Assigment 1 Task 1

Mateusz Kowalewski Piotr Filipowicz

10 11 2019

First task

Non-convex function have many local minima so while optimazing the function you can get stuck in one of them. Here is a function that computes a value z of funtcion

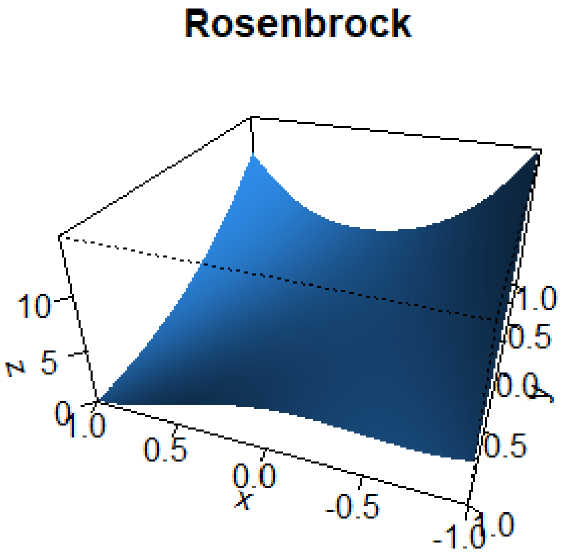
f(x, y) = (1 − x)^2 + e · (y − x2)^2, given an input tuple (x, y).

```
e <-exp(1)
f <- function(x,y)
{
  z <- (1-x)^2 + e*(y-x^2)^2
  z
}
```

There is a method that visualizes the Rosenbrock function in 3D, where x ∈ [−1, 1] and

y ∈ [−1, 1].

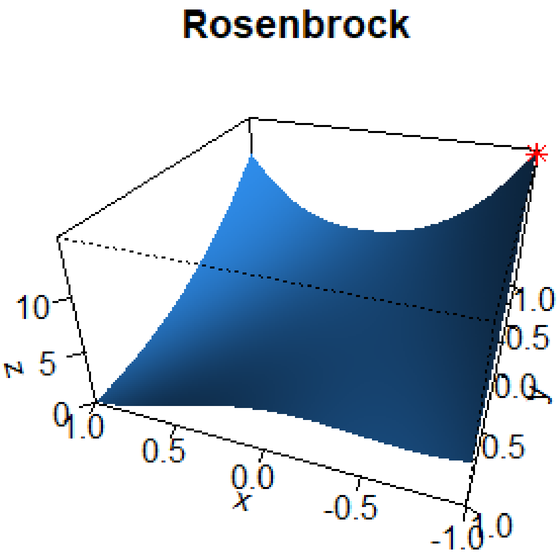
```
e <-exp(1)

f <- function(x,y)
{
  z <- (1-x)^2 + e*(y-x^2)^2
  z
}
x<-seq(from = -1, to = 1, by=0.005)
y<-x
z<- outer(x,y,f)
pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1), scale = TRUE,
            ltheta = -120, shade = 1,box = TRUE, border=NA,main  = "Rosenbrock",ticktype = "detailed")
```



Here is a genetic algorithm that tries to find a global maximum. There is a monitor function that monitor every iteration of algorithm. If you want to use It just add “monitor = monitor” in GA function.

First Version:

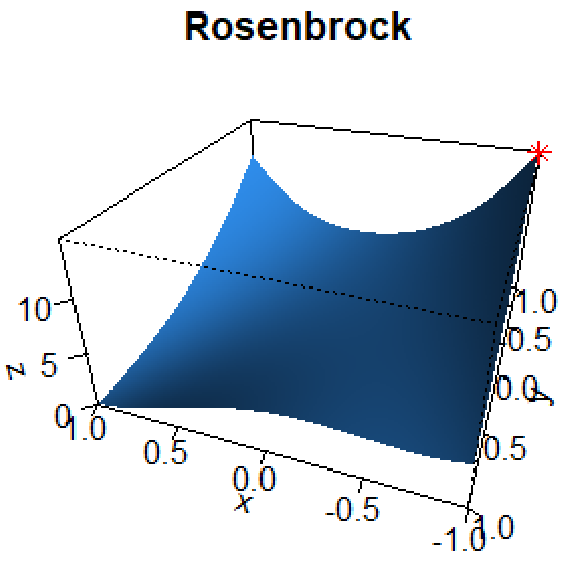
```
fitness <-function(x)
{
  fitness <- ((1-x[1])^2 + e*(x[2]-x[1]^2)^2)
}

x<-seq(from = -1, to = 1, by=0.005)
y<-x
z<- outer(x,y,f)



monitor <- function(obj)
{
  pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1),
scale = TRUE,
              ltheta = -120, shade = 0.5,box = TRUE, border=NA)
  points(trans3d(obj@population[seq(from = 1, to=200, by
=2)],obj@population[seq(from = 2, to=200, by =2)],
                 f(obj@population[seq(from = 1, to=200, by
=2)],obj@population[seq(from = 2, to=200, by =2)]),
                 pmat = pmat),pch = 20,col = "red")
  title(paste("iteration =", obj@iter), font.main = 1)
  Sys.sleep(0.5)
}

GA_1 <-ga(type = "real-valued",fitness = fitness,lower = c(-1,-1),upper = c(1,1),
         popSize = 100, crossover = gareal_spCrossover, mutation =
gareal_raMutation)



pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1), scale
= TRUE,
           ltheta = -120, shade = 1,box = TRUE, border=NA,main  =
"Rosenbrock",ticktype = "detailed")

solution <-
points(trans3d(GA_1@solution[1],GA_1@solution[2],f(GA_1@solution[1],GA_1@soluti
on[2]),
                         pmat = pmat),col = "red",pch=8)
```



```
GA_1@solution

##              x1         x2
## [1,] -0.9961762 -0.9824968

f(GA_1@solution[1],GA_1@solution[2])

## [1] 14.58626
```

