Evolution Strategies

- Particularities
- General structure
- Recombination
- Mutation
- Selection
- Adaptive and self-adaptive variants

Particularities

Evolution strategies: evolutionary techniques used in solving continuous optimization problems

History: the first strategy has been developed in 1964 by Bienert, Rechenberg si Schwefel (students at the Technical University of Berlin) in order to design a flexible pipe

Main ideas [Beyer &Schwefel – ES: A Comprehensive Introduction, 2002]:

- Use one candidate (containing several variables) which is iteratively evolved
- Change all variables at a time, mostly slightly and at random.
- If the new set of variables does not diminish the goodness of the design (solution), keep it, otherwise return to the old status.

Particularities

Data encoding: real (the individuals are vectors of float values belonging to the definition domain of the objective function)

Main operator: mutation (based on parameterized random perturbation)

Secondary operator: recombination

Particularity: self adaptation of the mutation control parameters

General structure

Problem (minimization):

Find x* in $D \subset R^n$ such that

f(x*)<f(x) for all x in D

The population consists of elements from D (vectors with real components)

Rmk. A configuration is better if the value of f is smaller.

Structure of the algorithm

Population initialization

Population evaluation

REPEAT

construct offspring by recombination

change the offspring by mutation

offspring evaluation

survivors selection

UNTIL <stopping condition>

Resource related criteria (e.g.: generations number, nfe)

Criteria related to the convergence (e.g.: value of f)

Recombination

Aim: construct an offspring starting from a set of parents

$$y = \sum_{i=1}^{\rho} c_i x^i, \quad 0 < c_i < 1, \ \sum_{i=1}^{\rho} c_i = 1$$

Intermediate (convex): the offspring is a linear (convex) combination of the parents

$$y_j = \begin{cases} x_j^1 & \text{with probability } p_1 \\ x_j^2 & \text{with probability } p_2 \\ \vdots & \\ x_j^{\rho} & \text{with probability } p_{\rho} \end{cases},$$

$$0 < p_i < 1, \ \sum_{i=1}^{\rho} p_i = 1$$

Discrete: the offspring consists of components randomly taken from the parents

Recombination

Geometrical recombination:

$$y_j = (x_j^1)^{c_1}(x_j^2)^{c_2}\ldots(x_j^{\rho})^{c_{\rho}}, \quad 0 < c_i < 1, \sum_{i=1}^{\rho} c_i = 1$$

Remark: introduced by Z. Michalewicz for solving constrained optimization problems with constraints involving the product of components (e.g. $x_1x_2\ldots x_n > c$)

Heuristic recombination:

$y=x^i+u(x^i-x^k)$ with x^i an element at least as good as x^k

u – random value from (0,1)

Recombination

Simulated Binary Crossover (SBX)

- It is a recombination variant (for real encoded data) which simulates the behavior of one cut point crossover used in the case of binary encoding
- It produces two children c1 and c2 starting from two parents p1 and p2

$$c_1 = \overline{p} - \frac{\beta}{2}(p_2 - p_1)$$

$$c_2 = \overline{p} + \frac{\beta}{2}(p_2 - p_1)$$

$$\overline{p} = (p_1 + p_2)/2$$

Rmk: β is a random value generated according to the distribution given by:

$$prob(\beta) = \begin{cases} 0.5(n+1)\beta^n & \beta \le 1 \\ 0.5(n+1)\dfrac{1}{\beta^{n+2}} & \beta > 1 \end{cases}$$

Rmk: n can be any natural value; high values of n lead to children which are close to the parents

Mutation

Basic idea: perturb each element in the population by adding a random vector

$$x' = x + z$$

$$z = (z_1, \ldots, z_n)$$

random vector with mean 0 and

covariance matrix $C = (c_{ij})_{i,j=1,n}$

Particularity: this mutation favors the small changes of the current element, unlike the mutation typical to genetic algorithms which does not differentiate small perturbations from large perturbations

Mutation

Variants:

- The components of the random vector are independent random variables having the same distribution ($E(z_i z_j)=E(z_i)E(z_j)=0$).

