

Práctica 7: búsqueda local

C. A. Estrada

3 de noviembre de 2020

1. Objetivo

Maximizar alguna variante de la función bidimensional $g(x, y)$ [1], con restricciones $-3 \leq x, y \leq 3$, en donde la posición actual es un par x, y en donde se requieren dos movimientos aleatorios en Δx y Δy , cuya combinación genera ocho posibles vecinos, de los cuales se selecciona el que obtiene el mayor valor para g , y crear, posteriormente, una animación de las réplicas simultáneas de la búsqueda local.

2. Metodología

Para la generación del código se emplea el paquete estadístico R versión 4.0.2 [2], y se emplea el código para la búsqueda local reportado previamente [1][3]. La función original $g(x, y)$ proporcionada se muestra a continuación:

```
g <- function(x, y) {  
  return (((x+0.5)^4-30*x^2-20*x+(y+0.5)^4-30*y^2-20*y)/100)  
}
```

Por lo tanto, se realiza una modificación al elevar el denominador al cuadrado:

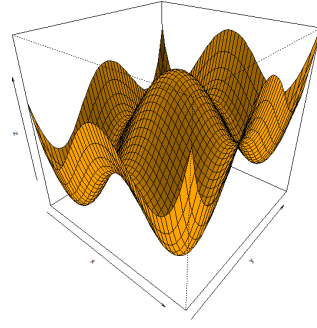
```
g <- function(x, y) {  
  return (((x+0.5)^4-30*x^2-20*x+(y+0.5)^4-30*y^2-20*y)^2/100)  
}
```

La función modificada se utiliza para la búsqueda local, considerando las restricciones $-3 \leq x, y \leq 3$ y generando movimientos al azar en Δx y en Δy , en pasos de 0.25, utilizando quince réplicas de búsqueda simultáneas en cien y en mil pasos. Finalmente, para producir la animación se utiliza el servicio en línea gratuito de Ezgif [4].

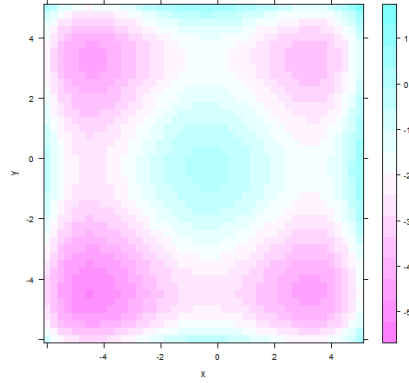
3. Resultados y discusión

En la figura 1 se observa la función $g(x, y)$ original y la versión modificada en tres y dos dimensiones. Se observa claramente que en la versión modificada se producen cuatro máximos de distinto valor en el eje z .

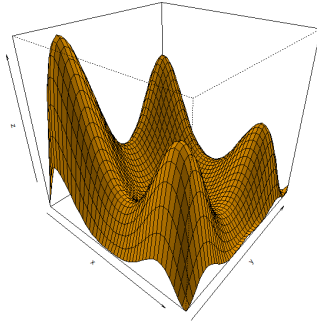
En la figura 2 se presentan los pasos 1, 19 y 98 de la búsqueda realizada en cien pasos, observando que las réplicas localizan los cuatro máximos de la función al finalizar



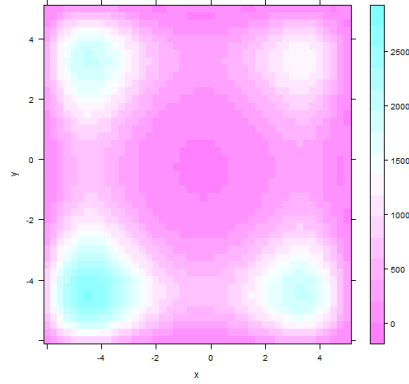
(a) Función original 3D.



(b) Función original 2D.



(c) Función modificada 3D.

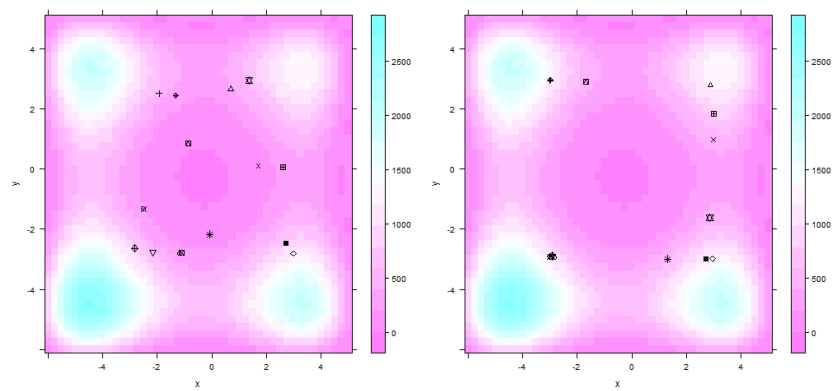


(d) Función modificada 2D.

