PSO Variations

Pouya Mehrannia

PSO Review

- A stochastic optimization approach that manipulates a number of candidate solutions at once,
- A solution is referred to as a ***particle***, the whole population is referred to as a ***swarm***,
- Each particle holds information essential for its movement.

PSO Review

- Each particle holds:
  - Its current position x_i,
  - Its current velocity v_i,
  - The best position it achieved so far, personal best, $pbest_i$ (sometimes p_i for short),
  - The best position achieved by particles in its neighbourhood *Nbest*
  - If the neighbourhood is the whole swarm, the best achieved by the whole swarm is called *global best*, $gbest_i$ (sometimes p_g for short).
  - If the neighbourhood is restricted to few particles, the best is called *local best, lbest (or* p_l*)*

PSO Review

- **Equations of motion:**

Inertia weight variation

$$v_{t+1}^{id} = w * v_t^{id} + c_1 r_1^{id}\left(pbest_t^{id} - x_t^{id}\right) + c_2 r_2^{id}\left(Nbest_t^{id} - x_t^{id}\right)$$

$$x_{t+1}^{id} = x_t^{id} + v_{t+1}^{id}$$

- **Where**
  - v is the velocity of particle id,
  - w is the inertia weight,
  - c_1, c_2 are the acceleration coefficients,
  - r_1,r_2 are randomly generated numbers in [0, 1],
  - x is the position of the particle
  - t is the iteration number,
  - i and d are the particle number and the dimension.

Assignment 4

Part I [25 pts] Code a simple PSO to solve the problem. To do this you need to encode the problem, initialize a population, select a velocity update equation, and select a stopping criterion. Run your code and report: final solution, plot the progress of the average fitness and the best particle fitness.

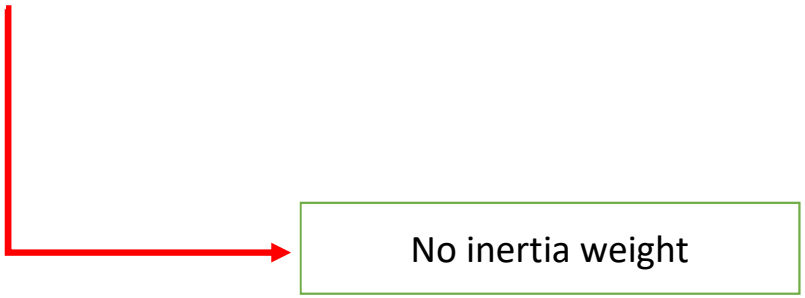
Part II [15 pts]

- [5 pts] Use the Inertia Weight version of velocity update equation with Global best. Run your code and report: final solution, and plot the progress of the average fitness of the population and the global best particle fitness.
- Use Constriction Factor version of velocity update equation with Global Best. Run your code and report: final solution, and plot the progress of the average fitness of the population and the global best particle fitness.
- Use the Guaranteed Convergence PSO (GVPSO). Run your code and report: final solution, and plot the progress of the average fitness of the population and the global best particle fitness.

Simple PSO

For gbest PSO method, the velocity of particle i is calculated by

$$v_{ij}^{t+1} = v_{ij}^{t} + c_1 r_{1j}^{t}\left[P_{best,i}^{t} - x_{ij}^{t}\right] + c_2 r_{2j}^{t}\left[G_{best} - x_{ij}^{t}\right]$$



Inertia weight PSO

The inertia weight, denoted by ω, is considered to replace V_{max} by adjusting the influence of the previous velocities in the process, i.e. it controls the momentum of the particle by weighing the contribution of the previous velocity. The inertia weight 'ω' will at every step be multiplied by the velocity at the previous time step, i.e. v_{ij}^{t}. Therefore, in the gbest PSO, the velocity equation of the particle i with the inertia weight changes

$$v_{ij}^{t+1} = \omega v_{ij}^{t} + c_1 r_{1j}^{t}[P_{best,i}^{t} - x_{ij}^{t}] + c_2 r_{2j}^{t}[G_{best} - x_{ij}^{t}]$$

If $\omega \geq 1$, then the velocities increase over time and particles can hardly change their direction to move back towards optimum, and the swarm diverges. If $\omega \ll 1$, then little momentum is only saved from the previous step and quick changes of direction are to set in the process. If $\omega = 0$, particles velocity vanishes and all particles move without knowledge of the previous velocity in each step

Constriction Coefficient PSO

In order to insure convergence of the PSO algorithm, the velocity of the constriction factor based approach can be expressed as follows:

$$V_i^{k+1} = K\left[V_i^k + c_1 r_1 \times \left(Pbest_i^k - X_i^k \right) + c_2 r_2 \times \left(Gbest^k - X_i^k \right) \right]$$

$$K = \frac{2}{\left| 2 - \varphi - \sqrt{\varphi^2 - 4\varphi} \right|}, \; where \; \varphi = c_1 + c_2, \; \varphi > 4$$

The convergence characteristic of the system can be controlled by φ. In the constriction factor approach, the φ must be greater than 4.0 to guarantee stability. However, as φ increases, the constriction factor, K decreases and diversification is reduced, yielding slower response.

