A Particle Swarm Optimization (PSO) Primer

With Applications

Brian Birge

Overview

- Introduction
- Theory
- Applications
- Computational Intelligence
- Summary

Introduction

- Subset of Evolutionary Computation
  - Genetic Algorithms
  - Evolutionary Programming
  - Evolution Strategies
  - Genetic Programming
- Behaviorally Inspired
  - Developed out of attempts to model bird flocks and fish schools
- Emergence
  - Complex behavior from simple rules

Theory - Qualitative Details

- PSO Algorithm in a nutshell
- Searches Hyperspace of Problem for Optimum
  - Define problem to search
    - How many dimensions?
    - Solution criteria?
  - Initialize Population
    - Random initial positions
    - Random initial velocities
  - Determine Global Best Position
  - Determine Personal Best Position
  - Update Velocity and Position Equations

Theory - Equations

- The Basic PSO algorithm consists of the velocity:

$$v_i(k+1) = v_i(k) + g_{1i}\left(p_i - x_i(k)\right) + g_{2i}\left(G - x_i(k)\right)$$

- ..and position:

$$x_i(k+1) = x_i(k) + v_i(k+1)$$

  - i – particle index
  - k – discrete time index
  - v – velocity of ith particle
  - x – position of ith particle
  - p – best position found by ith particle (personal best)
  - G – best position found by swarm (global best, best of personal bests)
  - $\gamma 1,2$ – random numbers on the interval [0,1] applied to ith particle

Theory - Equations ..

- The Common PSO Algorithm

$$v_i(k+1) = f(k)\,v_i(k) + a_1\left[g_{1i}\left(p_i - x_i(k)\right)\right] + a_2\left[g_{2i}\left(G - x_i(k)\right)\right]$$

  - ϕ - Inertia function
  - $\alpha 1,2$ – Acceleration constants

- As training progresses using a decreasing linear inertia function, the influence of past velocity becomes smaller.

Applications

- PSOt – A Matlab Toolbox
- Function Optimization
- Neural Net Training
  - Replacing Standard Backpropagation with PSO
- Engineering Examples

Applications - PSOt, A Matlab Toolbox

- Matlab: scientific computing language run in interpreter mode on a wide variety of operating systems.
- Toolbox: Suite of Matlab ‘plug-in’ programs developed by third parties.
- The PSOt includes:
  - Standalone MISO function optimizer
  - Neural Net Toolbox ‘plug-in’ for training

