Rossey Charleston
Vijaya Kumar V
I1800
Dec 23, 2013

Finding minimum optimum point in the Rosenbrock function using Particle Swarm Optimization technique.

Rosenbrock Function is represented by :
f(x,y) = (1-x)^2 + 100 * (y-x*x)^2

We use PSO to find the minimum value of variable x and y which makes the Rosenbrock function minimum. The particle has the following characteristics:

1)position – The current position of the particle.
2)fitness – Fitness of the particle.
3)velocity- Velocity of the particle.
4)bestPosition – Best position from group of particle.
5)bestFitness – Best Fitness in a group of particle.

Initially for our program we take 10 particles and assign them a random velocity and random position. The velocity and position are assigned in such a way that it does not exceed the limits of Rosenbrock function , which is from -10 to 10. The particles which are at random position in a parabolic shaped flat valley tries to adjust their position and velocity based on the global best position and global best fitness. The new velocity is calculated based on a formula :

NewVelocity = (w * currParticle velocity) +
(c1 * r1* (currParticle bestposition – currParticle position)) +
(c2 * r2 * (bestGlobalPosition – currParticle position))

NewPosition = currParticle position + NewVelocity

Where ,
w – Inertia Weight.
c1 – Cognitive weight.
c2 – Social weight
r1 , r2 – Random Numbers.

We set the value for *w*, the inertia weight, to 0.729. This value was recommended by a research paper that investigated the effects of various PSO parameter values on a set of benchmark minimization problems. Instead of a single, constant value for *w*, an alternative approach is to vary the value of *w*. For example, if the PSO algorithm is set to iterate 10,000 times, you could initially set *w* to 0.90 and gradually decrease *w* to 0.40 by reducing w by 0.10 after every 2,000 iterations. The idea of a dynamic *w* is that early in the algorithm you want to explore larger changes in position, but later on you want smaller particle movements. Then we set the values for both $c1$and $c2$, the cognitive and social weights, to 1.49445. Again, this value was recommended by a research study. The c1 weight can be set as larger than c2 , in that case we place more weight on a particle's best known position than on the swarm's global best known position , and vice versa. The random variables $r1$ and $r2$ add a random component to the PSO algorithm and help prevent the algorithm from getting stuck at a non-optimal local minimum or maximum solution. We allow this to run for 1200 iterations and eventually at the end we arrive at a global optimum solution at x=1.0 and y=1.0 which is f(x,y) = 0.0.

Below is an output showing the global optimum achieved for Rosenbrock function:

```
Initial Stage:
Initial Global Fitness
62628.985698838624

Initial Global Position:
x:4.14
y:-7.81
------------------------------------
Rosenbrock Function:
f(x,y) = (1-x)^2 + 100 * (y-x*x)^2
------------------------------------
Global Minimum found at
x = 1.0
y = 1.0
Best Global Fitsness :0.0
```

Reference - http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=00870279