

Búsqueda Local

3175

19 de marzo de 2019

1. Introducción

Una búsqueda local consiste en buscar determinados valores (ya sean máximos o mínimos) a lo largo de una función tomando en cuenta vecinos en un punto para posteriormente determinar hacia que valor nos queremos desplazar sucesivamente. En la práctica se usaron ejemplos basados en Womersley[2].

2. Objetivo

Observar el comportamiento de una búsqueda local de máximos en una función determinada.

3. Metodología

Usando de base el código proporcionado[1], se busca maximizar es decir, buscar el máximo la función bidimensional $(x + 0,5)^4 - 30 * x^2 - 20 * x + (y + 0,5)^4 - 30 * y^2 - 20 * y / 100$ dentro de un plano de tamaño delimitado por los valores de -3 a 3 donde tomamos en cuenta el valor máximo de z que está en función de las coordenadas de x y y

Se realizaron 100 pasos de un tamaño de 0.25 , esto quiere decir que se tomaron en cuenta márgenes de 0.25 arriba y abajo de cada punto para moverse a una nueva posición con mayor valor de z . Además se realizaron 15 repeticiones de manera simultánea y se fijaron las coordenadas dentro de un `data.frame` para su posterior uso y visualización.

```
1 low <- -3
2 high <- 3
3 step <- 0.25
4 replicas <- 15
5 coordinates <- data.frame("X"=0, "Y"=0, "T"=0, "R" = 0)
```

Para poder buscar un máximo en una función bidimensional se necesita buscar en 8 lugares o vecinos, mediante el cual se utilizó el siguiente criterio:

```
1   if (g(left , currY) > g(right , currY)) {
2     currX <- left
3   } else {
4     currX <- right
5   }
6   if (g(currX , currY) > g(bestX , bestY)) {
7     bestX <- currX
8     bestY <- currY
9   }
10  if (g(currX , down) > g(currX , up)) {
```

```

11     currY <- down
12 } else {
13     currY <- up
14 }
15 if (g(currX, currY) > g(bestX, bestY)) {
16     bestX <- currX
17     bestY <- currY
18 }
19 if (g(left, up) > g(right, up)) {
20     currX <- left
21     currY <- up
22 } else {
23     currX <- right
24     currY <- up
25 }
26 if (g(currX, currY) > g(bestX, bestY)) {
27     bestX <- currX
28     bestY <- currY
29 }
30 if (g(left, down) > g(right, down)) {
31     currX <- left
32     currY <- down
33 } else {
34     currX <- right
35     currY <- up
36 }
37 if (g(currX, currY) > g(bestX, bestY)) {
38     bestX <- currX
39     bestY <- currY

```

Finalmente se realizaron gráficas de los resultados obtenidos, marcando con puntos de colores las nuevas posiciones de las 15 repeticiones buscando el valor máximo de z .

4. Resultados

Los resultados mostrados en las figuras 1 a 3 muestran una vista superior de la $f(x, y)$ durante el inicio, a la mitad y al final de la prueba respectivamente y se puede ver un comportamiento de movimiento inclinado a aproximarse a la zona mas oscura de la gráfica, que representa los valores mas altos de la función. Mediante una imagen .gif que se encuentra en la carpeta del repositorio de la practica se puede apreciar mejor los resultados.

5. Conclusiones

Se puede concluir que la búsqueda local es una muy importante y efectiva herramieta para buscar valores en una función comparable con un conjunto de datos que pueden ser en una o varias dimensiones.

Referencias

- [1] Elisa Schaeffer. Práctica 7: Búsqueda local. Página Web, 2019. URL <https://elisa.dyndns-web.com/teaching/comp/par/p7.html>.

