**Project Report**

**Problem:**

Forest fires are unpredictable and highly destructive forces of nature. They cause a lot of ecological and environmental imbalance in nature. It is very important to determine the effects of forest fires from the past occurrences. One approach to this problem is to learn from the past events and try to predict the future events to reduce the catastrophic effect of the problem and try to mitigate the impact of the forest fires.

**Proposed Solution:**

Data set of an area (Montesinho Natural Park) where forest fire has occurred is analyzed and based on this an artificial intelligent algorithm named particle swarm optimization is used. The main attributes that are taken into consideration for forest fires are as follows:

1. Humidity
2. Rainfall
3. Wind Speed
4. Thundering

The values of these attributes recorded every time the forest fire occurs in the area is as follows:

|  |  |
| --- | --- |
| Attributes | Values |
| Humidity | 15.0 – 100.0 |
| Rainfall(mm/m2) | 2.0-6.4 |
| Wind Speed(km/hr) | 1.4-9.4 |
| Thundering | 30-150 |

Based on these values the cost function of the particles is calculated.

**Cost Function:**

Cost= 0.2\* Rainfall+0.2\*Humidity+0.3\*Wind Speed+0.3\*Thundering

The algorithm is based on the movements of the particles which move in the search space to find the optimal solution of the problem at hand.

**Key Features:**

pBest: personal best cost of the particle

gBest: Global best cost of all the particles in space

localBest: Best cost in the present iteration

BestLocation: Best location of the particle.

**Initialization:**

Velocity: set to 0.

Position: Randomly allocated

pBest = Calculated from cost function.

**Algorithm is as follows:**

1. Initialize the particle’s velocity, initial position, pBest, and the best position till date.
2. Cost function is calculated based on the above attributes.
3. Once the initialization is done, calculate localbest of the particles.
4. If localbest > globalbest then replace the global best
5. Velocity of the particle is calculated based on the following formula:

newVel=w\*currVel+(r1\*c1)\*(Particle’s best position-Current position)+(r2\*c2)\*(Best position of Gbest-current position

1. Update location of the particle

newLocation=currLoc\*curVel

In velocity:

W=inertia

C1,C2: acceleration co-efficients

R1,R2: random generated values

This will be done for many iterations/epochs.

The following output was achieved:

Epoch 200:

Epochs 500:

Epoch 1000:

Epoch 2000:

Graphs:

