I Order statistics - Effect of selection

We want to illustrate the effect of selection on the distribution of candidate solutions in a stochastic algorithm. More precisely we consider a $(1, \lambda)$-ES algorithm whose state is given by $X_t \in \mathbb{R}^n$. At each iteration $t$, $\lambda$ candidate solutions are sampled according to

$$X_{t+1}^i = X_t + U_{t+1}^i$$

with $(U_{t+1}^i)_{1 \leq i \leq \lambda}$ i.i.d. and $U_{t+1}^i \sim \mathcal{N}(0, I_d)$. Those candidate solutions are evaluated on the function $f : \mathbb{R}^n \to \mathbb{R}$ to be minimized and then ranked according the their $f$ values:

$$f(X_{1,\lambda}^{t+1}) \leq \cdots \leq f(X_{\lambda,\lambda}^{t+1})$$

where $i: \lambda$ denotes the index of the $i$th best candidate solution. The best candidate solution is then selected that is

$$X_{t+1} = X_{1,\lambda}^{t+1}.$$ 

We will compute for the linear function $f(x) = x_1$ to be minimized the conditional distribution of $X_{1,\lambda}^{t+1}$ (i.e. after selection) and compare it to the distribution of $X_{1,\lambda}^{t}$ (i.e. before selection).

1. What is the distribution of $X_{1,\lambda}^{t+1}$ conditional to $X_t$? Deduce the density of each coordinate of $X_{1,\lambda}^{t+1}$.

We remind that given $\lambda$ random variables independent and identically distributed $Y_1, Y_2, \ldots, Y_\lambda$, the order statistics $Y_{(1)}, Y_{(2)}, \ldots, Y_{(\lambda)}$ are random variables defined by sorting the realizations of $Y_1, Y_2, \ldots, Y_\lambda$ in increasing order. We consider that each random variable $Y_i$ admits a density $f(x)$ and we denote $F(x)$ the cumulative distribution function, that is $F(x) = \Pr(Y \leq x)$.

2. Compute the cumulative distribution of $Y_{(1)}$ and deduce the density of $Y_{(1)}$.

3. Let $U_{1,\lambda}^{t+1}$ be the random vector such that

$$X_{1,\lambda}^{t+1} = X_t + U_{1,\lambda}^{t+1}$$

Express for the minimization of the linear function $f(x) = x_1$, the first coordinate of $U_{1,\lambda}^{t+1}$ as an order statistic.

4. Deduce the conditional distribution and conditional density of the random vector $X_{1,\lambda}^{t+1}$. 

II Adaptive step-size algorithms

The implementations can be done in the programming language your prefer (Matlab/Python, ...)

We are going to test the convergence of several algorithms on some test functions, in particular on the so-called sphere function

\[ f_{\text{sphere}}(x) = \sum_{i=1}^{n} x_i^2 \]

and the ellipsoid function

\[ f_{\text{elli}}(x) = \sum_{i=1}^{n} (100 \frac{i-1}{n-1} x_i)^2. \]

1. What is the condition number associated to the Hessian matrix of the functions above? Are the functions ill-conditioned?

2. Implement the functions.

The (1 + 1)-ES algorithm is one of the simplest stochastic search methods for numerical optimization. We will start by implementing a (1 + 1)-ES with constant step-size. The pseudo-code of the algorithm is given by

Initialize \( x \in \mathbb{R}^n \) and \( \sigma > 0 \)
while not terminate
\[ x' = x + \sigma N(0, I) \]
if \( f(x') \leq f(x) \)
\[ x = x' \]
\[ \sigma = 1.5 \sigma \]
else
\[ \sigma = (1.5)^{-1/4} \sigma \]

where \( N(0, I) \) denotes a Gaussian vector with mean \( 0 \) and covariance matrix equal to the identity.

1. Implement the algorithm. You can write a function that takes as input an initial vector \( x \), an initial step-size \( \sigma \) and a maximum number of function evaluations and returns a vector where you have recorded at each iteration the best objective function value.

2. Use the algorithm to minimize the sphere function in dimension \( n = 5 \). We will take as initial search point \( x^0 = (1, \ldots, 1) \) and initial step-size \( \sigma = 10^{-3} \) and stopping criterion a maximum number of function evaluations equal to \( 2 \times 10^4 \).

3. Plot the evolution of the function value of the best solution versus the number of iterations (or function evaluations). We will use a log scale for the y-axis (semilogy).

4. Explain the three phases observed on the figure.

To accelerate the convergence, we will implement a step-size adaptive algorithm, i.e. \( \sigma \) is not fixed once for all. The method to adapt the step-size is called one-fifth success rule. The pseudo-code of the (1 + 1)-ES with one-fifth success rule is given by:

Initialize \( x \in \mathbb{R}^n \) and \( \sigma > 0 \)
while not terminate
\[ x' = x + \sigma N(0, I) \]
if \( f(x') \leq f(x) \)
\[ x = x' \]
\[ \sigma = 1.5 \sigma \]
else
\[ \sigma = (1.5)^{-1/4} \sigma \]
5. Implement the (1+1)-ES with one-fifth success rule and test the algorithm on the sphere function $f_{\text{sphere}}(x)$ in dimension 5 ($n = 5$) using $x^0 = (1,\ldots,1)$, $\sigma_0 = 10^{-3}$ and as stopping criterion a maximum number of function evaluations equal to $6 \times 10^2$. Plot the evolution of the square root of the best function value at each iteration versus the number of iterations. Use a logarithmic scale for the y-axis. Compare to the plot obtained on Question 3. Plot also on the same graph the evolution of the step-size.

6. Use the algorithm to minimize the function $f_{\text{elli}}$ in dimension $n = 5$. Plot the evolution of the objective function value of the best solution versus the number of iterations. Why is the (1 + 1)-ES with one-fifth success much slower on $f_{\text{elli}}$ than on $f_{\text{sphere}}$?

7. Same question with the function

$$f_{\text{Rosenbrock}}(x) = \sum_{i=1}^{n-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2).$$

8. We now consider the functions, $g(f_{\text{sphere}})$ and $g(f_{\text{elli}})$ where $g : \mathbb{R} \to \mathbb{R}, y \mapsto y^{1/4}$. Modify your implementation in Questions 5 and 6 so as to save at each iteration the distance between $x$ and the optimum. Plot the evolution of the distance to the optimum versus the number of function evaluations on the functions $f_{\text{sphere}}$ and $g(f_{\text{sphere}})$ as well as on the functions $f_{\text{elli}}$ and $g(f_{\text{elli}})$. What do you observe? Explain.