

# P7

Ismael Crespo

30 de marzo de 2022

## 1. Introducción

Durante esta practica se pretende buscar un valor óptimo, el mayor evaluado dentro de una función  $g(x, y)$ , por medio de una búsqueda local dentro de la matriz creada, seleccionando la mejor opción en cada iteración. Se utiliza como base el trabajo presentado E. Schaeffer [2] donde se realiza una búsqueda dentro de una función  $f(x)$ .

## 2. Objetivos

1.-Crea una visualización (animada) de cómo proceden por lo menos 5 réplicas simultáneas de por lo menos 500 pasos de la búsqueda encima de una gráfica de proyección plana.

2.-Simular el recocido simulado para optimizar una función  $f(x)$ , se genera para la solución actual  $x$  un sólo vecino  $x' = x + \Delta x$  (algún desplazamiento local). Se calcula  $\delta = f(x) - f(x')$ . Si  $\delta > 0$ , siempre se acepta al vecino  $x'$  como la solución actual ya que representa una mejora. Si  $\delta < 0$ , se acepta a con probabilidad  $\exp(\delta/T)$  y rechaza en otro caso. Aquí  $T$  es una temperatura que decrece en aquellos pasos donde se acepta una empeora; la reducción se logra multiplicando el valor actual de  $T$  con  $\varepsilon < 1$ , como por ejemplo 0,955. Examina los efectos estadísticos del valor inicial de  $T$  y el valor de  $\varepsilon$  en la calidad de la solución, es decir, qué tan alto el mejor valor termina siendo.

## 3. Programación en R

La búsqueda comienza seleccionando un punto en la matriz, coordenada en  $y$  y coordenada en  $x$ , este ejercicio se replica 5 veces para tener 5 puntos en donde para el primer objetivo se mueve aleatoriamente entre sus vecinos. El código 1 muestra como se generan aleatoriamente y posteriormente como es la dinámica de movimiento aleatorio de cada replica hacia sus vecinos, la mejor opción encontrada (**best**) es seleccionada utilizando un ciclo **if** con la condición de que este valor cambiará a uno nuevo cada que una replica actual sea mayor al mejor valor obtenido hasta este tiempo.

Código 1: Generación de cada replica y movimiento aleatorio hacia los vecinos

```
1  #Generación de cada replica
2  for(i in 1:replicas){
3    curr_x[i] <- sample(seq(low, high, step),1)
4    curr_y[i] <- sample(seq(low, high, step),1)
5    curr[i] <- z[as.character(curr_y[i]), as.character(curr_x[i])]
6    best_y[i] <- curr_y[i]
7    best_x[i] = curr_x[i]
8    bests[i] = curr[i]
9  }
10 #Movimiento hacia un vecino
11 for(i in 1:replica){
```

```

12 min_fila<-max(curr_y[i]- delta, low)
13 max_fila <-min(curr_y[i]+ delta,high)
14 min_col<-max(curr_x[i] - delta, low)
15 max_col <-min(curr_x[i] + delta,high)
16 filas<-(seq(min_fila,max_fila,delta))
17 columnas<-(seq(min_col,max_col,delta))
18 #vecindad<-z[as.character(filas),as.character(columnas)]
19 curr_y[i]<- sample(filas,1)
20 curr_x[i] <- sample(columnas,1)

```

En la segunda parte del trabajo el movimiento no es completamente aleatorio, tal como se describe en el objetivo 2, cada replica se va a mover siempre que la posición vecina, seleccionada aleatoriamente, evaluada en la función sea mejor, cuando esta posición no represente una mejora en la búsqueda se va a mover con una probabilidad dependiente de la temperatura. (véase el código 2). Para consulta detallada de las rutinas desarrolladas véase la referencia [1]

