Tarea 7: Búsqueda local

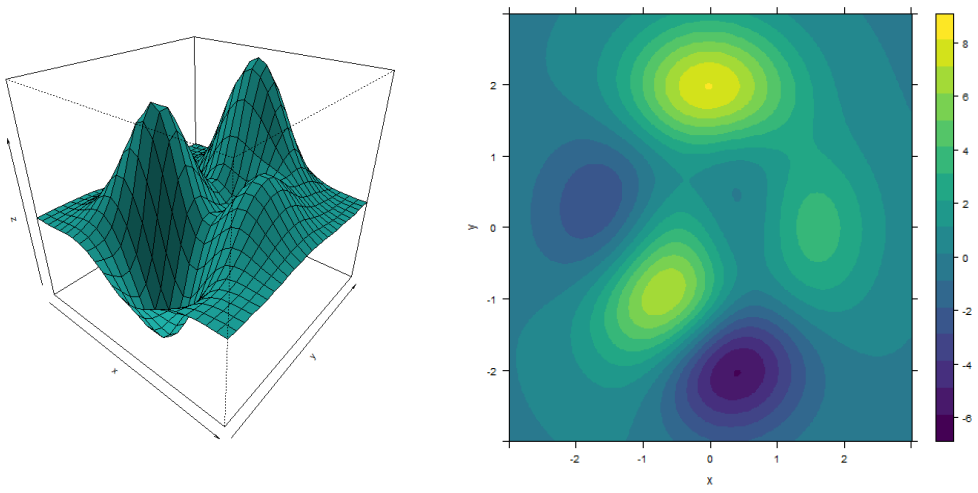
C. María Montemayor Palos

1. Objetivo

Examinar y encontrar el máximo local de la función

$$f(x,y) = 5e^{(-(0{,}8y+1)^2)-(0{,}8x)^2}(0{,}8x-1)^2 - \frac{1}{3}\cdot e^{-(0{,}8x+1)^2-(0{,}8y)^2} + e^{-(0{,}8x)^2-(0{,}8y)^2}(10(0{,}8x)^3 - 2(0{,}8x) + 10(0{,}8y)^5)$$

con restricciones $-3 \leq x, y \leq 3$. La figura 1 muestra la función en tres dimensiones (3D) y dos dimensiones (2D).



(a) (b)

Figura 1. Gráficas de la función.

2. Metodología

Para efectos de la tarea [2] se utiliza el programa R versión 4.0.4 [4] para Windows. Se maximiza la variante de la función bidimensional g(x, y) con las restricciones anteriormente mencionadas, en donde la posición actual es un par x, y y se ocupan dos movimientos aleatorios, Δx y Δy cuyas combinaciones posibles proveen ocho posiciones vecino, de los cuales aquella que logra el mayor valor para g es seleccionado. Se recurre a una visualización en 2D (como se muestra en la figura 2) debido a que en 3D de limita la visibilidad de los puntos generados.

3. Código

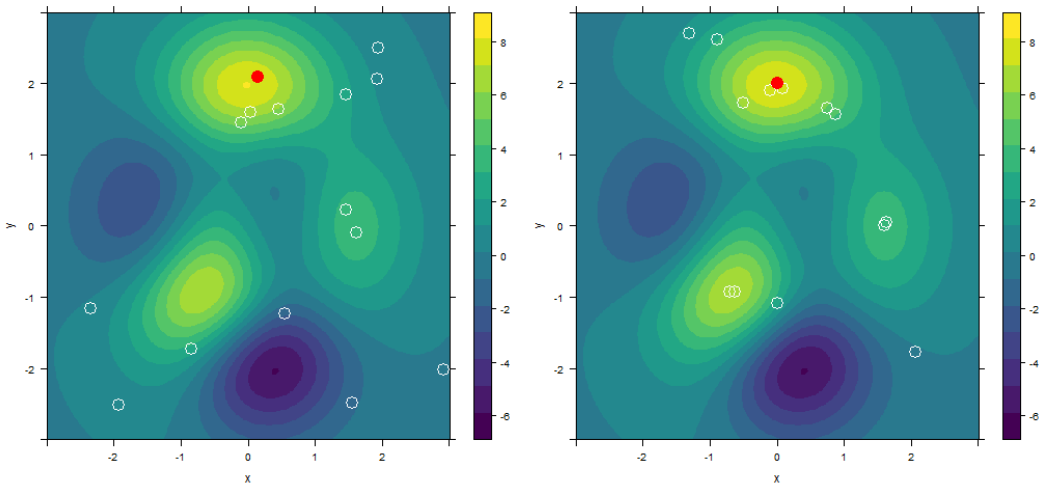
Se modifica el código de Schaeffer [1] y se mejora la calidad del gráfico en apoyo del repositorio de Vázquez [3]. A continuación se muestra la parte del código donde se encuentra la función `replica`.

```
low <- -3
high <- 3
step <- 0.25
replicas <- 15
replica <- function(t){
  curr <- c(runif(1, low, high), runif(1, low, high))
  best <- curr
  for (tiempo in 1:t) {
    delta <- runif(1, 0, step)
    x1 <- curr + c(-delta,0)  #izquierda
    x2 <- curr + c(delta,0)   #derecha
    y1 <- curr + c(0,-delta)  #arriba
    y2 <- curr + c(0,delta)   #abajo

    puntos <- c(x1,x2,y1,y2)
    for(k in 1:8){
      if(puntos[k] < (-3)){
        puntos[k] <- puntos[k]+3
      }
      if(puntos[k] > 3){
        puntos[k] <- puntos[k]-3
      }
    }
    vx <- c()
    vy <- c()
    for(p in 1:8){
      if(p %% 2 == 0){
        vy <- c(vy,puntos[p])
      }else{
        vx <- c(vx,puntos[p])
      }
    }
    vg <- c()
    for(q in 1:4){
      vg <- c(vg, g(vx[q], vy[q]) )
    }
    dm <- which.max(vg)
    curr <- c(vx[dm], vy[dm])
    if(g(curr[1],curr[2]) > g(best[1],best[2])){
      best <- curr
    }
  }
  return(best)
}
```

