Rescue -Bots

Abstract: According to statistics, the survival rate of victims drops drastically during large scale disaster. Unfortunately, the resource and equipment's are limited for rescuers during such situation. This increases the death rate owing to lack of clarity towards the safe exit and guidance towards the same. For this reason, there is always a need for commander to guide the victims towards a safe exit. This methodology is used to gather people towards the safe exit in the big organization manually using a floor warden. But in case of trivial situation where human intervention is not possible the situation goes out of control. Hence a significant need arises towards addressing this above scenario. A vital solution towards such scenario is to make use of the autonomous robots there by reducing human effort and death counts. This project briefly gives an overview and implementation about how to tackle the problem with best algorithm and good precision.

Introduction

Mobile Robotics gaining more popularity these days. They are being used in various fields of study. Currently there are two major studies namely semi-automatic space exploration and underwater exploration. One of such application would be here is to utilize autonomous robots to help people guide at times of natural calamities where human intervention is not safe. Basically addressing these project to make bots as rescue soldier. These activities have cent percent acceptance because of robots Robustness, Versatility, rough Terrain Navigation. As per one article [1] the first project that focused on robotics for SAR is Swarm-Bot [2]. The similar concepts are being discussed below in this project to identify the exit location in a building that is under emergencies or under serious collapse where humans get trapped without knowing the exit. The problem is addressed using a group of Robots to guide the people to find the near and safe exit to get humans out of the building as a result of rescue operations.

Approach

Swarm Given the current state-of-the-art in autonomous robotics, rescue operations can be carried out using a swarm of robots guiding towards a safer environment for the victims. A swarm robotics technique (PSO algorithm) can be utilized to identify the safe exit using a swarm of heterogeneous robots, the bots communicate with each other and change its position locally and giving a global solution towards a guided drone bot(Target). The basic idea is that each bot searches for the optimum. Each bot will move and hence has a velocity. The works like a dynamic programing technique where the bot stores the position it travels and where it has to move forward knowing the bots best position Each bot has a neighborhood associated with it and each bot knows the finesses of those in its neighborhood, and uses the position of the one with best fitness. This position is simply used to adjust the particle’s velocity and move towards the targeted location which is the Safe Exit.

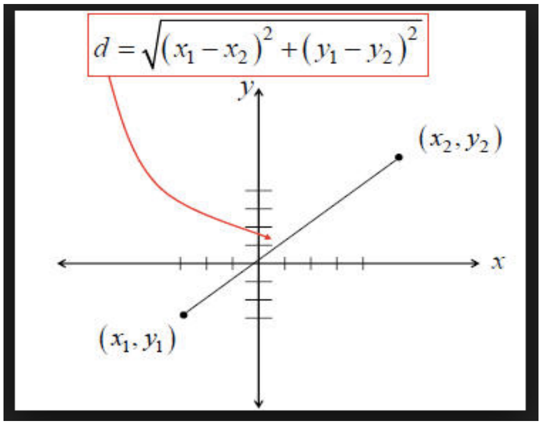
Algorithm

The base algorithm expended here is by Particle Swarm Optimization. The Particle swarm optimization is quite famous and important algorithm developed by James Kennedy and Russell Eberhart. Recently, PSO is quite promising algorithm for various optimization real world problems. This algorithm is derived from social and co-operative behavior shown by various species which are in need of search for something. This algorithm is designed with respective to Personal best experience (Local Best) and overall experience (Global Best) and the present motion of the particle to get the next position in their way of search. The motion or experience are accelerated by two factors C1 and C2 Their current motion is multiplied by an inertia factor w varying between [w_{Min} ,,w_{Max}].

In this section below we will concentrate on the problem specific algorithm towards this project within our scope. First we define the characteristics of a robot with the attributes of Position (x-component, y-component), Speed (Minimum, Maximum). We initialize the swarm population of fixed size (Active-Bots) denoted as swarm = [R1, R2,……., R40]. Each Particle is given an initial position and speed denoted by Robot Speed = [S1, S2,….,S40] and Robot Position = [P1,P2,….P40].