Figura 1: Visualización de la función bidimensional original y modificada en tres (3D) y dos dimensiones (2D).

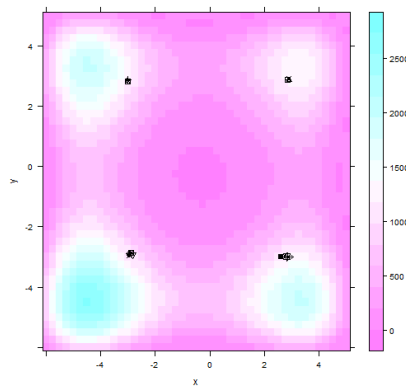
la simulación, aunque la cantidad de las réplicas en cada máximo no presenta un patrón respecto al valor de z de cada uno de ellos, y tampoco se localizan en el centro exacto. En el experimento con mil pasos, presentado en la figura 3, se observa una conducta similar, por lo que la distribución aleatoria inicial de las réplicas y la distancia entre cada uno de los máximos influye principalmente en la búsqueda, puesto que la réplica no localiza ningún valor mejor dentro del rango de vecinos, en comparación con el valor de los máximos en z y de la cantidad de pasos del experimento, para este caso en particular.

Las animaciones generadas para los experimentos en cien y en mil pasos se localizan en el repositorio de GitHub en <https://github.com/CrisAE/Simulacion/blob/master/P7/Cien-anim.gif> y en <https://github.com/CrisAE/Simulacion/blob/master/P7/Mil-anim.gif>, en donde se observa el comportamiento de las quince réplicas durante la búsqueda de los cuatro máximos de la función.



(a) Paso 1.

(b) Paso 19.



(c) Paso 98.

Figura 2: Búsqueda local de las 15 réplicas simultáneas en cien pasos.

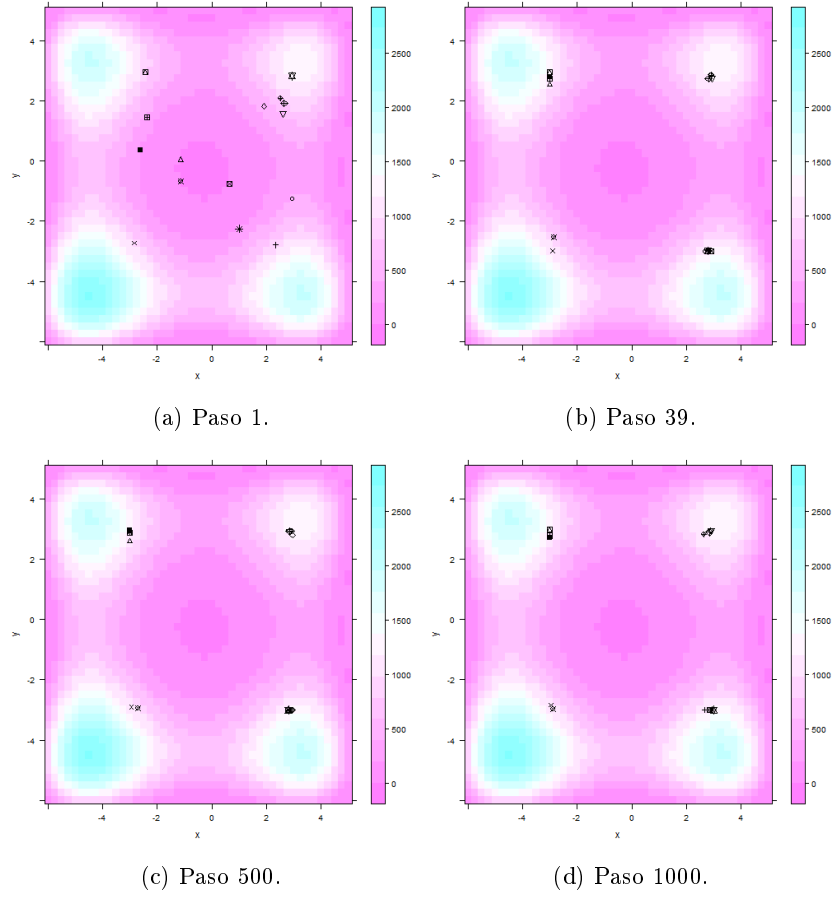


Figura 3: Búsqueda local de las 15 réplicas simultáneas en mil pasos.