Second Version:

```
fitness <-function(x)
{
  fitness <- ((1-x[1])^2 + e*(x[2]-x[1]^2)^2)
}

x<-seq(from = -1, to = 1, by=0.005)
y<-x
z<- outer(x,y,f)

monitor <- function(obj)
{
  pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1),
scale = TRUE,
              ltheta = -120, shade = 0.5,box = TRUE, border=NA)
  points(trans3d(obj@population[seq(from = 1, to=200, by
=2)],obj@population[seq(from = 2, to=200, by =2)],
                 f(obj@population[seq(from = 1, to=200, by
=2)],obj@population[seq(from = 2, to=200, by =2)]),
                 pmat = pmat),pch = 20,col = "red")
  title(paste("iteration =", obj@iter), font.main = 1)
  Sys.sleep(0.5)
}
GA_2 <-ga(type = "real-valued",fitness = fitness,lower = c(-1,-1),upper = c(1,1),
          popSize = 100, crossover = gareal_waCrossover,mutation  =
gareal_nraMutation)



pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1), scale
= TRUE,
            ltheta = -120, shade = 1,box = TRUE, border=NA,main  =
"Rosenbrock",ticktype = "detailed")
my<-
points(trans3d(GA_2@solution[1],GA_2@solution[2],f(GA_2@solution[1],GA_2@soluti
on[2]),
                   pmat = pmat),pch = 8,col = "red")
```

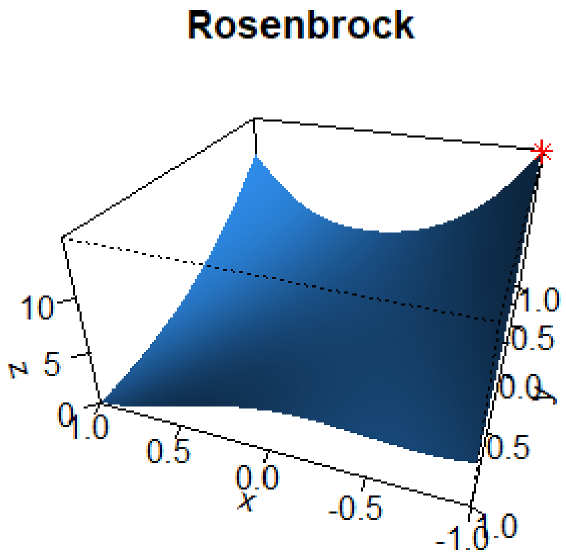


```
GA_2@solution

##              x1         x2
## [1,] -0.9986352 -0.9961029

f(GA_2@solution[1],GA_2@solution[2])

## [1] 14.79576
```

Third Version:

```
fitness <-function(x)
{
  fitness <- ((1-x[1])^2 + e*(x[2]-x[1]^2)^2)
}

x<-seq(from = -1, to = 1, by=0.005)
y<-x
z<- outer(x,y,f)

monitor <- function(obj)
```

```
{
  pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1),
scale = TRUE,
              ltheta = -120, shade = 0.5,box = TRUE, border=NA)
  points(trans3d(obj@population[seq(from = 1, to=200, by
=2)],obj@population[seq(from = 2, to=200, by =2)],
                 f(obj@population[seq(from = 1, to=200, by
=2)],obj@population[seq(from = 2, to=200, by =2)]),
                 pmat = pmat),pch = 20,col = "red")
  title(paste("iteration =", obj@iter), font.main = 1)
  Sys.sleep(0.5)
}
GA_3 <-ga(type = "real-valued",fitness = fitness,lower = c(-1,-1),upper = c(1,1),
          popSize = 100, crossover = gareal_waCrossover,mutation  =
gareal_nraMutation)


pmat<-persp(x, y, z, theta = 200, phi = 30, expand = 0.5,col = rgb(0.2,0.6,1), scale
= TRUE,
            ltheta = -120, shade = 1,box = TRUE, border=NA,main  =
"Rosenbrock",ticktype = "detailed")
my<-
points(trans3d(GA_3@solution[1],GA_3@solution[2],f(GA_3@solution[1],GA_3@soluti
on[2]),
                   pmat = pmat),pch = 8,col = "red")
```



```
GA_3@solution
```

```
##              x1         x2
## [1,] -0.9972357 -0.9982473

f(GA_3@solution[1],GA_3@solution[2])

## [1] 14.78313
```

#Discussion of the results:

If we increase number of iterations we will get better solution but only to around 500 iterations. After that value the performance won’t increase much or will stay similar to that of 500 iterations.For optimal solution 200 interations is enough.

If we decide to increase number of population we will get better solution but only to around 350 population size. After that value the performance won’t almost increase. For optimal solution 100 population size is enough.

A maximum or minimum is said to be local if it is the largest or smallest value of the function, respectively, within a given range. However, a maximum or minimum is said to be global if it is the largest or smallest value of the function, respectively, on the entire domain of a function.