Tarea 7: búsqueda local

Simulación de Sistemas

26 de septiembre de 2017

1. Introducción

Una búsqueda local es un procedimento iterativo que utiliza una solución inicial y va tratando de mejorarla en cada paso con movimientos locales. En otras palabras, busca dentro de su vecindario una solución mejor, y si dicha solución es encontrada, reemplaza la solución actual por la nueva solución, así continua el proceso hasta que no encuentra una mejor solución.

2. Especificaciones computacionales

La presente tarea se realizó en una máquina con las siguientes especificaciones: procesador Intel(R)Core(TM) i5-6200U CPU 2.30 GHz 2.40 GHz con 8GB en memoria RAM y sistema operativo Windows 10 Home. Se emplearon tres de los cuatro núcleos.

3. Tarea

En esta tarea se implementa una búsqueda local para encontrar los máximos locales de la función $g(x, y)$, donde

$$g(x,y) = \frac{(x+0{,}5)^4 - 30x^2 - 20x + (y+0{,}5)^4 - 30y^2 - 20y}{100}. \tag{1}$$

Dicha función la podemos ver dibujada en tres dimenciones en la figura 1.

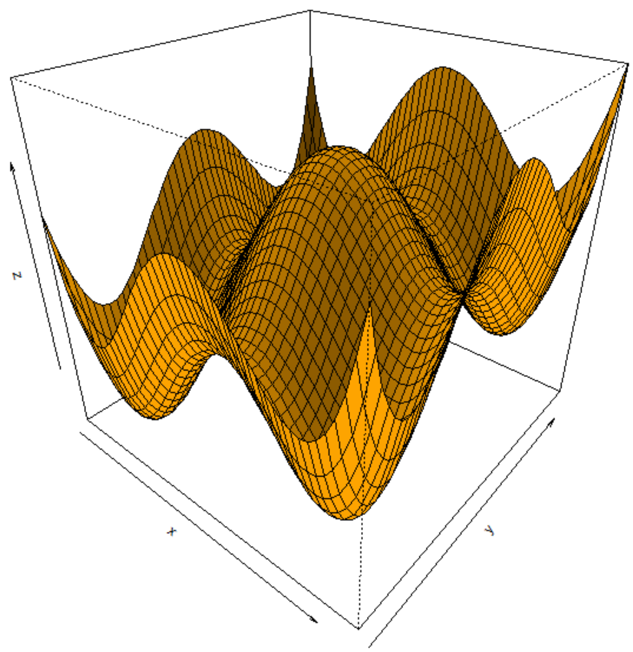

Figura 1: Función $g(x, y)$.

El procedimiento de la búsqueda local es simple, se iniciará con un punto (x, y) obtenido de manera aleatoria. Luego se elige al azar un Δ positivo y a partir de ahí se generan los vecinos $(x + \Delta, y), (x - \Delta, y), (x, y + \Delta), (x, y - \Delta)$. Se evalúan en la función los cuatro vecinos y se selecciona como el próximo valor de (x, y) aquel que haya tenido mejor evalución. Esto se reproducirá $t_{\text{máx}}$ veces y aquel punto (x, y) que haya otorgado el mayor valor para $g(x, y)$ se da como resultado. Se ejecutan cien réplicas, y el resultado de la búsqueda es el mayor obtenido de esas cien réplicas.

La función `replica` dada en el código tuvo las siguientes modificaciones. En lugar de hacer las comparaciones entre cada par de vecinos para obtener el mejor de ellos, lo que se hace ahora es agrupar en un solo vector los vecinos para ser evaluados por $g(x, y)$ y de ellos se obtiene el máximo. Además como queremos visualizar los puntos en una gráfica, lo que se tiene que considerar es que dichos puntos no se salgan de los límites de la imagen. Para que esto no suceda se concideró que nos encontramos en un toro, así si alguno de los vecinos en alguna coordenada superaba el límite establecido, su valor en esa coordenada será el valor del límite violado ± 10 según sea el caso, ya que nuestros límites son -5 y 5.