Constriction Coefficient PSO

Typically, when the constriction factor is used, φ is set to 4.1 (i.e. $c_1, c_2 = 2.05$) and the constant multiplier K is thus 0.729. This results in the previous velocity being multiplied by 0.729 and the terms $\left(Pbest_i^k - X_i^k\right)$ and $\left(Gbest^k - X_i^k\right)$ being multiplied by $0.729 \times 2.05 = 1.49445$ (times a random number between 0 and 1).

Guaranteed Convergence PSO (GCPSO)

When the current position of a particle coincides with the global best position, then the particle moves away from this point if its previous velocity is non-zero. In other words, when $x_{ij}^t = P_{best,i}^t = G_{best}^t$, then the velocity update depends only on the value of ωv_{ij}^t. Now if the previous velocities of particles are close to zero, all particles stop moving once and they catch up with the global best position, which can lead to premature convergence of the process. This does not even guarantee that the process has converged to a local minimum, it only means that all particles have converged to the best position in the entire swarm. This leads to stagnation of the search process which the PSO algorithm can overcome by forcing the global best position to change when $x_{ij}^t = P_{best,i}^t = G_{best}^t$

Guaranteed Convergence PSO (GCPSO)

To solve this problem a new parameter is introduced to the PSO. Let τ be the index of the global best particle, so that

$$y_\tau = G_{best}$$

A new velocity update equation for the globally best positioned particle, y_τ, has been suggested in order to keep y_τ moving until it has reached a local minimum. The suggested equation is

$$v_{\tau j}^{t+1} = -x_{\tau j}^{t} + G_{best}^{t} + \omega v_{\tau j}^{t} + \rho^{t}(1 - 2r_{2j}^{t})$$

where

'ρ^t' is a scaling factor and causes the PSO to perform a random search in an area surrounding the global best position G_{best}. It is defined in equation (4.10) below,

'$-x_{\tau j}^{t}$' resets the particle's position to the position G_{best}^{t},

'$\omega v_{\tau j}^{t}$' represents the current search direction,

'$\rho^t(1 - 2r_{2j}^{t})$' generates a random sample from a sample space with side lengths $2\rho^t$.

Guaranteed Convergence PSO (GCPSO)

for the global best particle τ yields the new position update equation

$$x_{\tau j}^{t+1} = G_{best}^{t} + \omega v_{\tau j}^{t} + \rho^{t}(1 - 2r_2^{t})$$

while all other particles in the swarm continue using the usual velocity update The parameter ρ^t controls the diameter of the search space and the value of ρ^t is adapted after each time step, using

$$\rho^{0} = 1.0$$

$$\rho^{t+1} = \begin{cases} 2\rho^{t} & \text{if } \#successes(t) > \epsilon_s \\ 0.5\rho^{t} & \text{if } \#failures\ (t) > \epsilon_f \\ \rho^{t} & \text{otherwise} \end{cases}$$

Guaranteed Convergence PSO (GCPSO)

where $\#successes$ and $\#failures$ respectively denote the number of consecutive successes and failures, and a failure is defined as $f(G_{best}^{t+1}) = f(G_{best}^{t})$. The following conditions must also be implemented

$$\#successes(t+1) > \#successes(t) \Rightarrow \#failures\,(t+1) = 0$$

and

$$\#failures\,(t+1) > \#failures(t) \Rightarrow \#successes(t+1) = 0$$

Therefore, when a success occurs, the failure count is set to zero and similarly when a failure occurs, then the success count is reset.

Guaranteed Convergence PSO (GCPSO)

The optimal choice of values for ϵ_s and ϵ_f depend on the objective function. It is difficult to get better results using a random search in only a few iterations for high- dimensional search spaces, and it is recommended to use $\epsilon_s = 15$ and $\epsilon_f = 5$. On the other hand, the optimal values for ϵ_s and ϵ_f can be found dynamically. For instance, ϵ_s may be increased every time that $\#failures > \epsilon_f$ i.e. it becomes more difficult to get the success if failures occur frequently which prevents the value of ρ from fluctuating rapidly. Such strategy can be used also for ϵ_f