Lecture 3 (for self-implementation)

Monte Carlo Simulations, Genetic Algorithms

TASK 1

Use a genetic algorithm to find an approximation of the square root of 2 with the given number of digits *n*. An exemplary solution is given in file *sqrt_genetic.r*.

TASK 2

Use the spreadsheet to estimate the value of π by randomly choosing *n* coordinates from the unit square and calculating how many points will hit the circle inscribed in the square. The number of these points to all points ratio approximates $\frac{\pi}{4}$.

TASK 3

Use R to estimate the value of π by randomly choosing *n* coordinates from the unit square and calculating how many points will hit the circle inscribed in the square. The number of these points to all points ratio is the approximation of $\frac{\pi}{4}$. Measure the time of the program depending on *n* if:

1. we choose *n* random points in the loop (one by one),
2. we choose *n* random points in the vector (all at once),
3. we choose *n* random points group by the vectors of length *m*.

Check:

- How many points are needed to calculate π with accuracy to 1, 2, and 3 digits after the decimal point?
- Which of the three methods of calculating π is fastest depending on *n*?
- Is initialization of the variables (memory allocation) before the loop influence the speed of the computations?