Tsunami Epicenter Detection Using PSO

Abstract:

The project is based on the implementation of the Particle Swarm Optimization algorithm, which deals with developing a solution to mitigate the consequences that arise when there appears to be a surge in seismic activity in an ocean, sea or any other water body.

Introduction:

Particle swarm optimization is one of the mutative computational techniques which is inspired by social behavior of bird flocking or fish schooling. In this algorithm, the particles represent the possible solutions to the final goal. Each particle in the swarm has a position vector $x_i(t)$ (i = 1, 2, . . . , N) and a velocity vector $v_i(t)$ in the search space at time t. The parameter N denotes the total number of particles in the swarm. Each particle has memory and hence can keep a track of its trail. The fitness of the particle at i^{th} position is estimated by the objective function or the fitness function f(xi(t)). The position at which the particle has the best value for the fitness function is known as the personal best position denoted by $x_i^p(t)$ and the best value amongst all particles in the swarm is known as global best position denoted by $x^g(t)$.

The velocity and position of the particle is updated iteratively based on previous position, inertia weight and acceleration coefficients.

Implementation:

In the project the amphibious bots used to intercept the seismic waves act as the particles. Every bot communicates with the bots in its locality to adjust and direct its search accordingly in the direction of epicenter. Every bot calculates its fitness function to find the epicenter i.e. the destination in PSO. The fitness function basically calculates the Euclidean distance from the particle to the point of maximum seismic activity occurrence. This process iteratively reduces the region of search for epicenter detection by finding the bot returning the most optimal fitness value for seismic activity and setting it as an intermediate center before zeroing down on the actual epicenter.

Algorithm:

Step1:

Initialize the swarm by setting their location and velocity which is randomly generated within the Sample space (X = [-4, 4] ; Y = [-4, 4])

Step 2:

Update the fitness value for each particle with respect to its current location and using the fitness function (Euclidian distance)

Step 3:

a) Compare the fitness value and the personal best of each particle and if the fitness value is less than the personal best then update the pBest (personal best) for each particle
b) Get the minimum fitness value from fitness value list and if the value is less than the global best and update the gBest (global best) to the new value

Step 4:

Calculate the new velocity using the PSO Equation:

$v_i(t + 1) = w * v_i(t) + c_1 * r_1 * [pBest_i(t) - x_i(t)] + c_2 * r_2 * [gBest(t) - x_i(t)]$

Step 5:

Calculate the new location using the PSO Equation:

$x_i(t + 1) = x_i(t) + v_i(t + 1)$

Step 6:

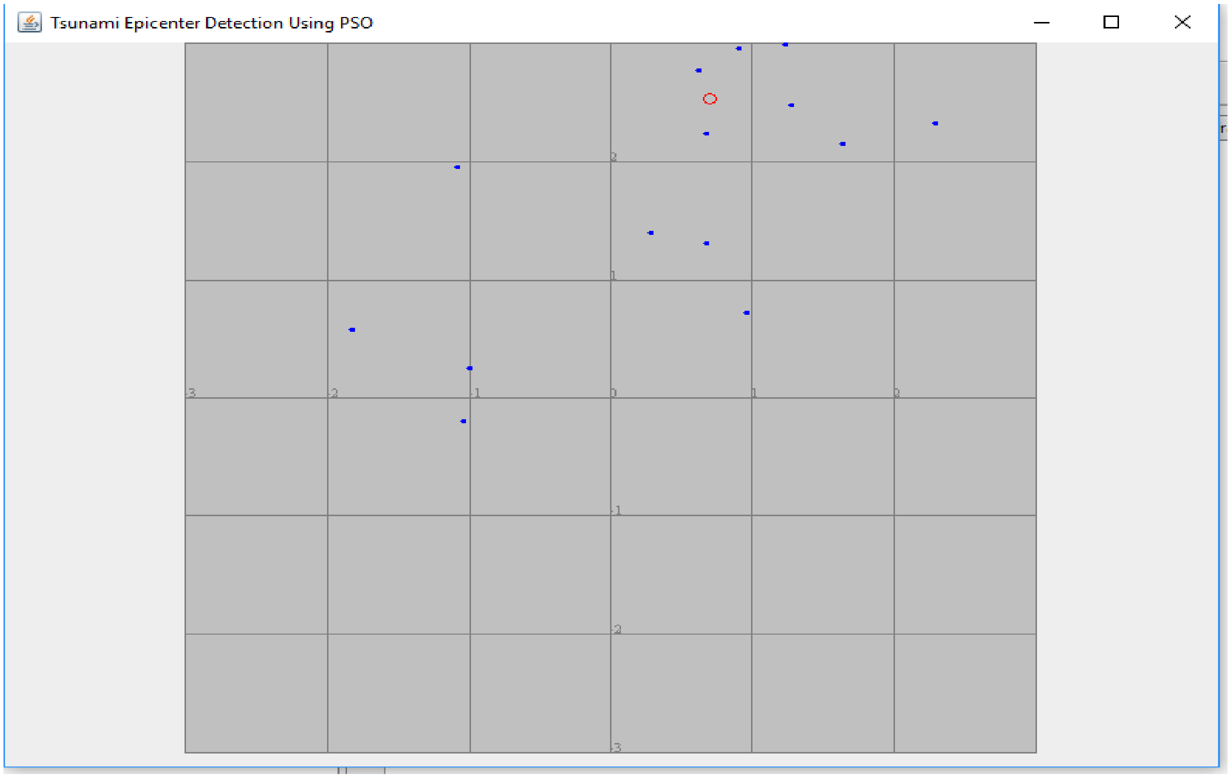
Update the fitness value as per the new location