  Examples:

  a) each component is a random value uniformly distributed in [-s,s]

  b) each component has the normal (Gaussian) distribution N(0,s)

  Rmk. The covariance matrix is a diagonal matrix $C=\text{diag}(s^2,s^2,\ldots,s^2)$ with s the only control parameter of the mutation

Mutation

Variants:

- The components of the random vector are independent random variables having different distributions ($E(z_i z_j)= E(z_i)E(z_j)= 0$)

  Examples:

  a) the component z_i of the perturbation vector has the uniform distribution on $[-s_i,s_i]$

  b) each component of the perturbation vector has the distribution $N(0, s_i)$

  Rmk. The covariance matrix is a diagonal matrix: $C=diag(s^2_1,s^2_2,\ldots,s^2_n)$ and the control parameters of mutation are $s_1,s_2,\ldots,s_n$

Mutation

Variants:

- The components are dependent random variables

  Example:

  a) the vector z has the distribution N(0,C)

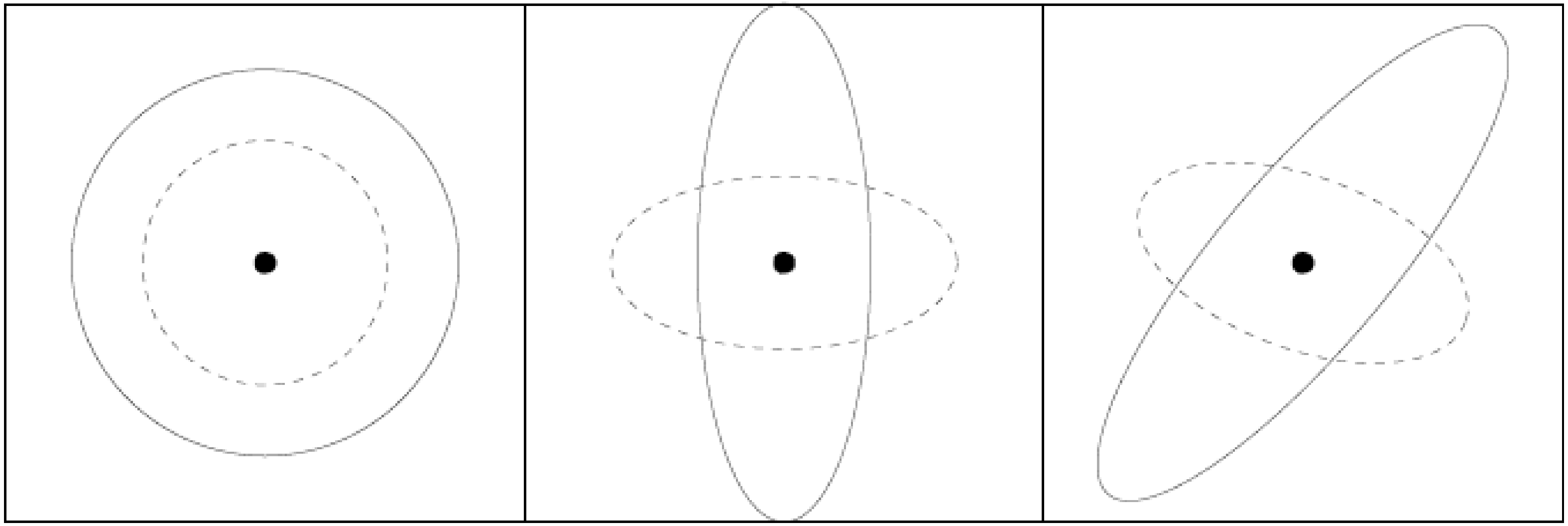
Rmk. There are n(n+1)/2 control parameters of the mutation:

$s_1, s_2, \ldots, s_n$ - mutation steps

$a_1, a_2, \ldots, a_k$ - rotation angles (k=n(n-1)/2)

