ASSIGNMENT 3

May 20, 2021

PARTICLE SWARM OPTIMIZATION

Evolutionary Computation - CSCI 547

Shreya Jayeshbhai Patel
Student Id: 201906408

ABSTRACT

The assignments implements Particle Swarm Optimization. Using PSO it stimulates behaviour of swarm and is useful in order to optimize numeric problems iteratively. We observe nature and try to learn how biological phenomenon can be implemented in a computer system in order to optimize the problems. In PSO our main focus is how behaviour of group of birds and their interaction with environment

INTRODUCTION

Particle Swarm Optimization:

Particle Swarm intelligence is inspired by a swarm of birds. The overall concept of PSO is on what biological phenomena, the working is based upon.PSO is a population based algorithm Each particles is attracted to some degree to the best location it has so far, found by any member. After some steps, the population can unite around one location, or can join together around a few locations, or can continue to move.Particle Swarm Optimization has some similarities with genetic programming. A collection of individuals called particles moves in steps through a region.At each step, the algorithm evaluates the objective function at each particle After evaluation, the algorithm decides on the new velocity of each particle. The particles move, then the algorithm reevaluates

PSO versus GA

The have common procedure: randomly generate initial population, estimate fitness value, reproduction based on the fitness value.In genetic algorithm we have operators like crossover and mutation whereas, we don't have such operators in particle swarm optimization. Particles in update with internal velocity. They also have memory. Information Sharing in different in PSO when compared with GA. In GA chromosomes share information with each other and the whole group of population moves towards optimum result. Whereas, in PSO only gBest can give information to others. It is a one-way mechanism and it looks for only the best result.

TEST FUNCTION FOR OPTIMIZATION

1 Beale Function

f(x,y)=(1.5−x+xy)2+(2.25−x+xy2)2+(2.625−x+xy3)2

2 Three Hump Camel Function

f(x,y)=2x2−1.05x4+(x6)/6+xy+y2

PsuedoCode

- w = inertia
- c1 = x velocity coeff
- c2 = velocity coeff
- rand1,2 = 0¡=rand¡=1

  Random allocation algorithm
  $foreachparticleinsystem$
  $fordimensionxandy$
  initialize positions x—y (p) within the range given
  initialize velocity of each particle within permissible range
  Fitness value calculation
  for each particle
  calculate dimension z as fitness value
  if fitness value is better than p_best value in history –¿ set current value as *pbest*
  choose the particle having best among the whole group as g_best particle

  Update position and calculate velocity
  for each particle
  for each dimension
  calculate velocity according to below
  v(k+1)=w.v(k)+c1.rand1.(pbest-x) +c2.rand2.(gbest-x)

  update position as x(k+1) =x(k) +v(k+1)

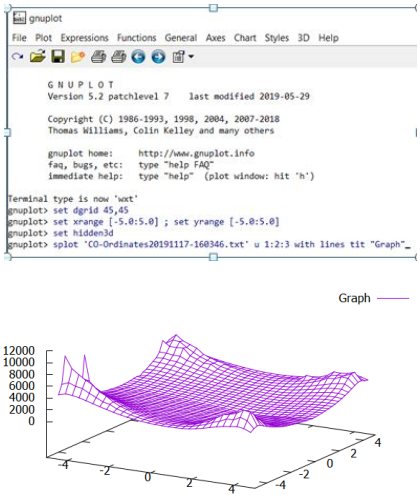
VISUALIZATION

For Visualization I have used gnuplot. Started with downloading gnuplot from https://sourceforge.net/projects/gnuplot/ After it is downloaded and installed start the execution file(wgnuplot.exe).
Now type the below code :

- set dgrid 45,45
- set xrange [-5.0:5.0] ; set yrange [-5.0:5.0]
- set hidden3d
- splot '¡coordinate file generated by py script¿' u 1:2:3 with lines tit "Graph"
  Co-ordinates file will be generated on every run
  I have done visualization for the following code:
  Run python script with 10000 elements. with 10 iteration each

