

Exercise 1

Consider an illustrative example of a particle swarm optimization system composed of three particles and $V_{\max} = 10$. To facilitate calculation, we will ignore the fact that r_1 and r_2 are random numbers and fix them to 0.5 for this exercise. The space of solutions is the two-dimensional real valued space R^2 and the current state of the swarm is as follows:

Position of particles: $x_1 = (5,5)$; $x_2 = (8,3)$; $x_3 = (6,7)$;

Individual best positions: $pbest1 = (5,5)$; $pbest2 = (7,3)$; $pbest3 = (5,6)$;

Social best position: $gbest = (5,5)$;

Velocities: $v_1 = (2,2)$; $v_2 = (3,3)$; $v_3 = (4,4)$.

Please answer the following questions:

1. What would be the next position of each particle after one iteration of the PSO algorithm using inertia $\omega = 1$?
2. And using $\omega = 0.1$?
3. Explain why the parameter ω is called inertia.
4. Give an advantage and a disadvantage of a high inertia value.

Exercise 2

1. Appliquer la méthode PSO à l'exemple simple $\text{Min } F(x) = x^2$, $x \in [-5; 5]$ et avec $c_1 = c_2 = c_3 = 0.7$.
(On demande d'écrire l'algorithme détaillé)
2. Tracer les trajectoires correspondantes des particules sur la parabole.
3. Appliquer la méthode PSO au cas de la fonction de Rastrigin sur $[-5; 5]^2$:
 $Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2)$ avec 10 particules. Qu'observe-t-on?

Exercise 3

Réécrire le pseudocode de l'algorithme PSO et discuter sa complexité.

Exercise 4

Afin d'appliquer PSO au TSP(n) : interpréter :

- Une particule ;
- Une position d'une particule ;
- Une vitesse d'une particule ;
- Une différence de 2 positions ;
- Le produit d'un scalaire par une vitesse ;
- Une somme de 2 vitesses ;
- Une somme d'une position et d'une vitesse ;
- Donner des exemples.

Même questions pour KSP(n) et SAT-3(n).

Exercise 5

A Particle Swarm Optimization (PSO) is applied to the search of a hidden RF transmitter. The search area is a square stretched out between coordinates $(-100, -100)$ and $(100, 100)$. The amplitude of a sampled signal as a function of distance from the emitter is given by:

$$A = \frac{1}{4\pi r^2} + kN(0, \sigma)$$

where A is the measured amplitude, r is distance between transmitter and the sampling location and N is a Gaussian noise distribution with zero mean and σ standard deviation, k is a parameter for adjusting the relative noise level.

- a. Explain the canonical PSO.
- b. Given 4 particles in a PSO with positions:

$$\begin{aligned}x_1 &= (10, 10) \\x_2 &= (12, 8) \\x_3 &= (11, -10) \\x_4 &= (-4, 9)\end{aligned}$$

Calculate an iteration of particle 1 assuming $\omega = 0.98, \omega_1 = 0.04, \omega_2 = 0.02$ and simulate the required probabilities. Also, assume that the position of the hidden emitter is $(0, 0)$ and that $k = 0$, i.e. a noise free system.

- c. Simulate the next iterations of this PSO problem by altering the the NetLogo version of PSO, found under:
‘File->Models Library->Sample Models->Computer Science->Particle Swarm Optimization’.
Remark, the UpdateParticleVelocity (the ‘to go’ function) in the NetLogo program is altered.
Also, you could use new random initial position and velocities for the particles.
- d. Release the 4 particles from $(-75, -75)$ plus some randomness. How does this affect the optimization?
- e. What happen if you add noise to the system? You could set $k = 0,0001$. Compare the two different initial positions of the particles. Would you use PSO in a real swarm robotic system where the mobile robots are released from same location?
- f. What could be alternative approaches for locating the hidden RF emitter using swarm robotics?
- g. Optional:
Play with different PSOs, parameters and possibly other swarm algorithms on this problem.