Metaheuristics

Particle Swarm Optimization

Kennedy and Eberhart proposed Particle Swarm Optimization in 1995. It is inspired by the movement of a flock of birds when searching for food.

Each particle i represents a solution for the problem. In the time t, it has a position $x^i(t) \in \mathbb{R}^d$ and a velocity $v^i \in \mathbb{R}^d$.

$$x^i(t) = < x_1^i, x_2^i, \ldots, x_d^i >$$

The positions and velocities are updated following the next equations, where P_{best}^i is the best position where the particle i has been, G_{best} is the best location founded until the moment, r_1 *and* r_2 are random numbers between 0 and 1, and $w, c_1,$ *and* c_2 are hyper parameters. Those last values can be initialized at 0.9 and gradually reducing it until 0.1.

$$v^i(t+1) = w\, v^i(t) + c_1 r_1\left(P_{best}^i - x^i(t)\right) + c_2 r_2\left(G_{best} - x^i(t)\right)$$
$$x^i(t+1) = x^i(t) + v^i(t+1)$$

PSO Algorithm

Parameters:
  N, Number of particles
  MaxIter, Maximum number of iterations
  func, objective function
  bounds, the search-space
Return: the best position G_{best}

Begin
  Initialize c_1, c_2, w
  Create the particles positions and velocities randomly
  Calculate the objective function values
  Calculate P_{best}^i as the current positions
  Calculate G_{best}
  While t < MaxIter or we haven't found a good solution
    For each particle i
      Update the velocity:
$$v^i(t+1) = w\, v^i(t) + c_1 r_1\left(P_{best}^i - x^i(t)\right) + c_2 r_2\left(G_{best} - x^i(t)\right)$$
      Update the position:
$$x^i(t+1) = x^i(t) + v^i(t+1)$$
      Calculate $func(x^i)$
      If $f(x^i) < func(P_{best}^i)$: update P_{best}^i
      If $f(x^i) < func(G_{best})$: update G_{best}
    End for
  Decrease c_1, c_2, w
  End while
End
Return G_{best}