Set Parameter $w_{Min,}$ w_{Max}, C1 and C2 for the defined System. Initialize the robots position and speed. calculate the fitness for each particle. The fitness of a particle is calculated as test function. The Objective function defined for this problem statement here is the distance between the particles current position and the target position co-ordinates. The is shown below. The objective function is the one that is desired to be maximize or minimize.

$$fitnessValue = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \cong 0 \ \ (Minimizing\ Function)$$



[3]: Image Reference

Calculate the fitness of each robots and find the best fitness among the group and get the index (x-component, y-component) of the best robot. Get the Local Best and Global Best fitness values and c-ordinates. Calculate the inertia factor by

$$w = wMax - IterationCount * \frac{wMax - wMin}{MaximumIteration}$$

Based on the factors the **new speed** for the particle changes as per the below equation.

$$v_{min} = w * v_{old} + c_1 r_1 (Pbest_x - Pold_x) + \ c_1 r_1 (Gbest_x - Pold_x)$$

$$v_{max} = w * v_{old} + c_1 r_1 \left(Pbest_y - Pold_y\right) + \ c_1 r_1 \left(Gbest_y - Pold_y\right)$$

once getting the new speed, calculate the new position for each robot and update the swarm's new positions based on the below function.

$$Pnew_x = (Pold_x) + \ v_{min}$$

$$Pnew_x = (Pold_x) + \ v_{min}$$

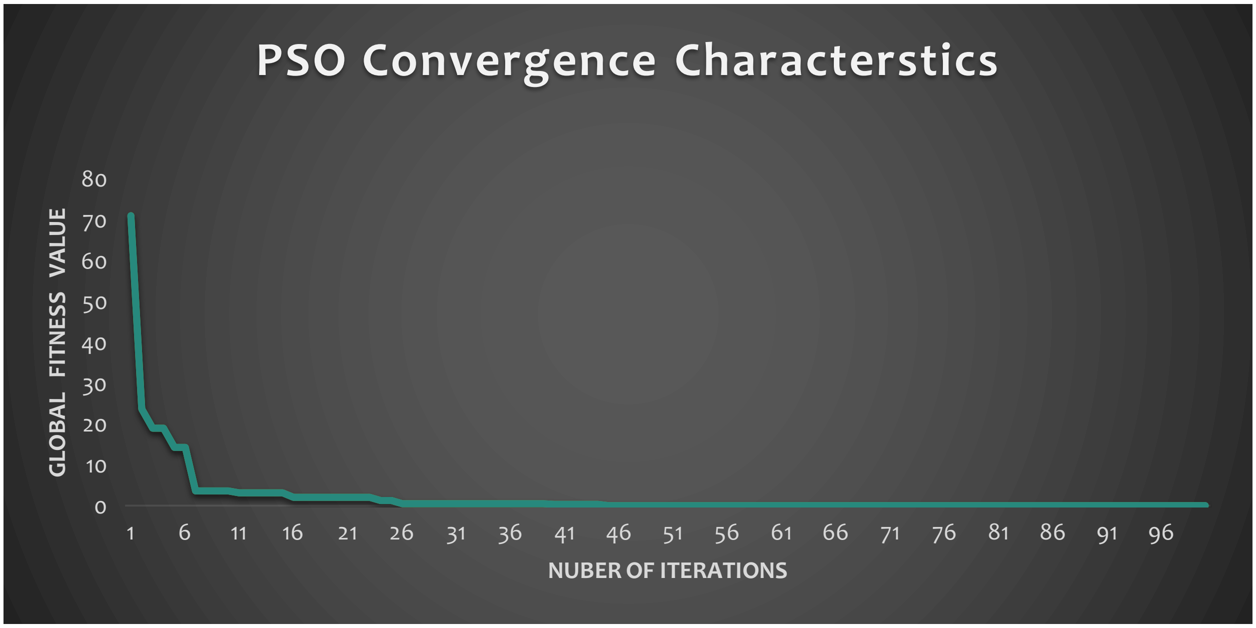
Update the fitness value with the newly derived speed and positions. And iterate the loop till the maximum iteration is reached or the error tolerance is reached. There can be one or more stopping criteria to stop these iterations as shown below [3.2]

Stopping Option	Stopping Test	Exit Flag
Maximum Iterations	Number of iterations reaches maximum Iteration	0
Max Stall Time	Best objective function value did not change in the last motioned seconds	4
Max Time	Function run time exceeds Max Time seconds	5
Error tolerance	Loop reaching the maximum error Tolerance	2

