Particle Swarm Optimization

Akash Patil

Problem Definition

- A simple example - given function
  - Find the maximum and position of maximum for 25 - (x - 05)^2
- Fitting a polynomial through a given data
  - Data for (x,y) given for some number of points
  - Find the coefficients a,b,c,… for polynomial of type a + b*x^2 + c*x^3 …

Basics of Particle Swarm Optimization

- Generate initial population randomly
- Each particle is searching for its optimum
- Each particle remembers its own personal best
- Each particle is moving and so has a velocity associated with it
- Velocity is has 2 main components –
  - Towards its pbest
  - Towards the gbest in the swarm
- v = v + c1*rand()*(pbest_position – current_position) + c2*rand()*(gbest_position – current_position)

Maximum of a given function

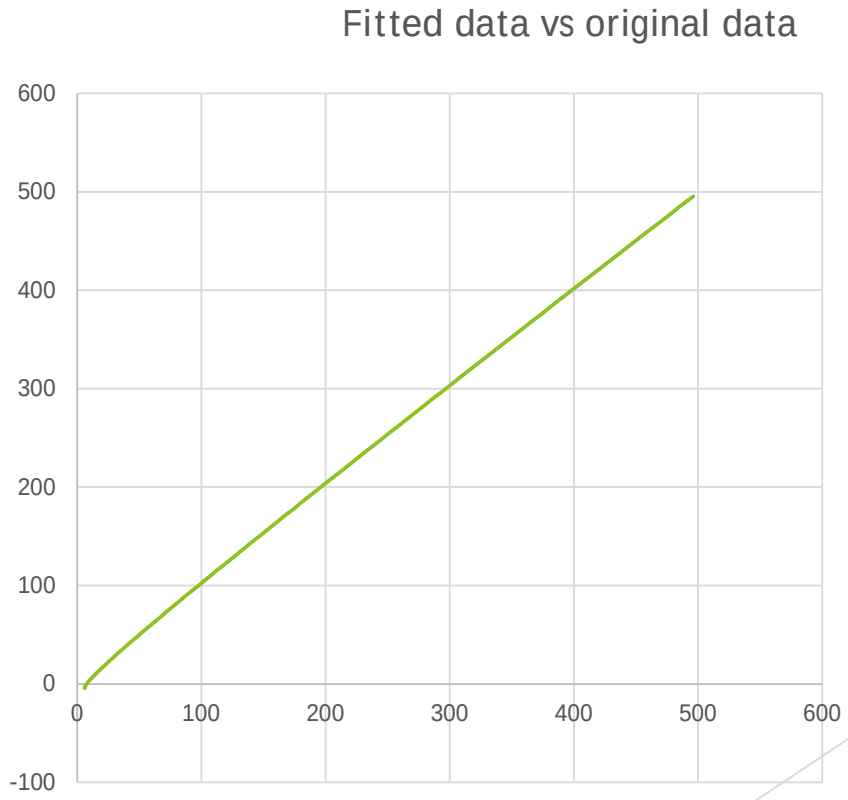
- Function chosen y = 25 - (x - 5)^2
- 10 particles
- 200 iterations
- Converges to (5,25)
- Convergence depends on values of C1 and C2
- Too low, particles can’t move much, too high, particles move too fast