A continuación se muestra la parte del código donde se encuentra la función `replica`.

```
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)
    x2 <- curr + c(delta,0)
    y1 <- curr + c(0,-delta)
    y2 <- curr + c(0,delta)
    puntos <- c(x1,x2,y1,y2)
    for(k in 1:8){
      if(puntos[k] < (-5)){
        puntos[k] <- puntos[k]+10)
      }
      if(puntos[k] > 5){
        puntos[k] <- puntos[k]-10
      }
    }
    vecx <- c()
    vecy <- c()
    for(p in 1:8){
      if(p %%2 == 0){
        vecy <- c(vecy,puntos[p])
      }else{
        vecx <- c(vecx,puntos[p])
      }
    }
    valg <- c()
    for(q in 1:4){
      valg <- c(valg, g(vecx[q], vecy[q]) )
    }
    dm <- which.max(valg)
    curr <- c(vecx[dm], vecy[dm])
    if(g(curr[1],curr[2]) > g(best[1],best[2])){
      best <- curr
    }
  }
  return(best)
}
```

En las figuras 2, 3 y 4 se muestra los puntos obtenidos de las cien réplicas con $0 < \Delta \leq 0.01$ para cien, mil y diez mil pasos. El punto rojo mostrado en las figuras es el punto que tuvo el valor máximo en g de las todas las réplicas.

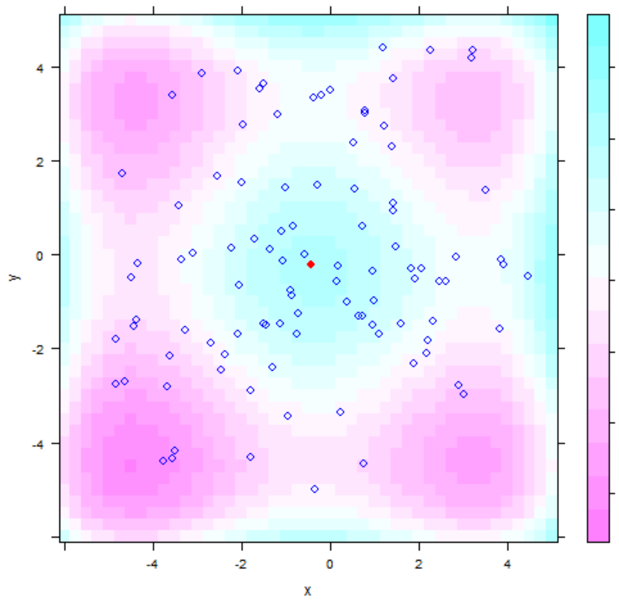

Figura 2: Puntos obtenidos de las cien réplicas con cien pasos.

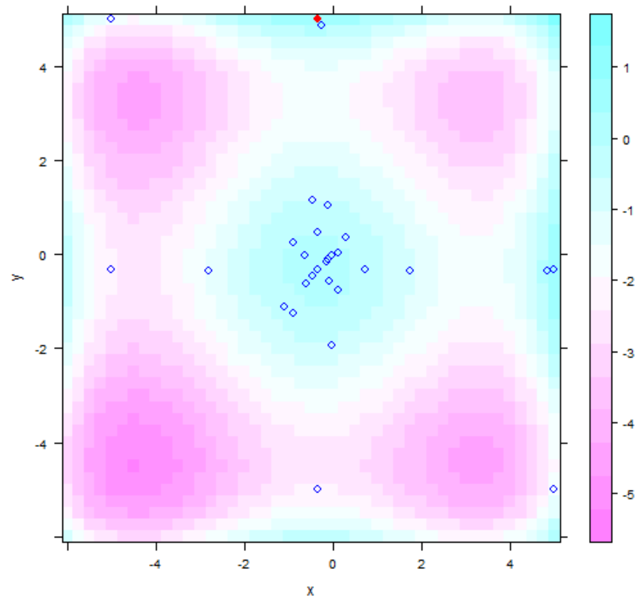

Figura 3: Puntos obtenidos de las cien réplicas con mil pasos.

