

# 732A90: Computational Statistics

Computer lab6 - Group11

*Sofie Jörgensen, Oriol Garrobé Guilera, David Hrabovszki*

*02 March 2020*

## Question 1: Genetic algorithm

In this exercise we are going to perform one-dimensional maximization by using a genetic algorithm.

1.

Firstly, we define the function `f()` as

$$f(x) := \frac{x^2}{e^x} - 2 \exp(-(9 \sin x)/(x^2 + x + 1)).$$

2.

Secondly, we define the function `crossover()`, that takes two scalars  $x$  and  $y$  as inputs, and returns a child as  $\frac{x+y}{2}$ .

3.

Thirdly, we define the function `mutate()`, that performs the integer division  $x^2 \bmod 30$ , for a scalar input  $x$ .

4.

Further, we will create a function called `genetic()`, with the parameters `maxiter` and `mutprob`. The settings of this `genetic()` function, as well as its output results, are presented in (a)-(e). The code can be found in the Appendix.

(a). The function  $f()$  is plotted in the range from 0 to 30 in Figure X, and we can observe that there is a maximum value located around  $x = 1$ .

(b). An initial population for the genetic algorithm is defined as  $X = (0, 5, 10, 15, \dots, 30)$ .

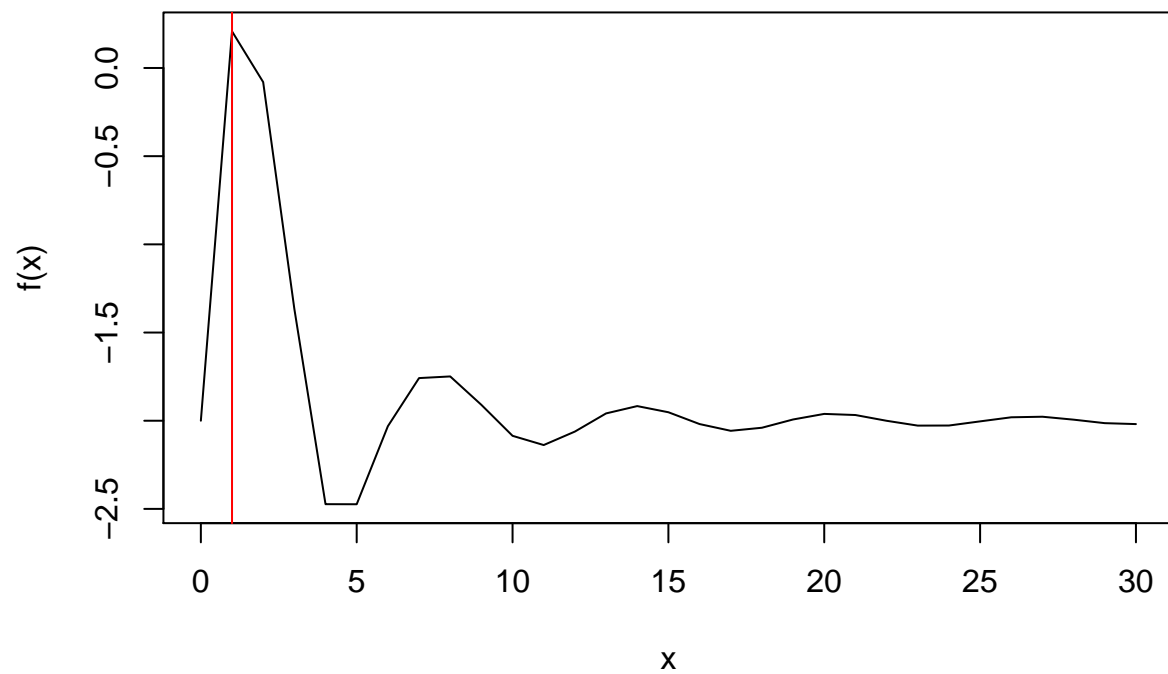
(c). A vector called `Values` are computed, containing the function values for each population point.