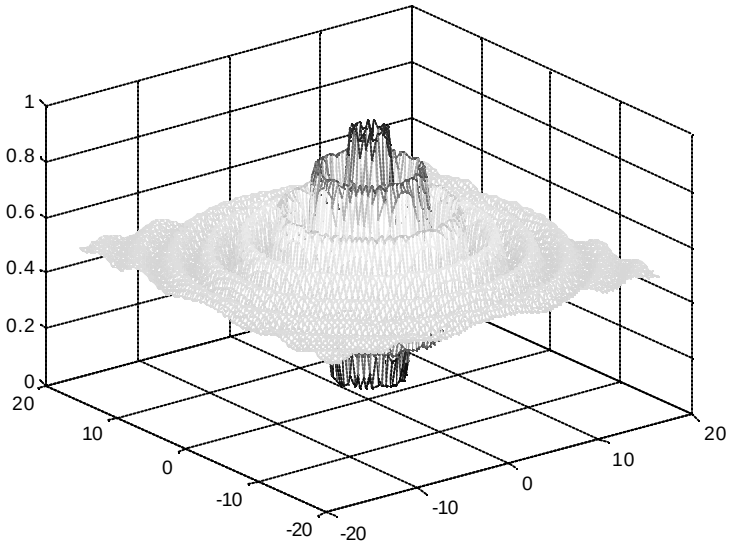
Applications – Function Optimization

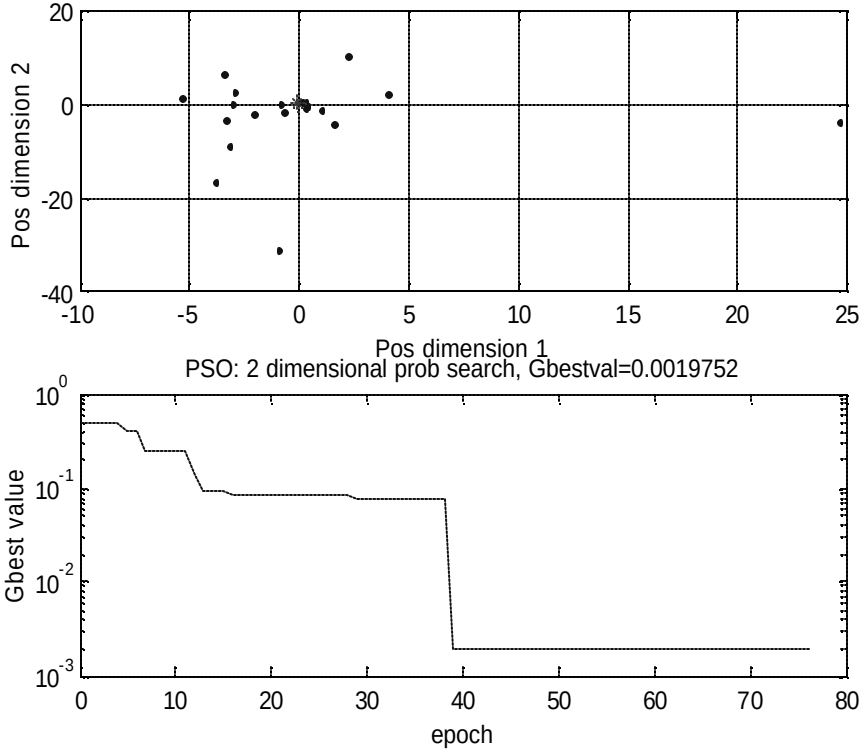
- Example: Schaffer's 'f6' function

```
>> pso('f6',2,[-100,100;-100,100],0)

PSO: 1/2000 iterations, GBest =
   0.499643.
PSO: 26/2000 iterations, GBest =
   0.0858499.
PSO: 51/2000 iterations, GBest =
   0.00197516.
PSO: 76/2000 iterations, GBest =
   0.00197516.
PSO: 89/2000 iterations, GBest =
   0.00197516.

ans =
   -4.400531404313024e-002
    4.527939759978006e-003
    1.975158000277422e-003
```