Step 7:

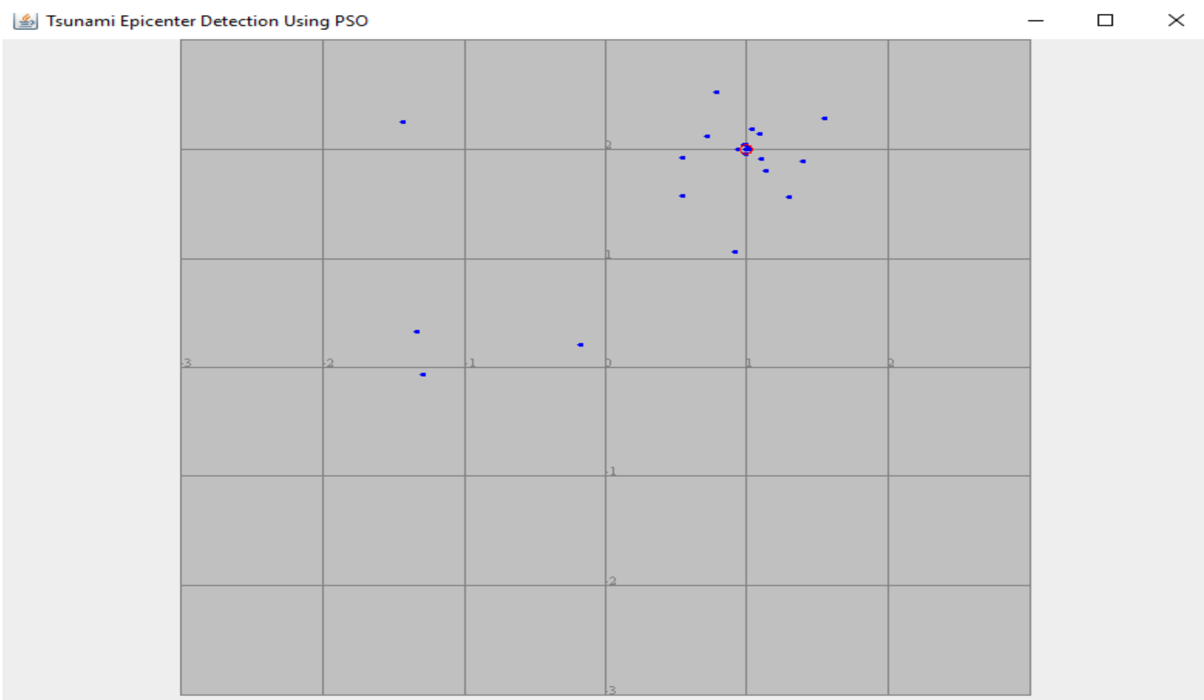
If the number of iterations is reached or the optimum criteria is met, then terminate the loop or else go back to Step 3.

Screenshots:

During 1st iteration



During 35th iteration



During 75th iteration (bots converge to epicenter)