Fitting a polynomial through given data

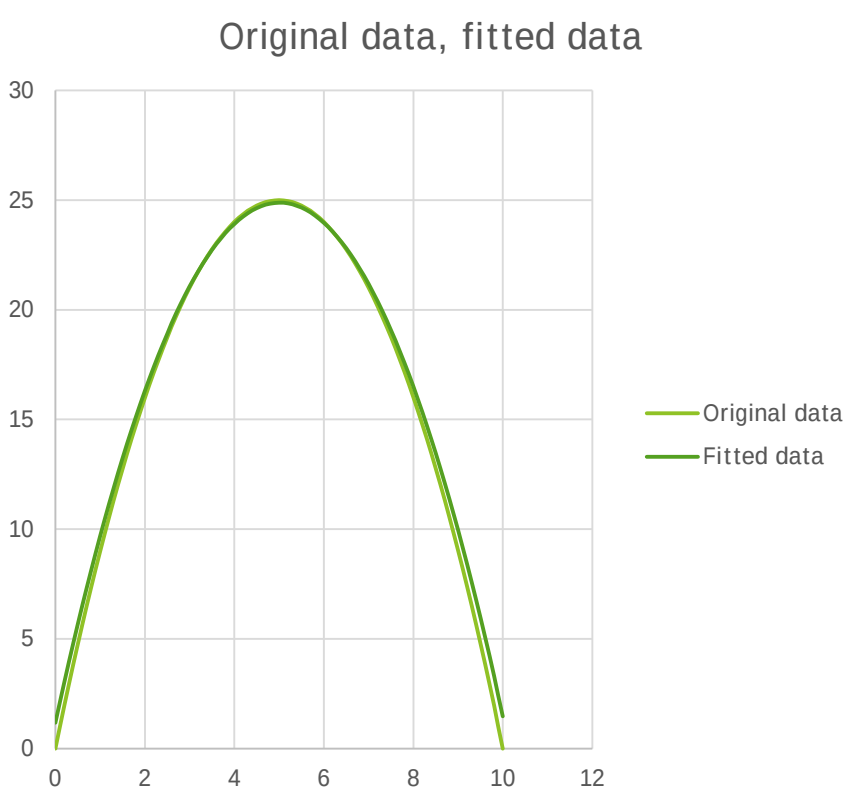
- Generated 2 arrays – x and y, with y = f(x) + rand()
- In this case, a particle would be a tuple of (a,b,c..)
- Can be considered to be a PSO in multiple dimensions
- The matlab function returns an array of coefficients of the fitted polynomial
- Given function -> 6 – x +5*x^2
  - output function -> -4.4168 + 3.427*x + 4.6557*x^2
- Given function -> x*(10 - x)
  - output function -> 1.1814+ 9.451*x + -0.9422*x^2
- Given function -> (x-1)*(x-4)*(x-8)
  - output function -> -3.7254 + 2.1919*x + -2.7297*x^2 + 0.2972*x^3

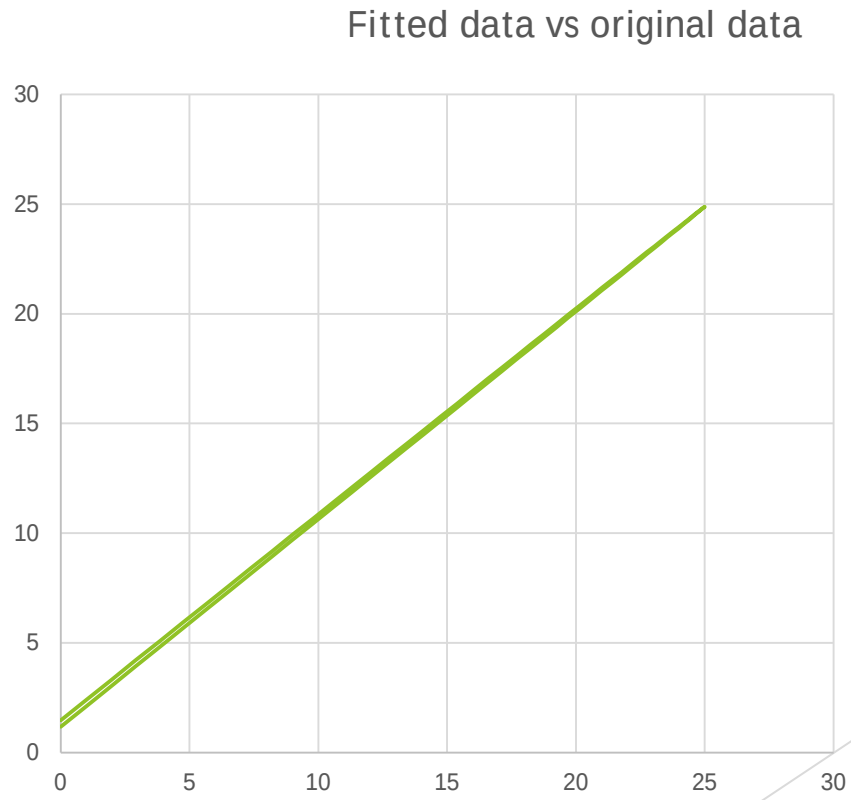
Plots 6 – x +5*x^2


Original data, fitted data
600
500
400
300
200
100
0
-100
0
2
4
6
8
10
12
Original data
Fitted data