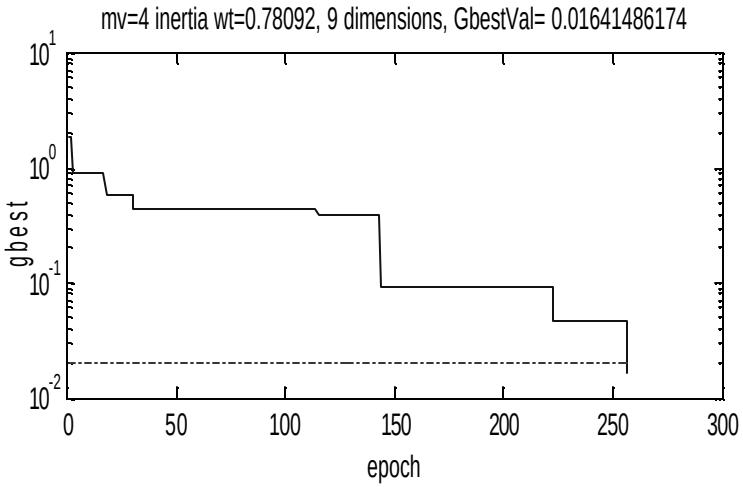


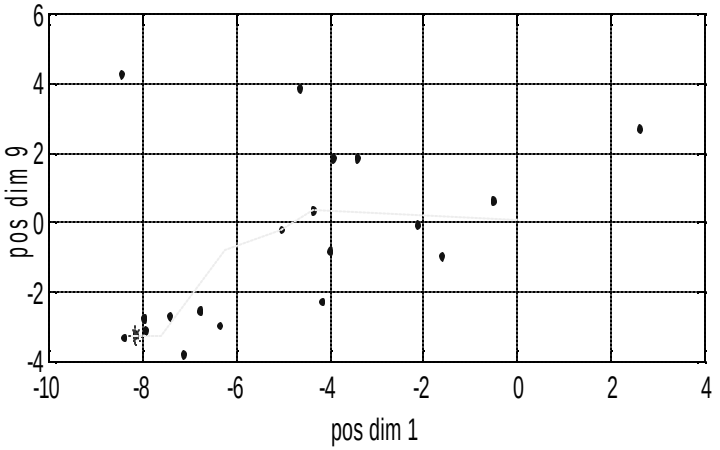
Applications – Artificial Neural Network Training

- Example of finding a neural net approximation to XOR

```
TRAINPSO: 100/1000 epochs,  gbest SSE
    0.428059141
  mv  = 4,  iwt = 0.8537691795
                      .
                      .
                      .
TRAINPSO: 225/1000 epochs,  gbest SSE
    0.04639357585
  mv  = 4,  iwt = 0.7953969313
TRAINPSO: 250/1000 epochs,  gbest SSE
    0.04454761798
  mv  = 4,  iwt = 0.7837224817

*********  Reached Goal   ********
TRAINPSO: 256/1000 epochs,  gbest SSE
    0.01641486174
  mv  = 4,  iwt = 0.7809206137
********* end of training ********
```





Applications – Engineering Examples

- Human Tremor Analysis
  - PSO used to evolve neural network weights
  - Neural network distinguishes between
    - Subjects with Parkinson's Disease, Essential Tremor
    - Normal Subjects
  - 60 input, 12 hidden neurons, 2 outputs
  - 100% success with generalization
  - Very fast human tremor classification with very small computational hit.

Applications – Engineering Examples..

- Ingredient Mix Optimization
  - Refers to a mixture of ingredients used to grow production strains of micro-organisms that naturally secrete or manufacture something of interest.
  - PSO provided optimized ingredient mix much better than then current state of the art
- Battery Pack State of Charge Estimation
  - PSO used in conjunction with Backpropagation
  - Important in development of electric and hybrid vehicle technology
  - PSO used to optimize training set for neural network in addition to training.

Computational Intelligence

- Artificial Intelligence
  - A name for a paradigm in which people attempt to elicit intelligence from machines.
- Computational Intelligence
  - Practical adaptation concepts, paradigms, algorithms, and implementations that enable or facilitate appropriate actions (intelligent behavior) in complex and changing environments.
- CI is different than AI
  - Deals well with complex, dynamic, poorly defined problems that AI has problems with.
- Examples
  - Neuro-Fuzzy controllers
  - 'Alternatively' trained neural nets
  - Evolution of Fuzzy-Expert Systems using PSO

Summary

- PSO is inspired but not based on animal behavior.
- A few simple rules result in complex action, i.e. Emergence.
- PSO is simple to code and has small computational cost.
- Successful in application to wide range of engineering and computer science problems.

References

- Birge, B., 2003, *PSOt, A Particle Swarm Optimization Toolbox for Matlab*, **IEEE Swarm Intelligence Symposium Proceedings**, April 24-26
- Eberhart, R., Simpson, P., Dobbins, R., 1996, *Computational Intelligence PC Tools,* **Academic Press, Inc.,** pp. 212-223.
- Haykin, S., 1999, *Neural Networks, a Comprehensive Foundation, Second Edition,* **Prentice-Hall**
- Houck, C., Joines, J., and Kay M., 1995, *A Genetic Algorithm for Function Optimization: A Matlab Implementation,* **ACM Transactions on Mathematical Software**, Submitted 1996
- Kennedy, J., Eberhart, R., 1995, *Particle Swarm Optimization*, from **Proc. IEEE Int'l. Conf. on Neural Networks** (Perth, Australia), IEEE Service Center, Piscataway, NJ, **IV:** 1942-1948
- Kennedy, J., Eberhart, R., Shi, Y., 2001, *Swarm Intelligence,* **Academic Press, Inc.**