Evolution Strategies $(\mu/\varrho + \lambda)$-ES for test functions optimization

1 Evolution Strategies in version (1+1)-ES

```
population ← initialize (μ)
g ← 0
while terminal_condition not TRUE do
    offspring ← prepare (λ)
    for o = 1 to λ do
        parents ← marriage (population, ϱ)
        s ← s_recombination (parents)
        x ← x_recombination (parents)
        s ← s_mutation (s)
        x ← x_mutation (x, s)
        f ← fitness_function (x)
        offspring_o ← struct (x, s, f)
    end for
    population ← selection (offspring + μ)
    g ← g + 1
end while
return population
```

2 Algorithm description

- *initialize*(μ) returns the μ individuals, which could be a structure/object:
  - x — 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*.
  - s — vector with d numbers which represents standard deviation to control the mutation strength in every dimention. Initialized randomly from uniform distribution and the range = $[0.1 : 10]$.
  - function — 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
  - μ —number of individuals in the population: recommended values from 1 to 100
  - ϱ — number of individuals in the pool of parents for reproduction: recommended values from 1 to μ
  - λ — the number of offsprings: recommended values from 1 to 100 .
  - max_num_gen - maxim number of generations e.g. 10000
- A $teminal\_condition$ control maximum number of steps or finding the solution close to the know global optimum e.g. $fitness\_function(best) - f\_global < 0,0001$.
- Selection of the pool of parents

  function $parents = marriage\,(population, \varrho)$

  $parents \leftarrow prepare\,(\varrho)$
  $selected\_individuals \leftarrow randperm\,(length\,(population))$
  for $i = 1$ **to** ϱ **do**
      $parents_i \leftarrow population_{selected\_individuals_i}$
  end for
  return $parents$

- Pseudo-code of the recombination algorithm for endogenous parameters

  function $s = s\_recombination\,(parents)$

  $n \leftarrow length\,(parents.s)$
  $s \leftarrow prepare\,(n)$
  for $i = 1$ **to** n **do**
      $s_i \leftarrow mean\_s\,(parents, i)$
  end for
  return s

  where: $mean\_s\,(parents, i)$ is the average value of all the parents.

- Pseudo-code of recombination

  function $x = x\_recombination\,(parents)$

  $n \leftarrow length\,(parents)$
  $x \leftarrow prepare\,(n)$
  for $i = 1$ **to** n **do**
      $x_i \leftarrow mean\_x\,(parents, i)$
  end for
  return x

  where, $mean\_x\,(parents, i)$ is the average value of all the parents.

- Endogenous parameters mutation

  function $out\_s = s\_mutation\,(s)$

  $n \leftarrow length\,(s)$
  $out\_s \leftarrow prepare\,(n)$

$c \leftarrow 1$
$\tau_0 \leftarrow e^{\frac{c}{\sqrt{2n}} \cdot \aleph(0,1)}$
$\tau \leftarrow \frac{c}{\sqrt{2\sqrt{n}}}$
for $i = 1$ **to** n **do**
  $out\_s_i \leftarrow \tau_0 \cdot s_i \cdot e^{\tau \cdot \aleph(0,1)}$
end for
return out_s

where:

- $\aleph\,(0,1)$ is a random number from a normal distribution with the mean equal to 0 and standard deviation of 1
- variable c is positive number could be set to 1.

- Problem parameters mutation

function $out\_x = x\_mutation\,(x, s)$
$n \leftarrow length\,(y)$
$out\_x \leftarrow prepare\,(n)$
for $i = 1$ **to** n **do**
  $out\_x_i \leftarrow x_i + s_i \cdot \aleph\,(0,1)$
end for
return out_x

where, $\aleph\,(0,1)$ is a random number from a normal distribution with the mean equal to 0 and standard deviation of 1

- Selection called plus $(+)$

function $new\_population = selection\,(offspring, population, \mu)$
$new\_population \leftarrow sort\,([offspring, population], ascd)$
$new\_population \leftarrow new\_population\,(1 : \mu)$
return $new\_population$

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