Código 2: Selección de un vecino con una probabilidad dependiente de la temperatura si el valor del vecino seleccionado no es mejor.

```

1  for(f in 1:length(filas)){
2      prob_y=sample(filas,1)
3      prob_x=sample(columnas,1)
4      prob <-z[format(round(prob_y, 2),
5      nsmall = 2),format(round(prob_x, 2), nsmall = 2)]
6      resta=curr[i]-prob
7      if(resta<0){
8          curr_y[i]<- prob_y
9          curr_x[i] <- prob_x
10         curr[i] <-z[format(round(curr_y[i], 2),
11         nsmall = 2),format(round(curr_x[i], 2), nsmall = 2)]
12         break
13     }
14     if(resta>0){
15         expo=exp(-abs((resta)/temperatura))
16         pick=runif(1)
17         if(pick<expo){
18             curr_y[i]<- prob_y
19             curr_x[i] <- prob_x
20             curr[i] <-z[format(round(curr_y[i], 2),
21             nsmall = 2),format(round(curr_x[i], 2), nsmall = 2)]
22             temperatura=temperatura*e
23             break
24         }}}

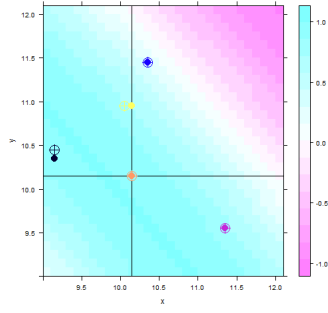
```

Para

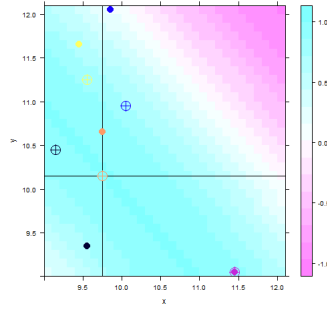
## 4. Resultados

Las secuencias gráficas de la búsqueda aleatoria del mejor resultado se presenta en la figura 1, y esta disponible de manera extensa como material complementario en el repositorio de la practica [1], se observa que la calidad búsqueda aleatoria depende de donde aparece por primera vez la replica, ya que la función en los límites evaluadas de 7 a 12 tanto en  $x$  y  $y$  acumula las mejores opciones en una sola zona.

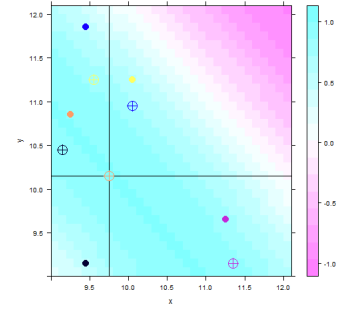
Para el análisis entre la búsqueda aleatoria y la búsqueda probabilística las rutinas descrita en la sección 3 se



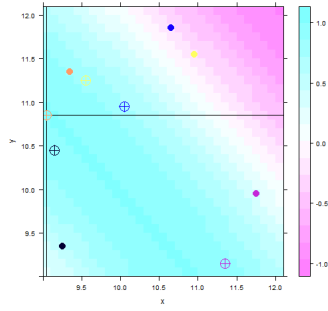
(a) Tiempo=2.



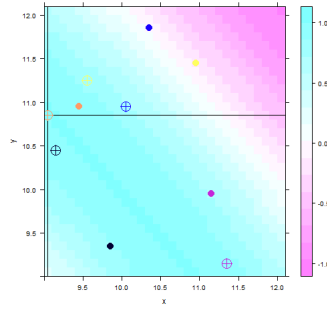
(b) Tiempo=50.



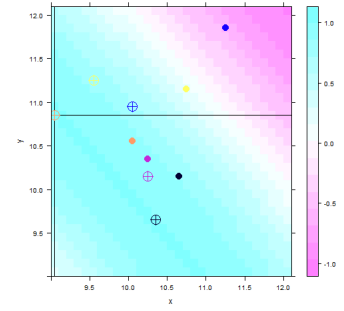
(c) Tiempo=100.



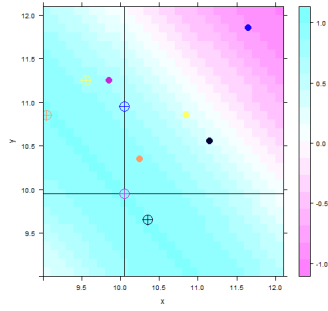
(d) Tiempo=150.



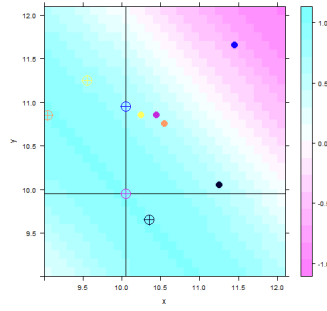
(e) Tiempo=200.