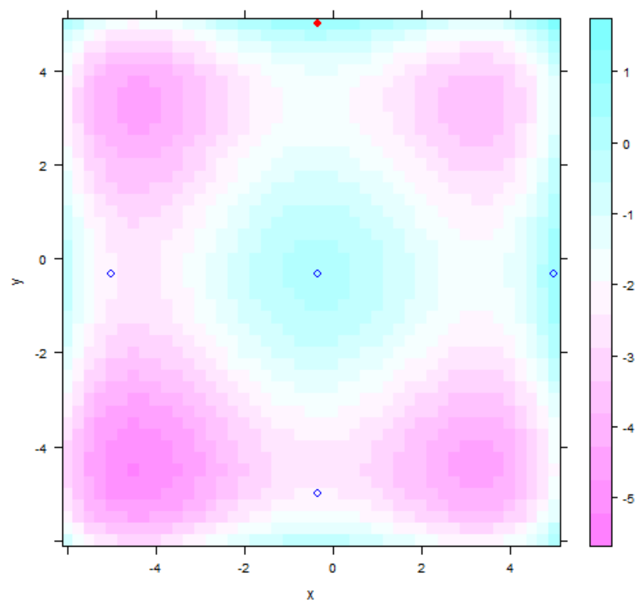

Figura 4: Puntos obtenidos de las cien réplicas con diez mil pasos.

Notemos que entre más cantidad de pasos se realicen el valor entregado por la búsqueda local mejora y los puntos se empiezan a concentar en distintas zonas, por ejemplo en la figura 2 los puntos se encuentran todavia muy dispersos pero cuando aumentamos la cantidad de pasos a mil como se muestra en la figura 3 se observa que los puntos se agrupan mayormente en el centro, ahora bien en la figura 4 donde se realizan diez mil pasos los puntos se encuentran en los extremos de la imagen.

4. Reto 1

Para este reto se desea tener una visualización de como se va encontrando el mejor valor de entre las réplicas, para ello se realizó un gif el cual puede ser encontrado dentro de los archivos de la carpeta con el nombre *reto1.gif*.

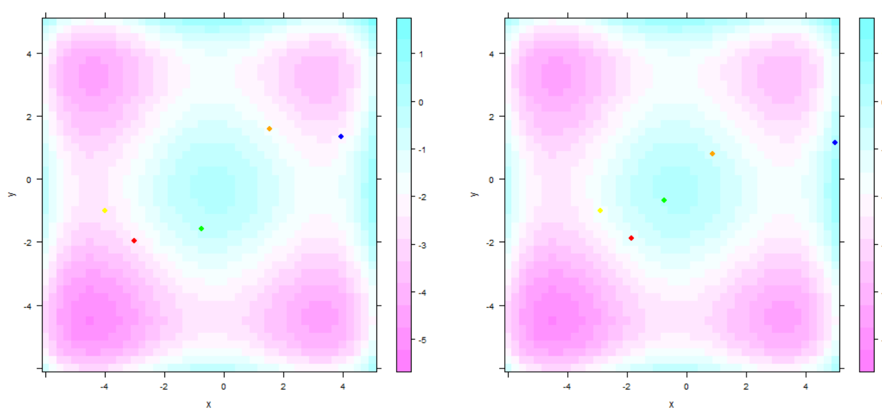

Figura 5: La primera imagen muestra la posición de los cinco puntos cuando inicia la búsqueda local; mientras que la segunda imagen se ve como cambió la posición al cabo de veinticinco pasos de la búsqueda.

