Particle Swarm Optimization for test functions optimization

1 PSO

1: $swarm\_initialize(swarm\_size)$
2: $t \leftarrow 0$
3: **while** $terminal\_condition$ **not** $TRUE$ **do**
4: **for** $i = 1$ **to** $swarm\_size$ **do**
5: **if** $f(x_i) < f(personal\_best_i)$ **then**
6: $personal\_best_i \leftarrow x_i$
7: **end if**
8: **if** $f(personal\_best_i) < f(global\_best)$ **then**
9: $global\_best \leftarrow personal\_best_i$
10: **end if**
11: **end for**
12: **for** $i = 1$ **to** $swarm\_size$ **do**
13: **for** $j = 1$ **to** d **do**
14: $v_{i,j} \leftarrow \omega \cdot v_{i,j} + \phi_p \cdot rU(0,1) \cdot (personal\_best_{i,j} - x_{i,j}) + \phi_g \cdot rU(0,1) \cdot (global\_best_j - x_{i,j})$
15: $x_{i,j} \leftarrow x_{i,j} + v_{i,j}$
16: **end for**
17: **end for**
18: $t \leftarrow t + 1$
19: **end while**
20: **return** $swarm$

2 Algorithm description

- A prarticle structure:
  - x — a vector with d numbers which represents optimization function arguments - point in the domain space. $\Re^n$). Initialized randomly from uniform distribution and the range = $domain$.
  - v — a vector with d numbers which represents a velocity.
  - $personal\_best$ — the best particle posititon so far. A vector with d numebers.
  - f — fitness function (calculated from the formula e.g. Schwefel function)

domain is a given domain for a specific test function (benchmark) that will be solved by ES.

- Initial parameters
  - $swarm\_size$ —a number of individuals in the swarm: recommended values from 10 to 50
  - ϕ_p — the parameter responsible for the individuality of the particle. If the value is high, the particles are willing to follow their best locations so far $0 < \phi_p < 4$.
  - ϕ_g — the parameter responsible for the social behavoir of the particle. If there is a high value, the best particle attracts the swarm $0 < \phi_g < 4$.
  - ω — Inertia affects the velocity. ω =1 — particles don't slow down, ω <1 particles slow down over time, $\omega > 1$ particles accelerate.
  - v_max — max velocity (usually the width of the optimised function domain) e.g. $domain = \{-5 \dots 5\}$ then $v\_max = 10$.
  - max_num_steps - maxim number of steps e.g. 10000
- A $teminal\_condition$ control maximum number of steps or finding the solution close to the know global optimum e.g.$f(global\_best) - f\_globalne < 0,0001$.
- Initialization

  Creating a swarm and calculating the local and global best positions.

  for $i = 1$ **to** $swarm\_size$ **do**
    for $j = 1$ **to** d **do**
      $v_{i,j} \leftarrow rand(-v\_max/3, v\_max/3)$
      $x_{i,j} \leftarrow rand(domain\_min, domain\_max)$
    end for
    $f \leftarrow f(x_i)$
    $personal\_best_i \leftarrow x_i$
    if $i == 1$ **then**
      $global\_best \leftarrow x_i$
    end if
    if $f(personal\_best_i) < f(global\_best)$ **then**
      $global\_best \leftarrow personal\_best_i$
    end if
  end for

  where:
  $domain\_min$, $domain\_max$ min and max value of the optimised function.

- $rU(0,1)$ — a random number from the uniform distribution from a range [0,1].

3 Benchmarks

To test the algorithm the following function should be used:

1. RASTRIGIN FUNCTION
   - `http://www.sfu.ca/~ssurjano/rastr.html`,
   - $domain = [-5.12, 5.12]$
   - Global minimum $f(0, 0, \ldots, 0) = 0$
2. GRIEWANK FUNCTION
   - `http://www.sfu.ca/~ssurjano/griewank.html`,
   - $domain = [-600, 600]$
   - Global minimum $f(0, 0, \ldots, 0) = 0$
3. SPHERE FUNCTION
   - `http://www.sfu.ca/~ssurjano/spheref.html`,
   - $domain = [-5.12, 5.12]$
   - Global minimum $f(0, 0, \ldots, 0) = 0$
4. ZAKHAROV FUNCTION
   - `http://www.sfu.ca/~ssurjano/zakharov.html`,
   - $domain = [-5, 10]$
   - Global minimum $f(0, 0, \ldots, 0) = 0$
5. EASOM FUNCTION
   - `http://www.sfu.ca/~ssurjano/easom.html`,
   - $domain = [-100, 100]$
   - Global minimum $f(\pi, \pi) = -1$
6. STYBLINSKI-TANG FUNCTION
   - `http://www.sfu.ca/~ssurjano/stybtang.html`,
   - $domain = [-5, 5]$
   - Global minimum $f(-2.903534, \ldots, -2.903534) = -39.16599d$, where d is dimention