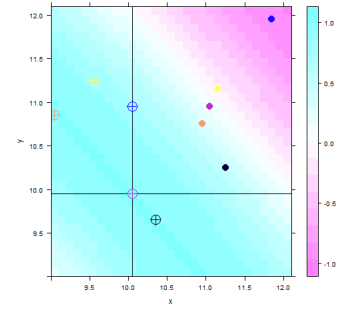
(f) Tiempo=250.



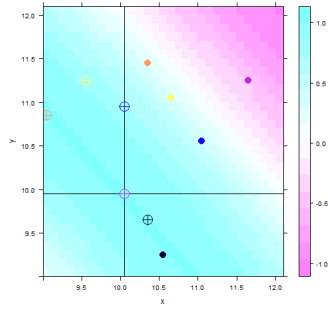
(g) Tiempo=300.



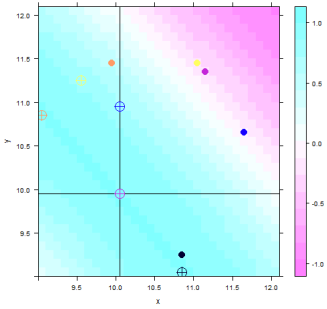
(h) Tiempo=350.



(i) Tiempo=400.



(j) Tiempo=450.



(k) Tiempo=500.

Figura 1: Búsqueda aleatoria del máximo en el plano, el círculo con la cruz dentro representa la mejor opción para cada replica en dado tiempo, el círculo representa la búsqueda actual y el punto de cruce entre la línea vertical y horizontal, representa la mejor opción de todas las replicas.

evaluaron 15 veces y se analizaron estadísticamente, se considera el tiempo en el que se llega al resultado mayor en cada una de las búsqueda (cuadro 1), igualmente se analiza la sumatoria de las 5 replicas, interpretando que a sumatorias mayores, las replicas en conjunto se acercaron más al resultado óptimo (cuadro 2). Para la rutina aleatoria durante 500 pasos no siempre se obtuvo el resultado óptimo y se excluyeron del análisis. Es claro que los tiempos en los que se obtiene el valor óptimo mejora al utilizar la segunda rutina y las replicas en conjunto se acercan a valores mas cercanos al mejor. El cuadro 3 presenta los valores  $p$  para ambos análisis, ambas relaciones tienen valores menores a 0,05 por lo que fueron aceptadas como significativas.

Cuadro 1: Datos estadísticos de los tiempos en encontrar el valor máximo de las replicas. El Tiempo 1 hace referencia a las replicas aleatorias y el Tiempo 2 a la simulación de recocido térmico.

Estadística	N	Mediana	Min	Max
Tiempo 1	12	117.58330	22	230
Tiempo 2	12	19.91667	4	87

Cuadro 2: Datos estadísticos de las sumatorias de las replicas. La Sumatoria 1 hace referencia a las replicas aleatorias y la Sumatoria 2 a la simulación de recocido térmico.

Estadística	N	Mediana	Min	Max
Sumatoria 1	15	4.76212	3.82853	4.99983
Sumatoria 2	15	4.99990	4.99985	4.99993

Cuadro 3: Valores  $p$  entre las sumatorias y los tiempos de las replicas para encontrar el valor máximo.

Variable 1	Variable 2	$p - value$
Sumatoria 1	Sumatoria 2	$7,94 \times 10^{-7}$
Tiempo 1	Tiempo 2	$1,553 \times 10^{-4}$

## 5. Conclusiones

La búsqueda aleatoria es muy dependiente de la posición inicial de las replicas, la forma y los limites en los que se evalúa la función determinan el plano donde se realizará la búsqueda. La búsqueda que se rige por la mejor opción entre los vecinos y probabilístico (dependiente de la temperatura) si la selección es peor mejora considerablemente las rutinas.

## Referencias

- [1] I. Crespo. P7: Búsqueda local. 2022. URL <https://github.com/IsmaelHC/Simulacion-NANO-2022/tree/main/P7>.
- [2] E. Schaeffer. Simulación. 2022. URL <https://satuelisa.github.io/simulation>.