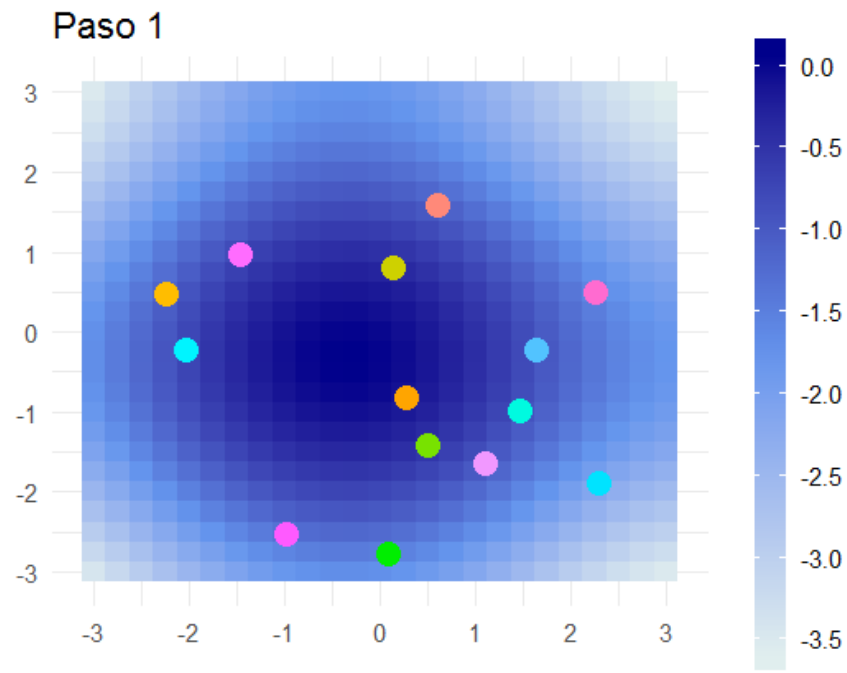


Figura 1: Posición al inicio

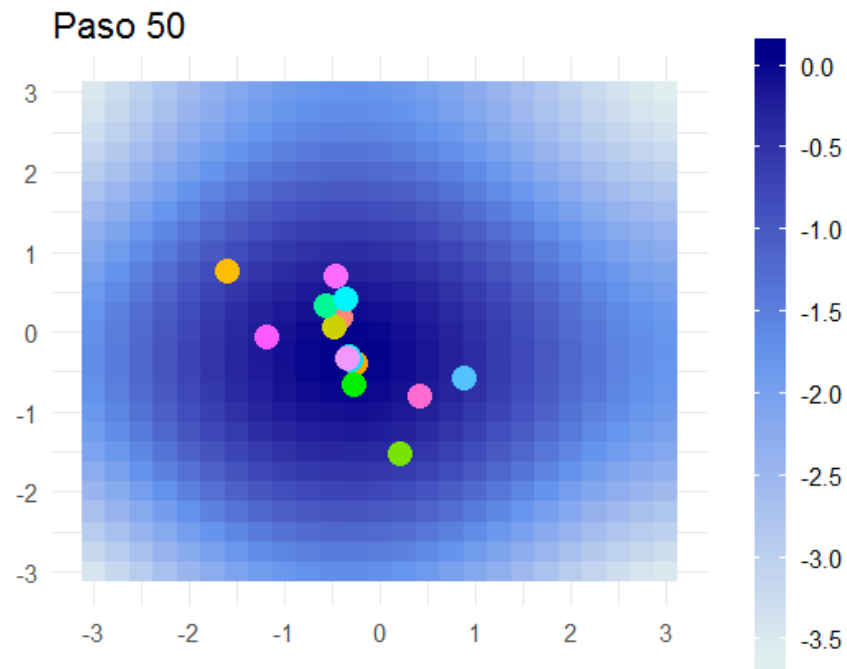


Figura 2: Posición a la mitad, en el paso 50

- [2] Rob Womersley. Local an global optimization. Página Web, 2018. URL <https://web.maths.unsw.edu.au/~rsw/lgopt.pdf>.

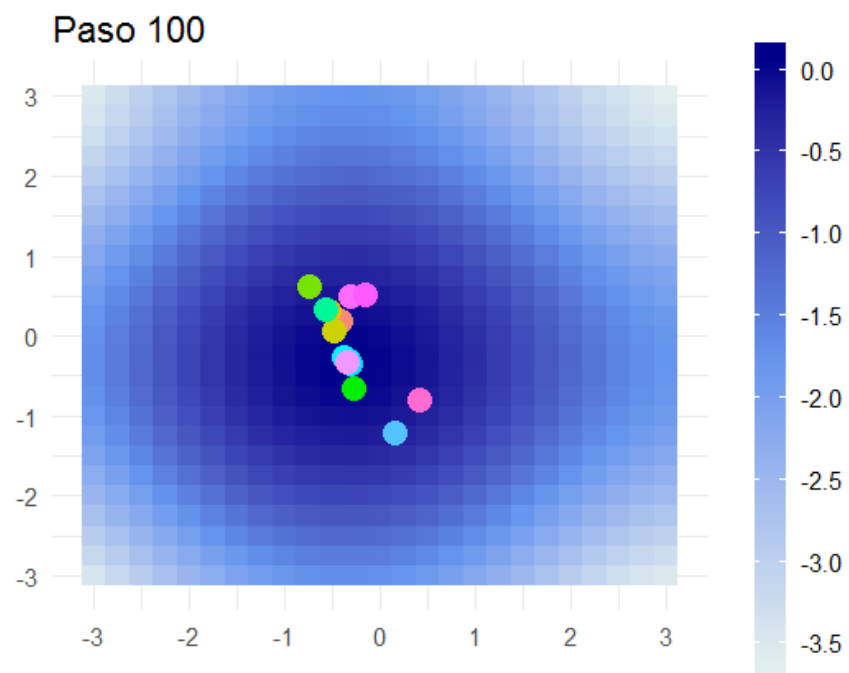


Figura 3: Posición final