$c_{ij} = ½ \cdot (s_i^2 - s_j^2) \cdot \tan(2\, a_l)$

Mutation



$\mathcal{N}(m, \sigma^2 \mathbf{I}) \sim m + \sigma \mathcal{N}(\mathbf{0}, \mathbf{I})$ $\quad$ $\mathcal{N}(m, \mathbf{D}^2) \sim m + \mathbf{D}\mathcal{N}(\mathbf{0}, \mathbf{I})$ $\quad$ $\mathcal{N}(m, \mathbf{C}) \sim m + \mathbf{C}^{\frac{1}{2}}\mathcal{N}(\mathbf{0}, \mathbf{I})$

Variants involving various numbers of parameters

[Hansen, PPSN 2006]

Control parameters

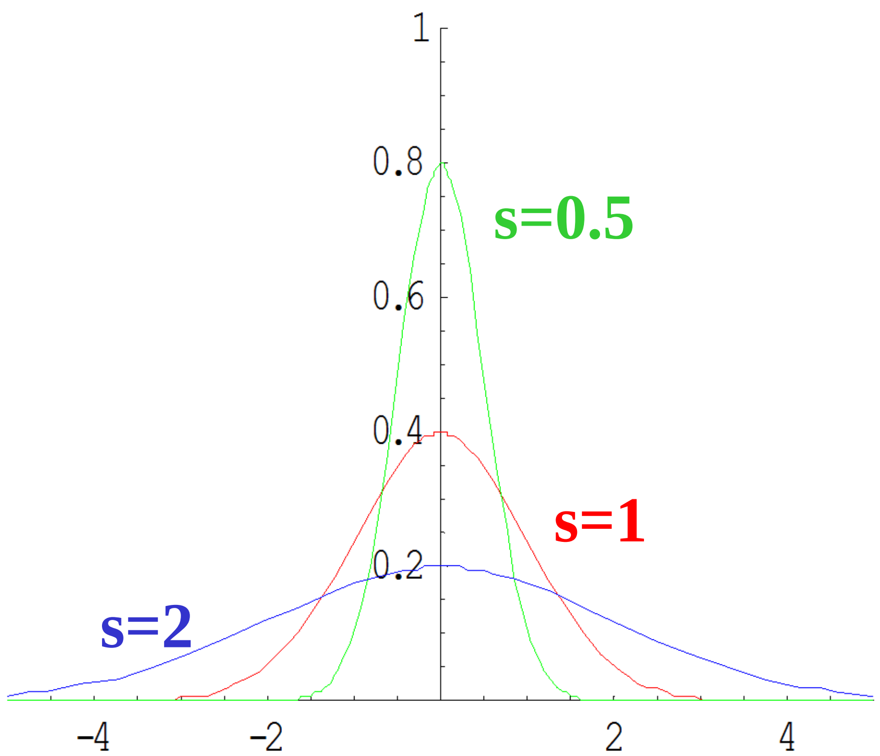
Problem: choice of the control parameters

Example: perturbation of type N(0,s)

- s large -> large perturbation
- s small -> small perturbation

Solutions:

- Adaptive heuristic methods (example: rule 1/5)
- Self-adaptation (change of parameters by recombination and mutation)



Adaptation

1/5 rule.

This is an heuristic rules developed for ES having independent perturbations characterized by a single parameter, s.

Idea: s is adjusted by using the success ratio of the mutation

The success ratio:

p_s= number of mutations leading to better configurations /
total number of mutations

Rmk. 1. The success ratio is estimated by using the results of at least n mutations (n is the problem size)

2. This rule has been initially proposed for trajectory based ES (just one element in population)

Adaptation

1/5 Rule.

$$s' = \begin{cases} s/c & \text{if } p_s > 1/5 \quad \text{(increase of s)} \\ cs & \text{if } p_s < 1/5 \quad \text{(decrease of s)} \\ s & \text{if } p_s = 1/5 \quad \text{(same level of s)} \end{cases}$$

Some theoretical studies conducted for some particular objective functions (e.g. sphere function) led to the remark that c should satisfy 0.8 <= c<1 (e.g.: c=0.817)

Remarks:

- This rule was proposed for ESs involving just one candidate; it cannot be directly extended in the case of populations of candidates