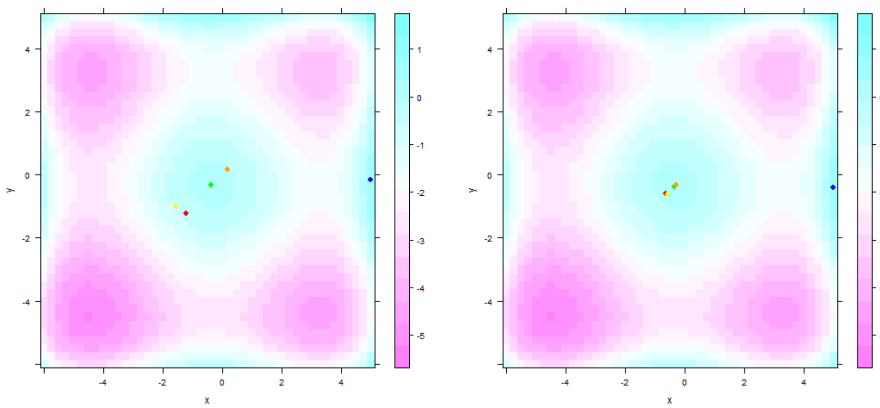

Figura 6: La primera imagen muestra la posición de los cinco puntos al finalizar cincuenta pasos de la búsqueda local y en la segunda imagen se ve como cambió la posición en setenta y cinco pasos de la búsqueda.

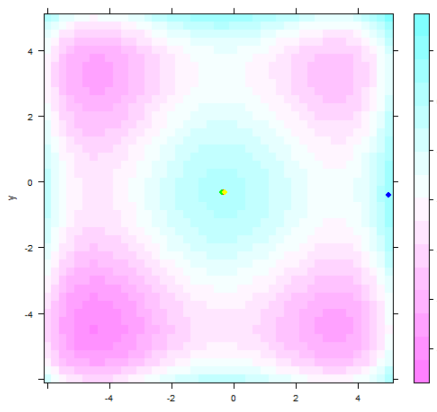

Figura 7: Posición de los puntos al finalizar la búsqueda en 100 pasos.

En la figuras 5 se muestra cinco puntos distintintos los cuales empiezan a realizar pasos en su vecindario con la idea de mejorar su valor en g. En la figura 5 se encuentra como se ven localizados los puntos para 1 y 25 pasos, la figura 6 muestra los puntos en 50 y 75 pasos y la figura 7 muestra la posición de los puntos al finalizar la búsqueda local en cien pasos. Podemos notar que mientras va aumentando la cantidad de pasos el valor en g para cada punto también mejora y como se van empezado a dirigir cuatro de los cinco puntos hacía el centro.

5. Reto 2

Para el segundo reto se nos pide implementar un recocido simulado y examinar los efectos del valor inicial de la temperatura en calidad de la solución. A grandes razgos lo que se hace en un recocido simulado es lo siguiente. Partiendo de una solución inicial (x, y) se obtiene un vecino sumando un Δ a dicha solución. Luego si resulta que la nueva solución toma un mejor valor en g se elige dicha solución como el proximo punto(x, y). Si la nueva solución no es mejor que la que ya teniamos se va aceptar reemplar (x, y) por la nueva solución si se satisface la probabilidad $e^{(\frac{\delta}{T})}$. Donde δ es la diferencia entre la nueva solución y (x, y) ambas evaluadas en g, y T es una temperatura que va decreciendo.

Para hacer el recocido simulado se toma el código que tenemos para la tarea, solo se modifica al momento de definir el movimiento. A continuación se muestra la función `replica` que fue la única parte modificada.

```
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)
    x2 <- curr + c(delta,0)
    y1 <- curr + c(0,-delta)
    y2 <- curr + c(0,delta)
    puntos <- c(x1,x2,y1,y2)
    for(k in 1:8){
      if(puntos[k] < (-5)){
        puntos[k] <- puntos[k]+10
      }
      if(puntos[k] > 5){
        puntos[k] <- puntos[k]-10
      }
    }
    vecx <- c()
    vecy <- c()
    for(p in 1:8){
      if(p %%2 == 0){
        vecy <- c(vecy,puntos[p])} else{
        vecx <- c(vecx,puntos[p])}
    }
    u <- sample(1:4,1)
    x.p <- c(vecx[u],vecy[u])
    delt <- g(x.p[1],x.p[2]) - g(curr[1],curr[2])
```

```
    if(delt > 0){
      curr <- x.p
    }else{
      if(runif(1)< exp((delt) / (t * ep))){
        curr <- x.p
        if(t == 1){
          t <-t
        }else{
        t <- t-1
        }
      }
    }

    if(g(curr[1],curr[2]) > g(best[1],best[2])){
      best <- curr
    }
  }
  return(best)
}
```

