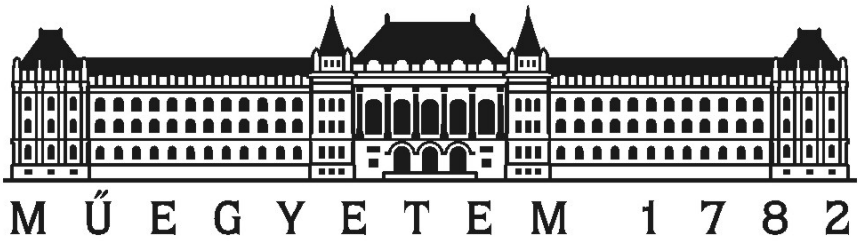


Evolutionary Algorithms

Homework

First task

Author
Carlos Sánchez Páez
http://www.github.com/csp98

Budapest University of Technology and Economics

Academic year 2018-2019

1. Let us suppose, that the run-time of an algorithm is monotone increasing in n, which is the length of the input. Investigate the following statements. Prove the true statement(s) and give counterexample(s) to the false one(s):

   (a) If we plot the run-time on log-log scale than we obtain a monoton increasing function.

   The run-time of the algorithm is given by the function $f(n)$. As $f(n)$ is monotone increasing, $f'(n) > 0$ for every $n \in \mathbb{N}$.

$$T(n) = f(n)$$
$$log(T(n)) = log(f(n))$$
$$\frac{1}{T(n)} \cdot T'(n) = \frac{1}{f(n)} \cdot f'(n) \quad \text{(by derivating)}$$
$$T'(n) = \frac{1}{f(n)} \cdot T(n) \cdot f'(n)$$

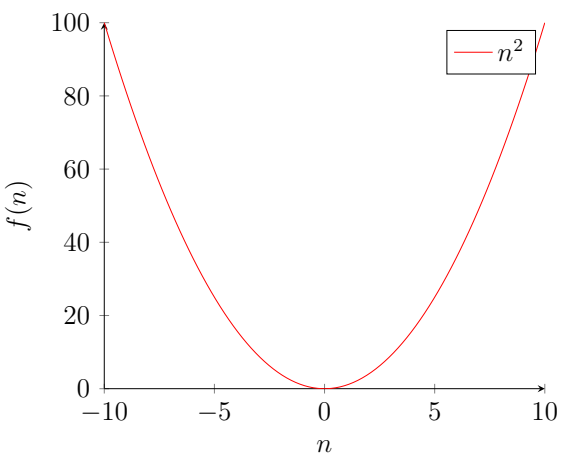
   We can se how $T'(n) > 0$ for every $n \in \mathbb{N}$ (all the members in the product are). This property defines a monotone increasing function, so we can conclude that the statement is **true**.

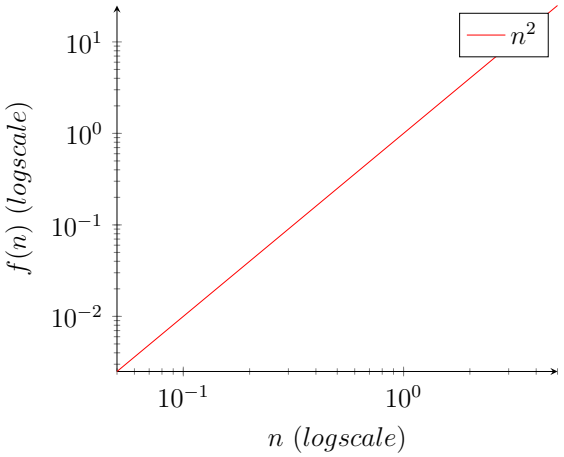
   (b) If the run-time is not only monotone increasing, but also convex, then the plot on log-log scale will be convex.

   Now we have another condition: the function is convex. That means $f''(n) > 0$ for every $n \in \mathbb{N}$. We will proceed in the same way but derivating twice:

$$T(n) = f(n)$$
$$log(T(n)) = log(f(n))$$
$$\frac{1}{T(n)} \cdot T'(n) = \frac{1}{f(n)} \cdot f'(n) \quad \text{(by derivating)}$$
$$T'(n) = \frac{1}{f(n)} \cdot T(n) \cdot f'(n)$$
$$T''(n) = \frac{f(n) \cdot [T'(n) \cdot f'(n) + f''(n) \cdot T(n)] - f'(n) \cdot T(n) \cdot f'(n)}{f^2(n)} \quad \text{(by derivating again)}$$

   We can see that we have a negative part in the numerator ($[f'(n)]^2 \cdot T(n)$), so the logscale plot won't be always convex. That means the statement is **false**. We can see a counterexample:





As we can see, n^2 is convex but its logscale plot isn't.

2. Let us suppose, that two possible solution are coded with 0101010 and 0001000. If they are the parents can any of their offspring be:

   - 0101010
   - 1111111
   - 0000000

   in case we use:

   - onepoint-crossover ?
   - multiplepoints-crossover ?
   - uniform-crossover ?

Offspring	**Onepoint-crossover**	**Multiplepoints-crossover**	**Uniform-crossover**
0101010	Yes, if we put the breakpoint in the first position (0**101010** + **0**001000)	Yes. If we put the breakpoints in the first and last position (0**10101**0 + **0**00100**0**)	Yes, if the probability chooses the right parent in the right moment. Example: 0**10101**0 + **0**0**1**0**0**0
1111111	No. In many positions there are not ones in the parents.	No. In many positions there are not ones in the parents.	No. In many positions there are not ones in the parents.
0000000	No. In some positions there are not zeros in the parents.	No, because in the 4^{th} position both parents have a one.	No, because in the 4^{th} position both parents have a one.

3. How can we represent with a fixed length 0-1 series the solutions of the following problem?

   We have an n $\times$ n sized grid, and on each edge a real positive number. We are looking for a path from the upper left corner point to the lower right point, which goes only to the right or down, where the sum of the numbers along the path is minimal. Write a program (using your favorite programming language) to solve this problem using a suitable genetic algorithm.

   The representation of the problem will be the following:

   - **0** = Move right
   - **1** = Move down

   Each individual will have a genotype (size $2n-2$ because to get to the lower right corner you must go $n-1$ positions down plus $n-1$ positions to the right, so $2 \cdot (n-1) = 2n-2$) which indicates the steps that have to be completed to reach the destination. The fitness funcion is the sum of the costs of the points crossed during the path (it has to be minimized).

   In my algorithm I use the onepoint-crossover technique for the reproduction. The population size is 10 and the mutation chance, 20%.

Bibliography

[1] Course Webpage
`http://math.bme.hu/ safaro/evolalgen.html`

[2] `http://www.rubicite.com/Tutorials/GeneticAlgorithms.aspx`

[3] `https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a`

[4] `https://docs.python.org/3/`

[5] `https://tex.stackexchange.com/`