Self-adaptation

Self-adaptation

Idea:

- Extend the elements of the population with components corresponding to the control parameters
- Apply specific recombination and mutation operators also to control parameters
- Thus the values of control parameters leading to competitive individuals will have higher chance to survive

Extended population elements (depending on the perturbation used in mutation:
- 1 parameter
- n parameters
- n(n+1)/2 parameters

$$\bar{x} = (x_1, \ldots, x_n, s)$$

$$\bar{x} = (x_1, \ldots, x_n, s_1, \ldots, s_n)$$

$$\bar{x} = (x_1, \ldots, x_n, s_1, \ldots, s_n, a_1, \ldots, a_{n(n-1)/2})$$

Self-adaptation

Steps:

- Adjust the control parameters (by applying specific operators)
- Change the decision variables (using the modified control parameters)

Example: the case of independent perturbations with different distributions

$$\bar{x} = (x_1, \ldots, x_n, s_1, \ldots, s_n)$$

$$s_i^{'} = s_i \exp(r) \exp(r_i),$$

Variables with lognormal distribution

- ensure that $s_i > 0$
- it is symmetric around 1

$$r \in N(0, 1/\sqrt{2n}), r_i \in N(0, 1/\sqrt{2\sqrt{n}})$$

$$x_i^{'} = x_i + s_i^{'} z \quad \text{with} \quad z \in N(0,1)$$

Remark:

- The recommended recombination for the control parameters is the intermediate recombination

Other adaptation variants

Variant proposed by Michalewicz (1996):

$$x_i^{'}(t) = \begin{cases} x_i(t) + \Delta(t, b_i - x_i(t)) & \text{if } u < 0.5 \\ x_i(t) - \Delta(t, x_i(t) - a_i) & \text{if } u \geq 0.5 \end{cases}$$

$$\Delta(t, y) = y \cdot u \cdot (1 - t/T)^p, \ \mathrm{p} > 0$$

- a_i and b_i are the bounds of the interval corresponding to component x_i
- u is a random value in (0,1)
- t is the iteration counter
- T is the maximal number of iterations

Rmk: the use of the maximal number of iterations to control the amount of iterations is used in many metaheuristics

Other adaptation variants

CMA – ES (Covariance Matrix Adaptation –ES) [Hansen, 1996]

Initialize $m \in \mathbb{R}^n$, $\sigma \in \mathbb{R}_+$, $\mathbf{C} = \mathbf{I}$, and $p_c = \mathbf{0}$, $p_\sigma = \mathbf{0}$,
set $c_c \approx 4/n$, $c_\sigma \approx 4/n$, $c_{cov} \approx \mu_{eff}/n^2$, $\mu_{cov} = \mu_{eff}$, $d_\sigma \approx 1 + \sqrt{\frac{\mu_{eff}}{n}}$,
λ, and $w_i, i = 1, \ldots, \mu$ such that $\mu_{eff} \approx 0.3\,\lambda$, where $\mu_{eff} = \frac{1}{\sum_{i=1}^{\mu} {w_i}^2}$

While not terminate

$$x_i = m + \sigma z_i, \quad z_i \sim \mathcal{N}_i(\mathbf{0}, \mathbf{C}),$$ sampling

$$m \leftarrow m + \sigma \langle z \rangle_{sel} \quad \text{where } \langle z \rangle_{sel} = \sum_{i=1}^{\mu} w_i z_{i:\lambda}$$ update mean

$$p_c \leftarrow (1 - c_c) p_c + \mathbb{1}_{\{\|p_\sigma\| < 1.5\sqrt{n}\}} \sqrt{1 - (1 - c_c)^2} \sqrt{\mu_{eff}} \langle z \rangle_{sel}$$ cumulation for $\mathbf{C}$

$$\mathbf{C} \leftarrow (1 - c_{cov}) \mathbf{C} + c_{cov} \frac{1}{\mu_{cov}} p_c {p_c}^T + c_{cov} \left(1 - \frac{1}{\mu_{cov}}\right) \mathbf{Z} \quad \text{where } \mathbf{Z} = \sum_{i=1}^{\mu} w_i z_{i:\lambda} z_{i:\lambda}^T$$ update $\mathbf{C}$

