Computational Intelligence

Unit # 6

Particle Swarm Optimization

- Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behavior of bird flocking or fish schooling.
- The system is initialized with a population of random solutions and searches for optima by updating generations.
- Unlike EA, PSO has no evolution operators such as crossover and mutation.

Concept

- In PSO, each single solution is a "bird" in the search space. We call it "particle".
- All of particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles.
- The particles fly through the problem space by following the current optimum particles.
- PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations.

Algorithm Description

- Each particle keeps track of its coordinates in the problem space which are associated with the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called *pbest*.
- Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the neighbors of the particle. This location is called *lbest*.
- When a particle takes all the population as its topological neighbors, the best value is a global best and is called *gbest*.

Algorithm Description (Cont'd)

- At each step of the algorithm, particles are displaced from their current position by applying a velocity (gradient) vector to them.
- The magnitude and direction of their velocity at each step is influenced by their velocity in the previous iteration of the algorithm, simulated momentum, and the location of the a particle relative to the location of its pbest and the gbest.
- Therefore, at each step, the size and direction of each particle's move is a function of its own history (experience), and the social influence of its peer group.

Algorithm Description (Cont'd)

- After finding the two best values, the particle updates its velocity and positions with following equation (a) and (b).
- v[ ] = v[ ] + c1 * rand() * (pbest[ ] - present[ ]) + c2 * rand() * (gbest[ ] - present[ ]) (a)
  present[] = persent[ ] + v[ ] (b)

  v[ ] is the particle velocity, persent[] is the current particle (solution). pbest[ ] and gbest[ ] are defined as stated before. rand () is a random number between (0,1). c1, c2 are learning factors. usually c1 = c2 = 2.

Basic Theme

- The particle swarm optimization concept consists of, at each time step, changing the velocity of (accelerating) each particle toward its p*best* and *lbest* locations (local version of PSO).
- Acceleration is weighted by a random term, with separate random numbers being generated for acceleration toward *pbest* and *lbest* locations.

Role of pbest and gbest

- At the start of the algorithm, the pbest for each particle is set at its initial location, and gbest is set to the location of the best of the pbests.
- In each iteration of the algorithm, a particle is stochastically accelerated towards its previous best position and towards a neighborhood (global) best position, thereby forcing particles to continually search in the most-promising regions found so far in the solution space.

Role of c1 and c2

- The weight coefficient c1 and c2 control the relative impact of the pbest and gbest locations on the velocity of a particle.
- Low values for c1 and c2 allow each particle to explore far away from already uncovered good points, high values of the parameters encourage more intensive search of regions close to these points.

Comparing PSO with the EA

- Like the EA, PSO is population-based, it is typically initialized with a population (swarm) of random encodings of solutions, and search proceeds by updating these encodings over a series of generations (iterations).
- Both systems do not guarantee success.
- Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population.
- Unlike the EA, PSO has no explicit selection process as all particles persist over time. Instead a *memory* in the form of gbest/lbest is substituted for selection.
- PSO also does not have genetic operators like crossover and mutation.

Comparing PSO with the EA (Cont'd)

- Compared with evolutionary algorithms (EAs), the information sharing mechanism in PSO is significantly different.
- In EAs, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area.
- In PSO, only gBest (or lBest) gives out the information to others. It is a one -way information sharing mechanism. The evolution only looks for the best solution.
- Compared with EA, all the particles tend to converge to the best solution quickly even in the local version in most cases.

Artificial Immune Systems (Source: Wikipedia)

- The field of Artificial Immune Systems (AIS) is concerned with abstracting the structure and function of the immune system to computational systems, and investigating the application of these systems towards solving computational problems from mathematics, engineering, and information technology.
- AIS is a sub-field of Biologically-inspired computing, and Natural computation, with interests in Machine Learning and belonging to the broader field of Artificial Intelligence.

History

- AIS began in the mid 1980s with Farmer, Packard and Perelson's (1986) and Bersini and Varela's papers on immune networks (1990).
- However, it was only in the mid 90s that AIS became a subject area in its own right.

Inspiration

- AIS is inspired by the working of immune systems.
- The immune system is comprised of an intricate network of specialized tissues, organs, cells and chemical molecules.
- The capabilities of the natural immune system include the ability to recognize, destroy and remember an almost unlimited number of pathogens (foreign or non-self objects that enter the body, including viruses, bacteria, multi-cellular parasites, and fungi), and also to protect the organism from misbehaving cells in the body.

Working of Immune System

- To assist in protecting the organism, the immune system has the capability to distinguish between self and non-self.
- Critically, the system does not require exhaustive training with negative (non-self) examples to make these distinctions, but can identify a pathogen as being non-self even though it has never been encountered before.

Working of Immune System (Cont'd)

- Both the innate and acquired immune systems are compromised of a variety of molecules, cells and tissues.
- The most important cells are leukocytes (white blood cells) which can be divided into two major categories: phagocytes and lymphocytes.
- Lymphocytes circulate constantly through the blood, lymph, lymphoid organs and tissue spaces.
- A major component of the population of lymphocytes is made up of B and T cells.

Working of Immune System (Cont'd)

- These cells are capable of recognizing and responding to certain antigen (foreign molecules) patterns presented on the surface of pathogens.
- Antigens are not the invading pathogens themselves, rather they are molecular signature expressed by the invading pathogen.
- The control of adaptive immunity can be divided into two branches: *humoral immunity* which is controlled by B-cells, and *cellular immunity* which is controlled by T-cells.

Destruction Process

- Antigen-secreting pathogen enters the body.
- B-cells are activated by the foreign antigen.
- With help of T-cells, B-cells undergo cloning and mutation.
- Plasma B-cells secrete immunoglobulins which attach to the antigen.
- Marked antigens are attacked by the immune system.
- Memory of the antigen is maintained by B memory cells.

Metaphor

- Once the immune system detects a pathogen, B-cells begin to clone at a rate proportional to their affinity to the antigen that stimulated them.
- Adopting this metaphor in order to design an optimization algorithm, the antibody can be considered as a potential solution, the antigen is a test dataset, and the degree of binding or fit between the antibody and the antigen represents the fitness or the quality of the solution.
- The objective, therefore, is to start from an initial population of solutions, test them against the dataset, and, using the algorithm iteratively, improve the quality of the solutions in the population.

AIS Algorithm

- Create an initial random population P of solution vectors (antibodies).
- Select a subset F of the solutions from P, biasing the selection process towards better solutions.
- For each member of F (the parents), create a set of clones, with better members of F producing more clones.
- Mutate each of these clones, in inverse proportion of their parent's fitness. Better solutions are mutated less.

AIS Algorithm (Cont'd)

- Select a subset of the newly generated solutions S.
- Create a number of newly created random solutions R.
- Replace poorer members of P with better solutions from S and R.

AIS Algorithm

(Source: http://www.artificial-immune-systems.org)

```
input   : S = set of patterns to be recognised, n the number of worst elements to select for removal
output  : M = set of memory detectors capable of classifying unseen patterns

begin

  Create an initial random set of antibodies, A

  forall  patterns in S   do
    Determine the affinity with each antibody in A
    Generate clones of a subset of the antibodies in A  with the highest affinity.
      The number of clones for an antibody is proportional to its affinity
    Mutate attributes of these clones to the set A , and place a copy of the highest
      affinity antibodies in A  into the memory set, M
    Replace the n lowest affinity antibodies in A with new randomly generated antibodies
  end

end
```