Exercise 8: Performance of the Simple Evolution Strategy

Summer Term 2024

This exercise explores to what extent the performance of an evolutionary algorithm depends on its basic parameters.

Review: What are the basic parameters that *configure* an evolution strategy?

1.
2.
3.

Please, recall how the mutation operator distributes offspring around the parent.

To Do: Basically, this exercise requires two different types of work. First, you have to adapt your program, and then you have to run some experiments. You may complete the table printed below.

Tasks:

1. Recycle the first version of your personal implementation of the simple evolution strategy. This (test) version was using the sphere model $f(x_1, \ldots, x_n) = \sum_i x_i^2$ as the fitness function.

2. *Theoretical work:* Assume, for the moment, that you have just one parent, i.e., $\mu = 1$. Now, discuss how an increasing number λ of offspring changes their distribution around the parent. How does this affect the rate of progress? For this discussion, the rate of progress $\varphi = (f(\vec{x}^t - f(\vec{x}^{t+1}))/f(\vec{x}^t)$ might be defined as the normalized change in the fitness values. Please remind yourself that an evolutionary algorithm is a stochastic optimization procedure, and that you can thus give only statistical, i.e., average, answers.

   Now, keep the number λ of offspring fixed and vary the number μ of parents. How do you think that this affects the rate of progress? Also, please diskuss how you can measure performance/rate of progress?

3. *Practical work:* Validate your theoretical results by running some experiments by using the quadratic fitness function. Modify the `main()`-routine, such that it executes everything from `make_populations()` to `destroy_populations()` a certain number times (e.g., 10 times). In each of these runs, the evolution strategy has to work until the fitness has dropped below 1 % of its initial fitness value. Complete the following two tables. In order to be able to compare you results with those from your classmates, use the following parameter settings: problem size of $n = 100$, initialize all parameters to $x_i = 10$, and use the step size $\sigma = 1$.

$\mu = 1$ parent:

λ	2	3	4	5	6	7	8	10	20	40
generations										
time										

$\lambda = 20$ offspring:

μ	1	2	4	6	8	10
generations						
time						

Questions:

(a) What are the optimal values for μ and λ?

(b) Why didn't you try a (1,1)-evolution strategy? What would its performance be?

4. Finally, you are asked to investigate the difference between a (μ, λ)-evolution strategy and $(\mu+\lambda)$-evolution strategy. Guess, which one will be faster? What are the reasons for your opinion?

   Run a (1,6)-evolution strategy on an $n = 100$ dimensional sphere for 500 generations. Again, use problem size of $n = 100$, initialize all parameters to $x_i = 10$, and use the step size $\sigma = 1$. In every generation, report, i.e., print onto the screen, both the generation number and the parent's fitness value. Save the output into a file. Run the program again but this time a (1+6)-evolution strategy and do the same subsequent steps. Generate a comparative plot. On Linux, you can use `gnuplot` and on Windows, `excel`.

Have fun, Theo and Ralf.