(d). The `genetic()` function performs `maxiter` iterations. For each iteration...

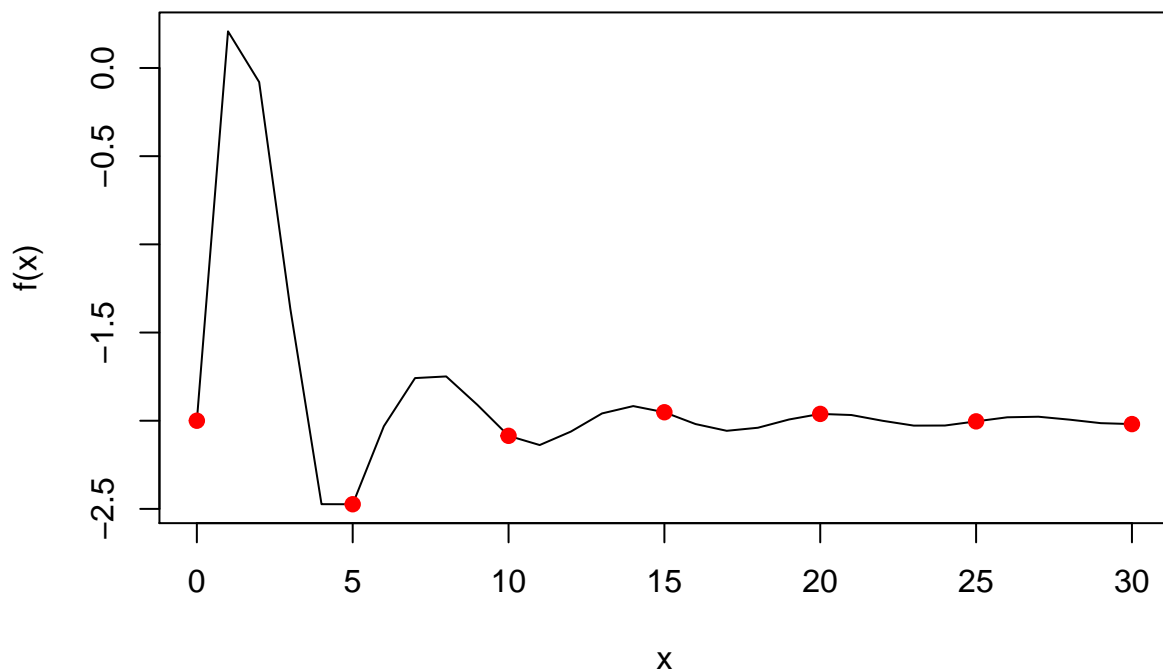
(e).

5.

By using the defined functions from previous tasks (1.1-1.4), we are going to observe the initial population and final population. This is done by running the code with different combinations of `maxiter`= 10, 100 and `mutprob`= 0.1, 0.5, 0.9.



```
## [1]  0  5 10 15 20 25 30
## [1] -2.000000 -2.473573 -2.085654 -1.951947 -1.961344 -2.003663 -2.019194
```



## Question 2: EM algorithm

The purpose with this exercise is to implement the EM algorithm. For this, we are given the data file `physical1.csv`, containing a behavior of two related physical processes  $Y = Y(X)$  and  $Z = Z(X)$ .

1.

The first step is to examine the data set `physical1.csv`, to see if the two processes are related to each other.