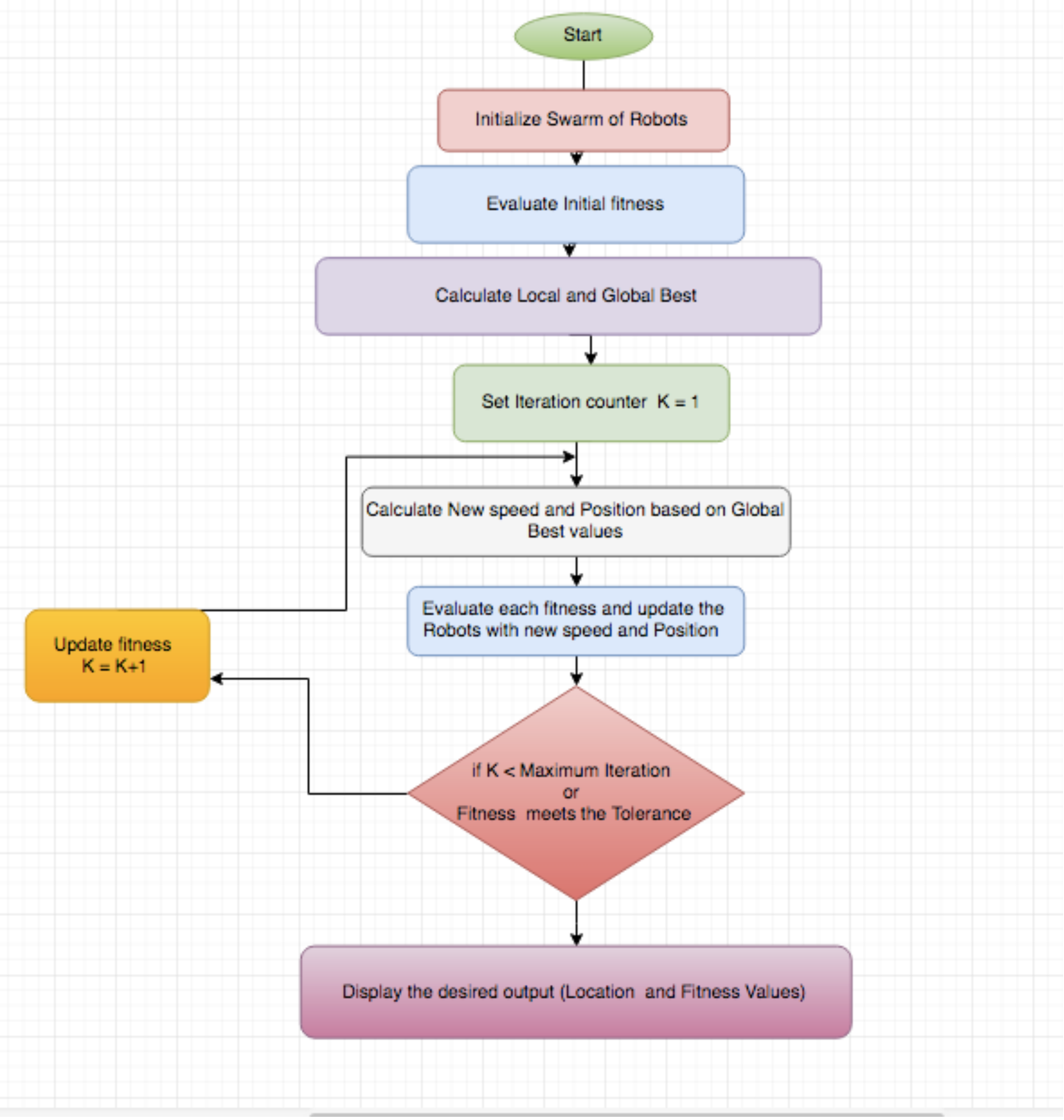
Update the fitness value with the newly derived speed and positions. And iterate the loop till the maximum iteration is reached or the error tolerance is reached.

Analysis

Below is the quick analysis of the robot's convergence characteristics



Flow Chart


Start
Initialize Swarm of Robots
Evaluate Initial fitness
Calculate Local and Global Best
Set Iteration counter K = 1
Calculate New speed and Position based on Global Best values
Evaluate each fitness and update the Robots with new speed and Position
Update fitness K = K+1
if K < Maximum Iteration or Fitness meets the Tolerance
Display the desired output (Location and Fitness Values)

Conclusion

In this report the concept of PSO to achieve rescue robots have been discussed. Also the algorithm to achieve the same have been elaborated. The autonomous robots can be utilized to solve the issue of human non-intervention sites to rescue people using PSO algorithm. The robots communicate with each other defined within a bounded speed limit achieves the track to move to the futuristic position and the system as such remembers the track of all best position at each point to command the robots to move accordingly. This technique always has scope for future improvements and enhancements. The detailed code for the above scenario based can be requested from Punniyamoorthy.v@husky.neu.edu.

Reference:

[1] Search and Rescue using Swarm Robots Complex Systems Seminar Mahmood Rahmani 770810-6435 rahmani@student.chalmers.se

[2] Dorigo, M., et al. 2004, Swarm-Bot: A New Distributed Robotic Concept, Autonomous Robotics 17, 193-221.

[3.1] https://www.google.com/search

[3.2] https://www.google.com/search/MAthworks