3. METHODOLOGY

3.1 OVERALL METHODOLOGY

The overall Research methodology is shown below in figure 3.1.1

- **Forest Area**
  - Superimposed over a grid
  - Grid is decomposed into cells of equal size
    - **Sensor** which knows its location based on coordinates is deployed one in each cell
      - Temperature inside a cell is greater than a certain threshold
        - Senses/Measures
          - Sent message in form of coordinates
            - **Actor** is deployed in one cell which knows its location
              - The **developed path planning algorithms** (available in actor) uses the coordinates of location of the actor and the location of the cell in which fire occurred to find the path
                - Actor moves through the path and extinguishes fire
The forest area is superimposed on grid. The grid is decomposed into m x n cells of equal size. In each cell a sensor is deployed deterministically which knows its location based on coordinate information. The actor then using its own coordinate as starting point and the location of the cell in which fire occurred as ending point and uses the developed path planning algorithms through which it can navigate and extinguish fire. An actor which knows its location is deployed in one cell. When the temperature inside a cell goes beyond the particular temperature indicating fire, then the sensor inside the cell will send the message in the form of coordinates to the actor.

3.2 TESTING:

Testing is a process of executing a program with the interest of finding an error. Testing should systematically uncover different classes of errors in a minimum amount of time with minimum amount of efforts. In this research work the developed path planning algorithms are tested to uncover any defects in the algorithm by subjecting it properly to white box and black box testing. This can be done by internal evaluation of code and developing test cases which meets all the requirements and executing the test cases. The test cases are developed both for the environment with obstacle and without obstacle.

The test cases developed for environment without obstacle is shown below
1. Both the starting and ending points lie within the grid and produced the path correctly.
2. The ending point lies out of the grid.

The test cases developed for environment with obstacle is shown below
1. Both the starting and ending points lie within the grid and produced the
path correctly.

2. The ending point lies out of the grid.
3. By generating and varying the number of obstacles along the path.
4. By generating obstacles around the robot so that it cannot able to find a path and displays the message no path exists.

In general the developed algorithms can work for any m X n grid. But the path planning algorithms developed uses 5 x 5, 10 x 10 and 20 x 20 grids as examples. The 5 x 5 grid is used as example because some of the works [119,120] used 5 x 5 grids to explain the concepts. The 10 x 10 and 20 x 20 grids are used as examples in some of the works [120,121] for execution purposes. All the algorithms are executed for 50,75,100 times respectively by satisfying all the test cases without replication of the input. The reason for choosing the number of times of execution is based on [122] because for 100 x 100 grids 500 sample runs were performed. The 100 x 100 grid will contain 10000 cells. For 10000 cells 500 sample runs were performed. The 20 x 20 grid will contain 400 cells. The number of cells to number of runs ratio (i.e. for 10000 cells 500 runs, for 400 cells 20 runs) we will get 20 sample runs for 20 x 20. But in this research work for 20 x 20 grids 100 runs are performed indicating that the algorithm is verified and validated for more number of inputs. Since the 5 x 5 and 10 x 10 grids use 50 and 75 runs respectively, the algorithm is verified and validated for more number of inputs as compared to [122]. The algorithms are implemented using variety of softwares such as Adobe flash, Java, C++ and MATLAB. The graphs are plotted using MS-Excel. Finally a comparison of developed algorithms with A* algorithm is performed in terms of execution and path cost.