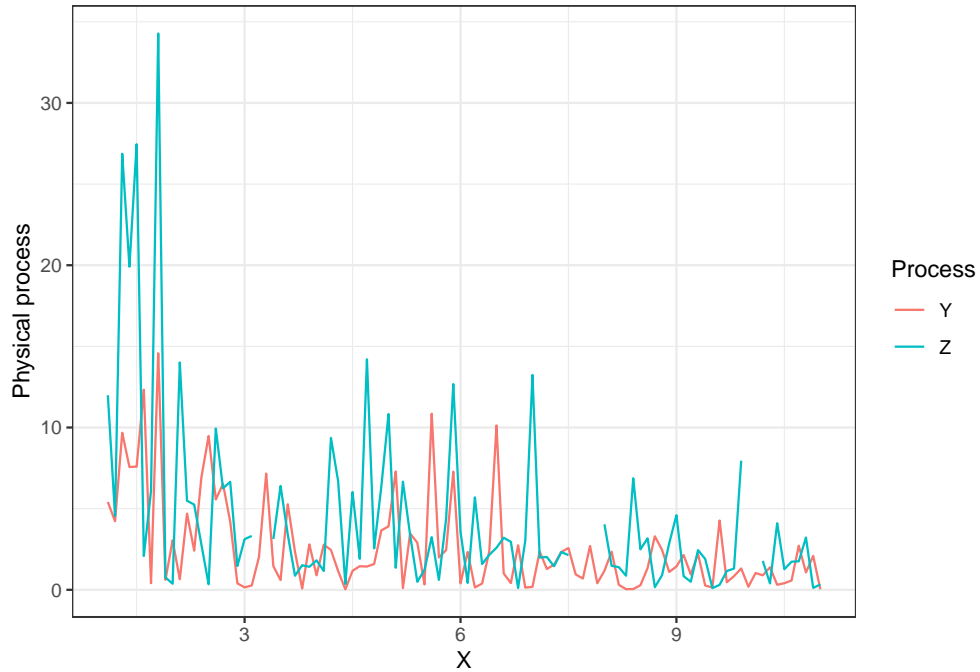


Figure 1: Time series plot of the dependence of Z and Y versus X.

In Figure 1 it seems that the two processes are related to each other, with respect to  $X$ , since the graphs follows similar patterns. We can also observe that the physical process  $Z$  has a greater variation, especially at the beginning of the series, but also in general, compared to the process  $Y$ .

2.

3.

4.

---

## Appendix

```
knitr::opts_chunk$set(echo = FALSE)
# R version
RNGversion('3.5.1')
library("ggplot2")
#1.1
f <- function(x){
  return(x^2/exp(x) - 2*exp(-1*(9*sin(x)) / (x^2 + x + 1)))
}

#1.2
crossover <- function(x,y){
  return((x+y) / 2)
}

#1.3
```

```

mutate <- function(x){
  return(x^2 %% 30)
}
#4
#4
genetic <- function(maxiter, mutprob){
  #a
  plot(x = seq(0,30), y = f(seq(0,30)), type = "l", xlab = "x", ylab = "f(x)")
  abline(v=seq(0,30)[which.max(f(seq(0,30)))], col="red" )

  #b
  X = seq(0,30,5)

  #c
  Values = f(X)

  #d
  #set seed
  set.seed(1234567890)
  for (i in 1:maxiter) {
    #i
    parents = match(sample(X, 2),X)

    #ii
    victim = order(Values)[1]

    #iii
    kid = round(crossover(parents[1],parents[2]))
    p = runif(1)
    if (p < mutprob) {
      kid = mutate(kid)
    }

    #iv
    X[victim] = kid
    Values = f(X)

    #v
    max = max(Values)
  }

  #e
  print(X)
  print(Values)
  plot(x = seq(0,30), y = f(seq(0,30)), type = "l", xlab = "x", ylab = "f(x)")
  points(x = X, y = Values, col = "red", pch = 19)
}

# Just testing no change, i.e. initial population
genetic(1,0)
# 1.1
physical <- read.csv2("physical1.csv", sep = ",")

```

```

X <- as.numeric(as.character(physical$X))
Y <- as.numeric(as.character(physical$Y))
Z <- as.numeric(as.character(physical$Z))

data <- data.frame(X = c(X,X), value = c(Y,Z), Process= rep(c("Y","Z"), each= 100))

# var(Z,na.rm = TRUE)
# var(Y,na.rm = TRUE)

# Time series plot
ggplot(data = data, aes(x = X, y = value, col = Process)) +
  geom_line() +
  ylab("Physical process") +
  theme_bw()

# R version
RNGversion('3.5.1')
# Packages

```