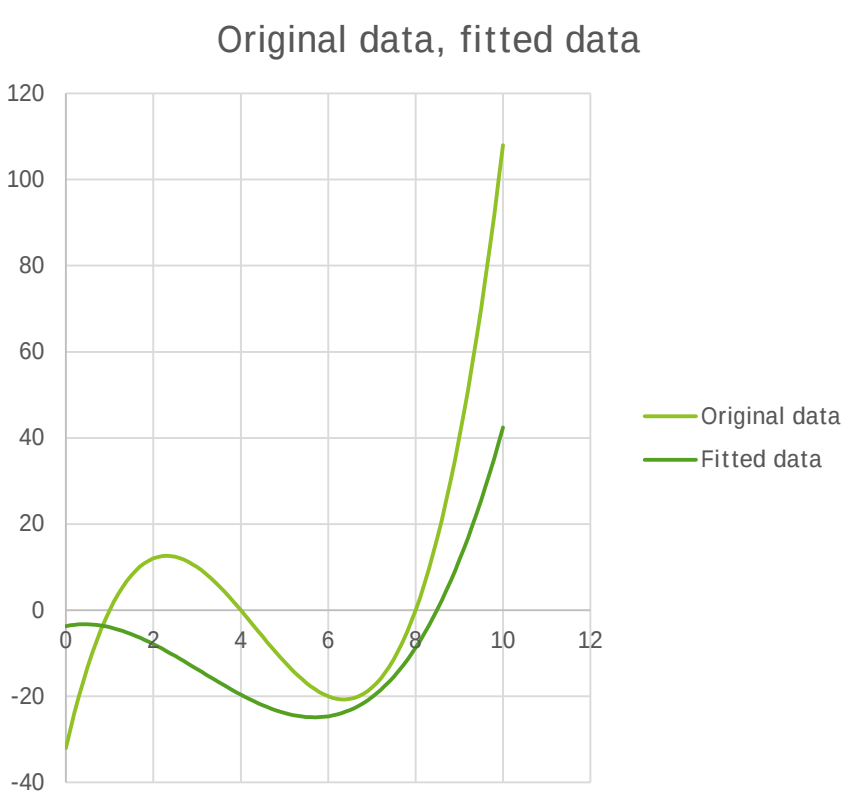

Fitted data vs original data
600
500
400
300
200
100
0
-100
0
100
200
300
400
500
600

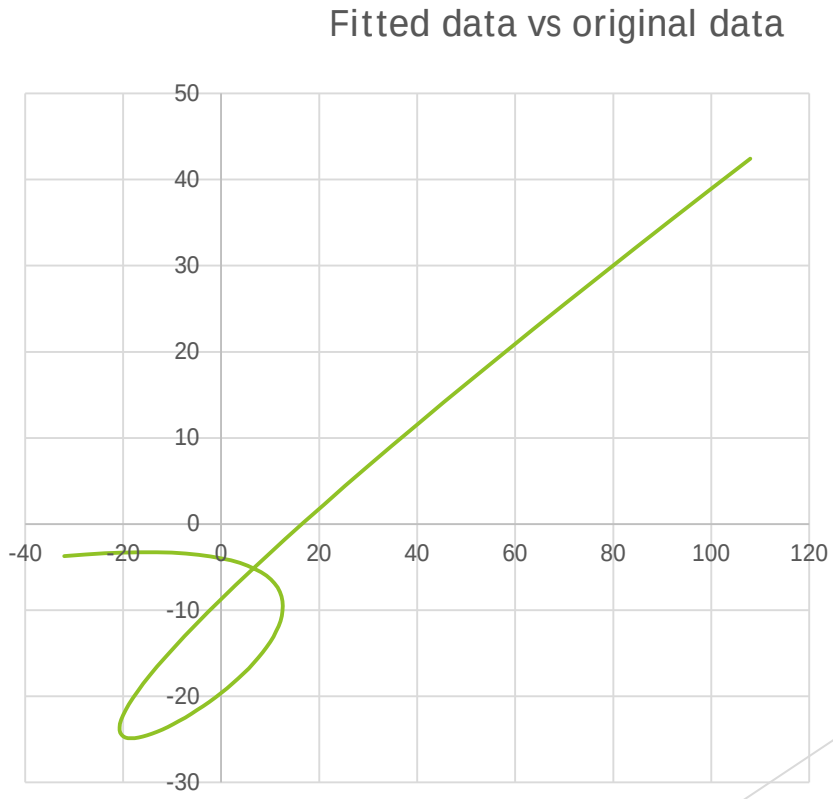
Plots x*(10 - x)


Original data, fitted data
30
25
20
15
10
5
0
0
2
4
6
8
10
12
Original data
Fitted data


Fitted data vs original data
30
25
20
15
10
5
0
0
5
10
15
20
25
30

Plots (x-1)*(x-4)*(x-8)





Conclusion

- Particle swarm optimization code, in this case, works good in finding the optimal solution of the given problems if the degree of polynomial is given as 2 i.e. quadratic
- When the degree is increased to 4, the code isn't able to provide a good solution
- The different parameters that can be varied to control the performance of the algorithm are –
  - Number of particles (generally 10-50)
    - more particles, more region covered
  - number of iterations
  - C1 – importance of personal best
  - C2 – importance of global best