VISUALIZATION FOR BEALE FUNCTION

- python PSOTest.py 10000 10 0.5 0.8 0.9 4.5 4.5 -4.5 -4.5
- files are generated in the folder. Co-Ordinate file : CO-Ordinates20191117-160346



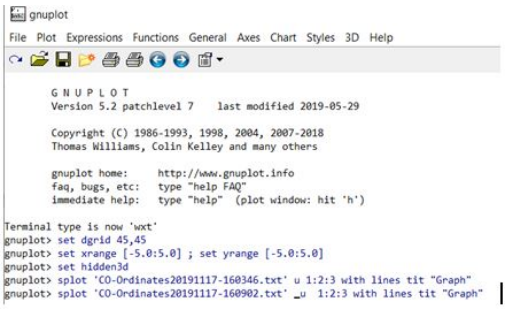
VISUALIZATION

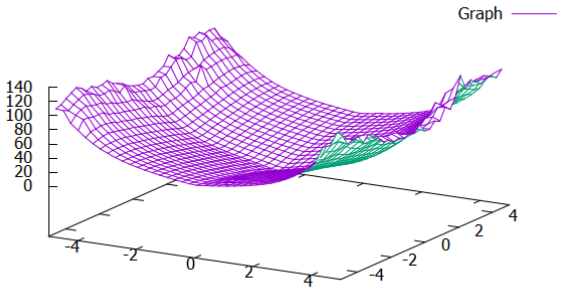
- **VISUALIZATION FOR Three Hump Camel FUNCTION**

- python PSOTest.py 10000 10 0.5 0.8 0.9 4.5 4.5 -4.5 -4.5

- For second function do following code changes: **Comment line no.91 i.e (mat[2]=(1.5-x+x*y)**2 + (2.25-x+x*y**2)**2 +(2.625-x+x*y**3)**2 (Beale function) and uncomment no. 92 i.e (mat[2]=2*(x**2)-1.05*(x**4)+(x**6.0)/6.0+x*y+(y**2) (three hump camel function)**

- CO-Ordinate file : CO-Ordinates20191117-160902 is generated





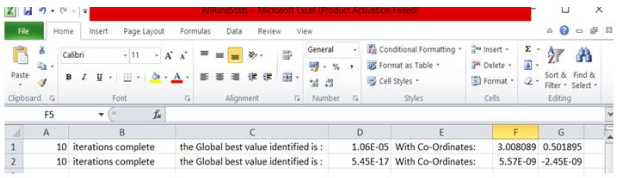
visualization for two function is obtained.

Output

when we run this program three files are generated in the folder(AssignPSO) namely AllRunStats,IndividualRun and CO-Ordinates file.

Stats

Statistics of all the runs made is in file AllRunStats.csv



	A	B	C	D	E	F	G
1	10	iterations complete	the Global best value identified is :	1.06E-05	With Co-Ordinates:	3.008089	0.501895
2	10	iterations complete	the Global best value identified is :	5.45E-17	With Co-Ordinates:	5.57E-09	-2.45E-09

References

[1] https://en.wikipedia.org/wiki/Test_functions_for_optimization

[2] https://en.wikipedia.org/wiki/Particle_swarm_optimization

[3] https://www.mathworks.com/help/gads/what-is-particle-swarm-optimization.html

[4] http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO_files/Page288.htm

[5]]https://adowney2.public.iastate.edu/projects/The_simplest_Particle_Swarm/The_simplest_Particle_Swarm.html

[1] https://nathanrooy.github.io/posts/2016-08-17/simple-particle-swarm-optimization-with-python/

Citation

[1] https://medium.com/analytics-vidhya/implementing-particle-swarm-optimization-pso-algorithm-in-python-9efc2eb179a6

[2] https://sourceforge.net/projects/gnuplot/

[2] https://www.sfu.ca/~ssurjano/beale.html

[2] http://benchmarkfcns.xyz/benchmarkfcns/threehumpcamelfcn.html