



**Fig 2 parallel distributed neural network model**

where

s- Starting cell in first iteration and it is the next sequence of cells where the Actor has to
Move calculated in next iterations

g- It is the goal cell where the fire has occurred.

n- Number cells in a row or column of the decomposed forest domain.

w1, w2, w3 – weights connected to the neurons

AF1, AF2, AF3 are Activation functions for neurons 1, 2 and 3 respectively

o- Net output of the Activation functions summed together.

The output of the each neuron is calculated as follows:

$$\text{Output of neuron1} = \begin{cases} s+n+1 & \text{if } w1 = 1 \\ n+1 & \text{if } w1 = 0 \end{cases}$$

$$\text{Output of neuron2} = \begin{cases} s+n & \text{if } w2 = 1 \\ n & \text{if } w2 = 0 \end{cases}$$

$$\text{Output of neuron3} = \begin{cases} s+1 & \text{if } w3 = 1 \\ 1 & \text{if } w3 = 0 \end{cases}$$

The weight will vary from iteration to iteration due to change in value of s in each iteration. The cell g is always constant and it is not shown explicitly in the model. The model also uses 3 activation functions which are calculated as shown below:

$$AF1 = \begin{cases} s+n+1 & \text{if output of neuron1 is } s+n+1 \\ 0 & \text{otherwise} \end{cases}$$

$$AF2 = \begin{cases} s+n & \text{if output of neuron 2 is } s+n \\ 0 & \text{otherwise} \end{cases}$$

$$AF3 = \begin{cases} s+1 & \text{if output of neuron 3 is } s+1 \end{cases}$$

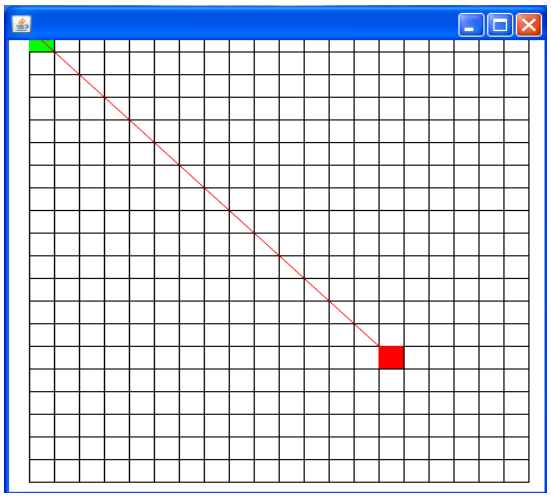
Initially the input value s is fed to the neural network. We assume that g is available as environment variable as it is constant. The weight will be calculated for each neuron separately. The output of the each neuron is fed as input to the activation function. The net value of three activation functions decides which move is selected. Then selected move is tested for the presence of obstacle. If there is no obstacle in the cell selected for next move then store the cell number and assign the cell number to s .If any obstacle is present in the cell selected for next move, then two cells obtained using remaining two movements will be checked for obstacles. If any one cell is free (definitely one cell will be free because of the assumptions of shape of obstacles) then store the cell number in the memory and assign the cell number to s.The process is repeated with new s till the cell number of s is same as cell g.The actor will use the sequence of cell numbers stored in the memory to reach the cell where the fire has occurred and extinguish it by a suitable means once the computation of path using the cell numbers is complete.

**6. Simulation results:**

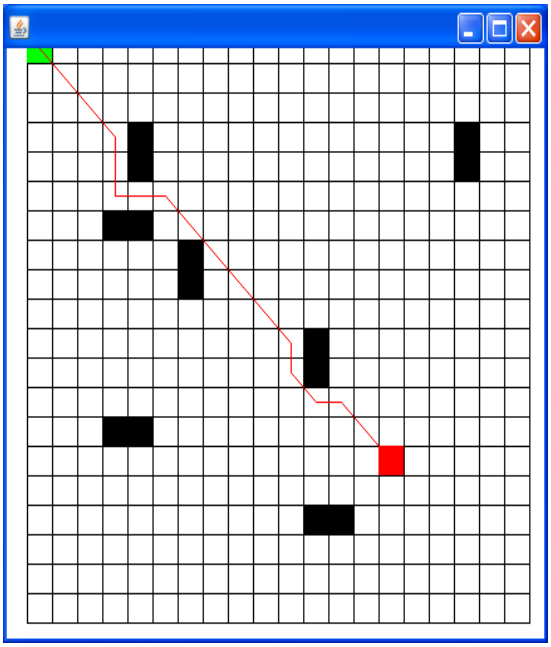
In this work we have considered the forest domain as a grid decomposed into m x n cells based on integer values. Also we have considered a single point of occurrence of fire since we have only one actor for the entire forest domain. We have devised a path planning algorithm for the actor based on parallel distributed neural network model in order to extinguish fire in both types of

environment. i.e. Environment with and without obstacles. We have used java for simulation purposes. We created a 20 x 20 grid. The actor is represented by a green square located at the top left corner, the cells containing obstacles are represented using black color and the cell where fire occurs is shown in red and a line in red color shows the path planning of the Actor to travel and reach the target area to extinguish fire. To test the efficacy of the proposed algorithm we have created fire in various cells and by varying the number of obstacles. The Computer simulation result shows that in all the cases sequence of cells which constitute a path is created using the algorithm by avoiding collision with the obstacle. The simulation results are shown below:

