Report Analysis for Testing Particle Swarm Optimization Parameter
Generalized Rosen-Brocket Function

Wendy Mendoza
University of Texas Rio Grande Valley
Statistical Methods PHYS 5394
May 2022

Abstract:

In this lab I analyze PSO Generalized Rosen-Brocket function to easily see where the global minimum of the benchmark fitness function should be and the global minimum value. I directly compare the success of PSO in terms of how close it could reach either the minimizer or minimum. By studying the performing of PSO with different numbers of parallel runs, different number of iterations and seeing different space dimensionalities.

Software & Specifications:

All methods were performed using MATLAB Online.
Operation System: MacBook Air (13-inch, Early 2015)
Processor: 1.6 GHz Inter Core i5
Memory: 4 GB 1600 MHz DDR3
MATLAB Drive: 102.7MB / 5GB used
CPU time (MATLAB online): 16s
CPU (online) op-modes(s): 32-bit, 64-bit
Thread(s) per core (online): 2
Core(s) per socket (online): 8

PSO Performance:

For this analysis I have chosen benchmark function f(x)= sum{100(xi+1)-xi^2) ^2+(xi-1)^2} named Generalized Rosen-brock. Rosen-brock function is a non-convex function used as a performance test problem for optimization. Ranges of [-30,30] global minimum and its function value is 0. F(x) has a unique minimum at the point x = [1,1] where f(x)=0. This example shows several ways to minimize f(x) starting at the points xi= 1, and 2. Throughout this performance the best run of 4 with best location by x = 1 1 matrices and it best fitness: 8.1787e-19 with dimensionality of the search space for 2 and 4. Was find to be the correct minimum for this function being giving the correct PSO performances. However, dimensionality of 30 best fitness of 2.0688e-07 and multiple of x matrices best location of starting 0.999089 with other random matrices which was interesting to see being given the rmin of -30 and rmax of 30.

Results different dimensionalities & Elapsed time(s):

nDim = 2: 6.927304 seconds

nDim = 4: 6.990465 seconds

nDim = 15: 10.113200 seconds

nDim = 30: 11.134966 seconds