$$p_\sigma \leftarrow (1 - c_\sigma) p_\sigma + \sqrt{1 - (1 - c_\sigma)^2} \sqrt{\mu_{eff}}\, \mathbf{C}^{-\frac{1}{2}} \langle z \rangle_{sel}$$ cumulation for σ

$$\sigma \leftarrow \sigma \times \exp\left(\frac{c_\sigma}{d_\sigma}\left(\frac{\|p_\sigma\|}{\mathsf{E}\|\mathcal{N}(\mathbf{0}, \mathbf{I})\|} - 1\right)\right)$$ update of σ

Survivors selection

Variants:

(μ, λ)

From the set of μ parents construct λ> μ offspring and starting from these select the best μ survivors (the number of offspring should be larger than the number of parents)

$(\mu + \lambda)$

From the set of μ parents construct λ offspring and from the joined population of parents and offspring select the best μ survivors (truncation selection). This is an elitist selection (it preserves the best element in the population)

Remark: if the number of parents is rho the usual notations are:

$(\mu / \rho + \lambda)$ $(\mu / \rho, \lambda)$

Survivors selection

Particular cases:

(1+1) – from one parent generate one offspring and chose the best one

(1,/+λ) – from one parent generate several offspring and choose the best element

(μ+1) – from a set of μ construct an offspring and insert it into population if it is better than the worst element in the population

Survivors selection

The variant (μ+1) corresponds to the so called steady state (asynchronous) strategy

Generational strategy:

- At each generation is constructed a new population of offspring
- The selection is applied to the offspring or to the joined population
- This is a synchronous process

Steady state strategy:

- At each iteration only one offspring is generated; it is assimilated into population if it is good enough
- This is an asynchronous process

ES variants

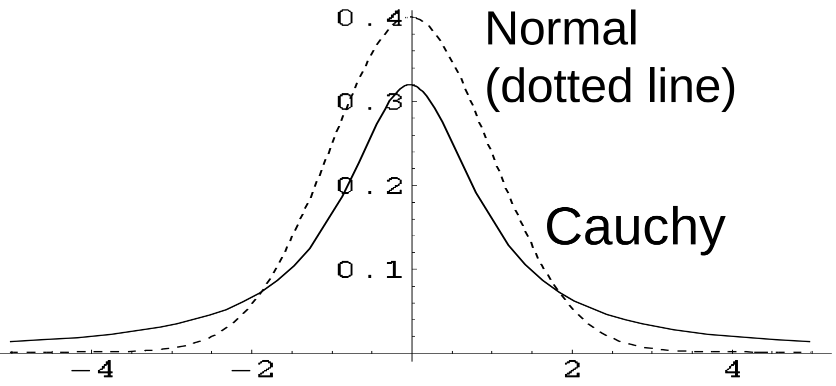
(μ, k, λ, ρ) strategies

Each element has a limited life time (k generations)

The recombination is based on ρ parents

Fast evolution strategies:

The perturbation is based on the Cauchy distribution



$$pdf(x) = \frac{s}{\pi(x^2 + s^2)}$$

Simulation:

- Ratio of two random variables with standard normal distribution (N(0,1))
- Rejection method: u/v

where u and v are uniformly distributed in [-1,1] and $u^2+v^2<=1$

Analysis of the behavior of ES

Evaluation criteria:

Effectiveness:

- Value of the objective function after a given number of evaluations (nfe)

Success ratio:

- The number of runs in which the algorithm reaches the goal divided by the total number of runs.

Efficiency:

- The number of evaluation functions necessary such that the objective function reaches a given value (a desired accuracy)

Summary

Encoding	Real vectors
Recombination	Discrete or intermediate
Mutation	Random additive perturbation (uniform, Gaussian, Cauchy)
Parents selection	Uniformly random
Survivors selection	(μ,λ) or $(\mu+\lambda)$
Particularity	Self-adaptive mutation parameters