4. Conclusión

Las quince réplicas localizan los cuatro máximos de la función modificada conforme aumentan los pasos, independientemente del valor en el eje z que posean y de la cantidad de pasos del experimento, para este caso en particular.

5. Reto 1

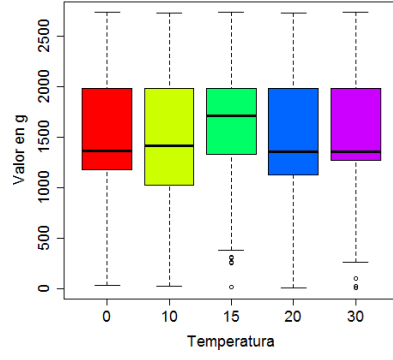
Cambiar la regla de movimiento de una solución \mathbf{x} a la siguiente a la de recocido simulado [1]: para optimizar la función $f(x)$, se genera una solución actual \mathbf{x} un solo vecino $\mathbf{x}' = \mathbf{x} + \Delta\mathbf{x}$. Se calcula $\delta = f(\mathbf{x}') - f(\mathbf{x})$ (para minimizar, al revés para maximizar). Si $\delta > 0$, siempre se acepta al vecino \mathbf{x}' en la solución actual como mejora, y si $\delta < 0$, se acepta a \mathbf{x}' con probabilidad $\exp(-\delta/T)$, en donde T es una temperatura en donde la reducción se logra multiplicando el valor actual de T con $\xi < 1$. Examina los efectos estadísticos del valor inicial de T y el valor de ξ en la calidad de la solución.

Para ello, se emplea la función $g(x, y)$ modificada, y se utiliza el mismo código generado para la búsqueda local [3][5] modificándolo para que incluya el movimiento del valor de recocido simulado. Los valores iniciales de T son de 0, 10, 15, 20 y 30, mientras que el valor inicial de ξ es de 0.20, 0.40, 0.70 y 0.99, para cien réplicas en cien pasos.

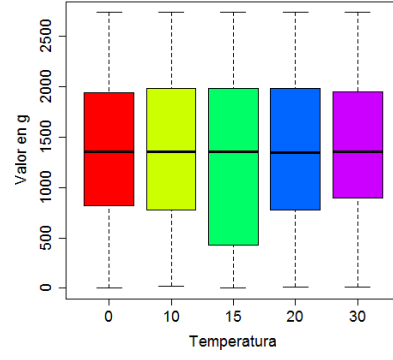
En la figura 4 se presentan los valores en g obtenidos, observando que no se presenta ningún patrón en el valor respecto a la temperatura, mientras que para el valor de ξ tampoco hay un patrón en especial, con la excepción de ξ a 0.99, en donde las medianas de los datos presentan un valor en g más bajo.

Referencias

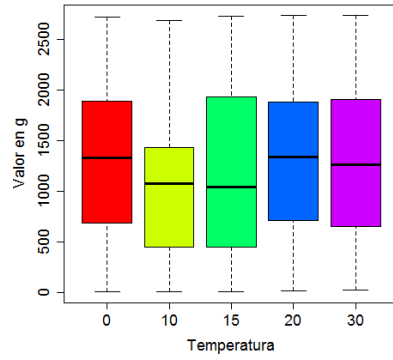
- [1] E. Schaeffer. Práctica 7: búsqueda local, 2020. URL <https://elisa.dyndns-web.com/teaching/comp/par/p7.html>.
- [2] The R Foundation. The R Project for Statistical Computing, 2020. URL <https://www.r-project.org/>.
- [3] C. Téllez. Código de práctica 7, 2020. URL <https://github.com/claratepa/Simulacion/blob/master/Practica7/Pr7sim.R>.
- [4] Ezgif. Animated GIF Maker, 2020. URL <https://ezgif.com/maker>.
- [5] R. Parga. Práctica 7: código de reto 2, 2018. URL <https://github.com/RParga/R-simulation/blob/master/p7/r2.R>.



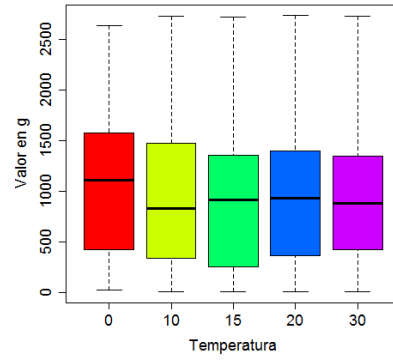
(a) ξ a 0.20.



(b) ξ a 0.40.



(c) ξ a 0.70.



(d) ξ a 0.99.

Figura 4: Valores de g en las diferentes temperaturas utilizadas a distintos valores de ξ .