En las figuras 8, 9 y 10 se muestra los puntos obtenidos de las cien réplicas con $0 < \Delta \leq 0.01$, valor inicial de $T = 12$ y $\xi = 0.4$ para cien, mil y diez mil pasos. El punto rojo mostrado en las figuras es el punto que tuvo el valor máximo en g de las todas las réplicas.

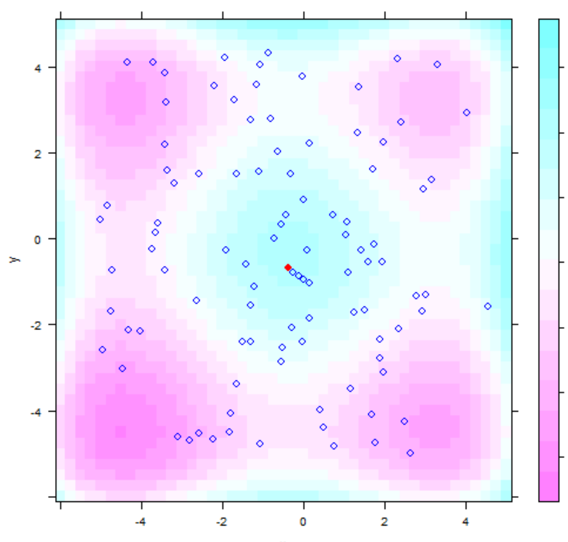

Figura 8: Puntos obtenidos de las cien réplicas con cien pasos.

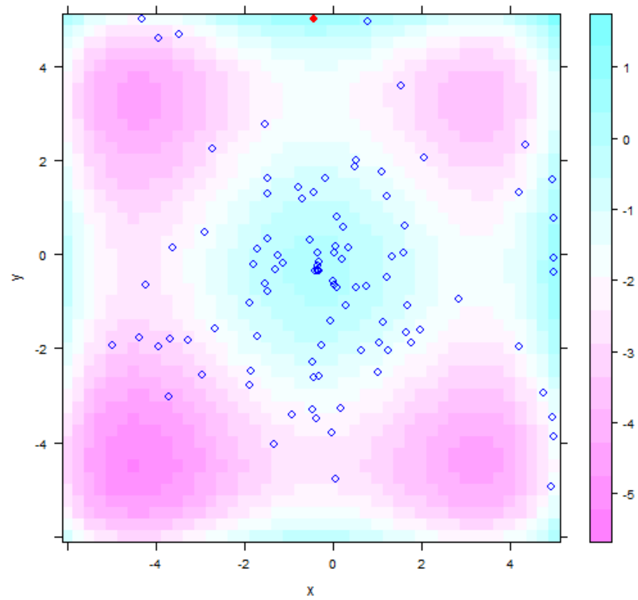

Figura 9: Puntos obtenidos de las cien réplicas con mil pasos.

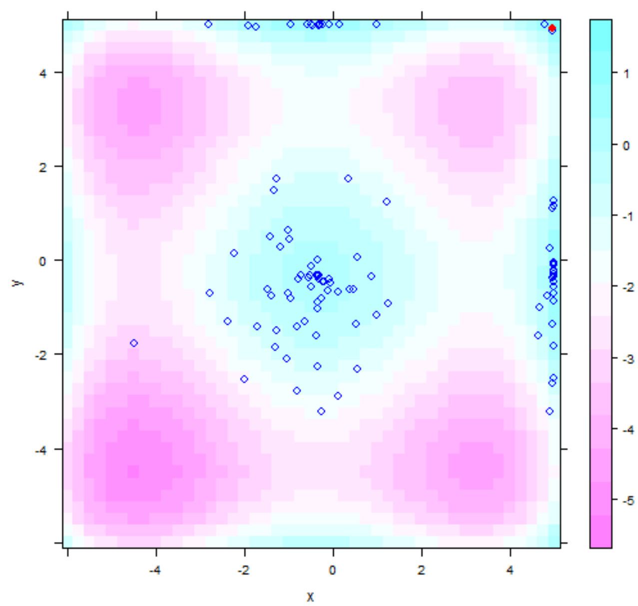

Figura 10: Puntos obtenidos de las cien réplicas con diez mil pasos.