### 6.1Environment without obstacles



### 6.2 Environment with Obstacles



## 7. Conclusion

In this work a parallel distributed neural network Model based on reinforcement learning used by Actor for path planning to extinguish Forest fire is presented. We have considered single point of occurrence of fire with only one actor to extinguish fire. Also static obstacles with predefined shapes are only considered in this work. Simulation results show that in all cases path has been found successfully by avoiding collision with obstacles. In future we have plans to implement multiple Actors with multiple occurrences of fires and works related to dynamic obstacles and no constraints on the shape of the obstacle.

## 8. References

[1] James A. Freeman, David M.Skapura, "Neural Networks, Algorithms, Applications and Programming Techniques" Pearson Education Asia

[2] Isabelle Rivals, Daniel Canvas, Leon Peronnaz, Gerard Dreyfus "Modeling and Control of Mobile Robots and intelligent vehicles by neural networks" IEEE conference on intelligent Vehicles 1994.

[3] Ben J.A.Krose, Mark Eecen "A self organizing representation of sensor space for mobile robot navigation" International conference on Intelligent Robots and Systems IROS'94, 1994 pp 9-14.

[4] Michail G.Lagoudakis, Anthony S.Maida "Neural Maps for Mobile Robot Navigation" IEEE 1999, pp.2011-2016

[5] Joseph Djugash, Bradley Hammer "Neural Networks for Obstacle Avoidance" Informal Publication

[6] Danica Janlova "Neural Networks in Mobile Robot Motion" International Journal of Advanced Robotic Systems, Volume 1 Number 1, 2004, pp.15-22

[7] Alberto Yukinobu Hata, Denis Fernando Wolf, Gustavo Pessin, Fernando Osorio "Terrain mapping and Classification Using Neural Networks" Conference 2004 ACM

[8] Subhrajit Bhattacharya and Siddharth Talapatra "Robot Motion Planning Using Neural networks: A Modified

Theory" International Journal of Lateral Computing, vol.2, No.1, 2005, pp.9-13.
[9] Istvan Engedy, Gabor Horvath "Artificial Neural Network based Mobile Robot navigation" Informal Publication
[10] Amir Reza Saffari Azar Alamdari "Unknown Environment Representation for Mobile Robot Using Spiking Neural Networks" Proceedings of world Academy of science, Engineering and Technology Volume 6, 2005, pp.49-52
[11]Panrasee Ritthipravat, Kenji Nakayama "Obstacle Avoidance Using Modified Hopfield Neural Network" Informal Publication
[12] I.Engedy, G.Horvath "Artificial Neural Network based Local Motion Planning of a Wheeled Mobile Robot" Informal Publication
[13] Limin Fu "Neural Networks in Computer Intelligence" Tata Mc Graw-Hill Limited Edition 2003.
[14] Jacek M.Zurada "Introduction to Artificial Neural systems" Jaico Books
[15] M.P.Sivaram Kumar, S.Rajasekaran, "Detection and Extinguishing Forest Fires using Wireless Sensor and Actor Networks", International Journal of Computer Application, New York, USA, Vol.24 (1), June 2011, pp.31-35.

**About the Authors**

**M.P.Sivaram Kumar** Graduated with a Bachelor degree in Computer science and Engineering in 1998 from University of Madras. He then obtained his Post graduate degree in Computer Science and Engineering from Madurai Kamaraj University in 2002.Currently working as Assistant Professor in Computer science and Engineering Department of DMI college of Engineering, pursues part time PhD in B S Abdur Rahman University, Chennai-600048. His areas of Research includes Wireless sensor and Actor Networks, Robotics, Data structures, Algorithms and Theory of Computation.

**Dr.S.Rajasekaran** received Predoctoral degree in Mathematics from Madurai Kamaraj University in 1981.He obtained his Doctororate degree from Indian Institute of Technology, Kanpur in 1986.He has authored more than 20 papers in peer reviewed international journals. Currently working as Professor of Mathematics in B S Abdur Rahman University, Chennai-600048.His areas of Research include Modeling, Artificial Intelligence, Algorithms and Theory of Computation.