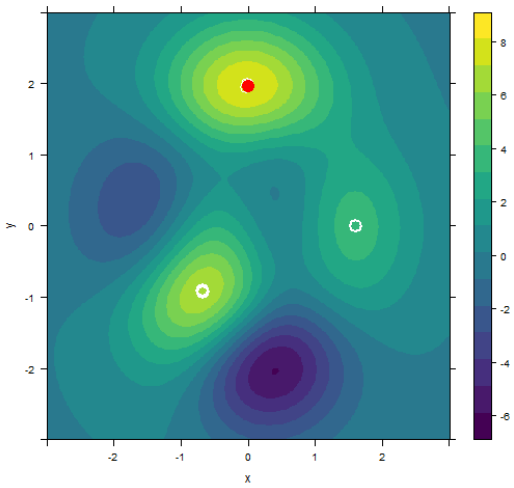
4. Resultados y discusión

En la figura 2 se muestran los resultados obtenidos siendo los círculos blancos obtenidos de las 40 réplicas para los 15 pasos. El punto rojo mostrado en las figuras es el punto al que le corresponde el valor máximo de todas las réplicas generadas. El gif de las gráficas se muestra en el repositorio de Montemayor [5].

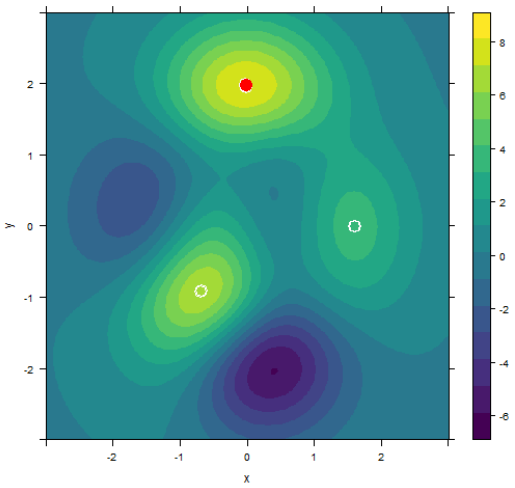


(a) Paso 1.

(b) Paso 10.



(c) Paso 30.



(d) Paso 40.

Figura 2. Búsqueda local de la función.

5. Conclusión

Como es de esperarse, en la figura 2a y 2b se muestra una mayor dispersión de los puntos en cuanto al acercamiento del punto máximo, mientras que en las figuras 2c y 2d tienden a irse a los puntos más altos de la función, esto también depende del pico alto más cercano que esté del punto generado al azar.

Referencias

[1] Schaeffer E. Búsqueda local., 2021. URL https://github.com/satuelisa/Simulation/blob/master/LocalSearch/replicas.R.

[2] Schaeffer E. Práctica 7: Búsqueda local., 2021. URL https://elisa.dyndns-web.com/teaching/comp/par/p7.html.

[3] Vázquez F. Práctica 7: Búsqueda local., 2021. URL https://github.com/fvzqa/Simulacion/blob/master/Tarea07/Tarea7.ipynb.

[4] The R Foundation. The R Project for Statistical Computing, 2021. URL https://www.r-project.org/.

[5] Montemayor M. Práctica 7: Búsqueda local., 2021. URL https://github.com/MariaMontemayor/Simul/tree/main/tarea7.