Para comparar la calidad de la solución con distinatos valores iniciales de temperatura T y distintos ξ se consideró ejecutar cien replicas para el recocido simulado con cien pasos, donde los valores de temperatura fueron: $5, 10, 15, 20$ y 40. Los distintos valores de ξ que se consideraron son: 0.01, 0,1, 0.3 y 0.6. En las figuras 11 y 12 se muestran los resultados.

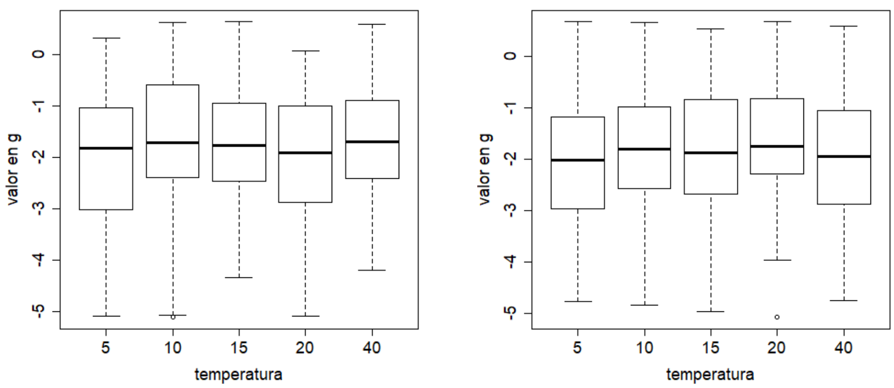

Figura 11: En la primera imagen se dan lo valores en g para un $\xi = 0.01$; En la segunda imagen $\xi = 0.1$.

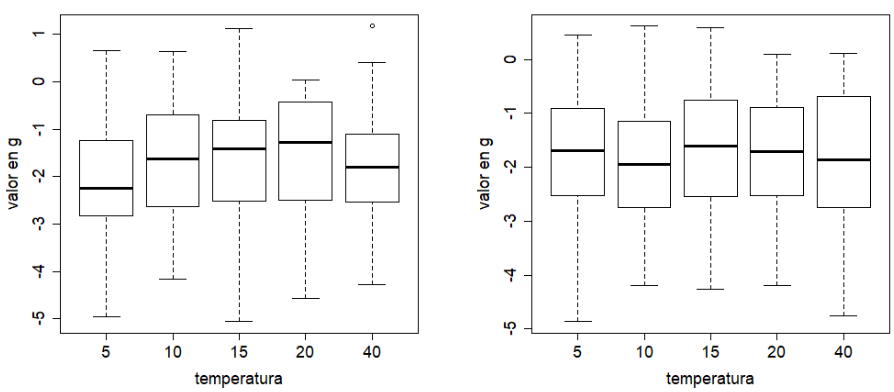

Figura 12: En la primera imagen se dan lo valores en g para un $\xi = 0.3$; En la segunda imagen $\xi = 0.6$.

Por los resultados obtenidos en las figuras 11 y 12 podemos argumentar que la variación de ξ para las distintas temperaturas no afecto en la calidad de la solución obtenida, ya que podemos notar que se presentan resultados similares cuando se varia ξ y la temperatura. Se esperaría observar un cambio más relevante al variar estos parámetros, pero dicho cambio no se dío, posiblemente la manera en que definí la disminución de la temperatura cada vez que se acepta una empeora está mal definida y es por esto que no se vieron los resultados esperados.