***PSO AND ROBOT PATH PLANNING PROBLEM***

As a particle move through the search space, gaining one new position for

iteration, a conditional statement checks to see if the position of the particle will fall

within the boundaries of the obstacle. If this condition is true, the obstacle avoidance

section of the algorithm is initiated.

Movement of a robot position is realized by the Particle Swarm Optimization

algorithm. PSO convergence to the best solution by adjusting the trail of each

individual particle toward its best location based on the best of itself and global best

on the neighbor particles. The modification of a robot position is realized by position

and velocity information. The position of a robot is represented by Cartesian

coordinate, such as x, y axis position and the velocity is modification by PSO.

The initial position for the robot is initialized with random coordinates. Robot is

defined within the context of topological obstacles comprising itself in the

population. Neighbors of robot can be carried out based on the distance of robot

position. In this study, global path is considered for robot movement path planning.

Global path is carried out based on the fitness function. Fitness value is evaluated for

each particle over its obstacles based on the coverage of the target area. When a

particle discovers a pattern that is better than any it has found is stores the coordinate

as new best position. Robot rotates in order to reduce the possible collisions with

obstacles.

**SOLVING THE CONSTRAINT PATH PLANNING PROBLEM ROBOT USING PSO**

In this section we propose a solution to path planning problem using PSO. The

proposed scheme presumes current position of robot and their speeds, and determines

the next position of robot by optimizing the given constrained objective function.

We propse an algorithm for trajectory of robot using PSO.

*1.* Add current position of robot to the trajectory.

*2.* If the distance between the current position of the robot and the goal is less

than or equal to a predefined threshold, go to f.

*3.* Relocate the obstacles and the goal according to their corresponding relocation

probability.

*4.* Initialize PSO swarm around current position of the robot. Evolve PSO swarm

according to Algorithm 1.

*5.* Choose global best position the swarm to be the current position of the robot

and go back to a.

*6.* Add the goal position to the trajectory and stop.