Optimización y metaheurísticas I

Unidad 2: Métodos de búsqueda y optimización

Dr. Jonás Velasco Álvarez

jvelascoa@up.edu.mx

Búsqueda Aleatoria

Búsqueda Aleatoria Simple

Ventajas:

- Es el algoritmo estocástico más simple.
- Fácil de programar.
- No requiere de un ajuste de parámetros.
- El algoritmo se puede aplicar en cualquier función.

Desventajas:

- Puede requerir de muchas iteraciones para alcanzar una solución óptima, especialmente cuando el espacio de búsqueda es grande.
- Puede tener una limitada utilidad en la práctica.

Búsqueda Aleatoria Simple

Paso 0 **(Inicialización)** Elegir un vector inicial $\mathbf{x}_0$ de manera aleatoria o determinista. Calcular $f(\mathbf{x}_0)$. Definir $k \leftarrow 0$.

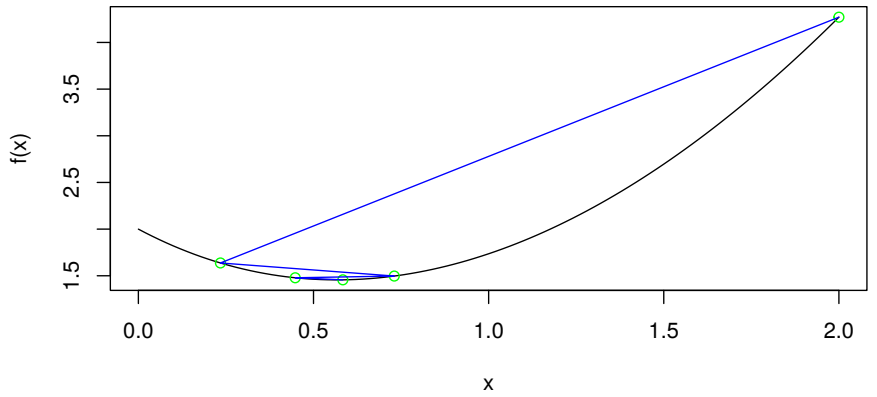
Paso 1 Generar un nuevo vector independiente $\hat{\mathbf{x}}$, de acuerdo a una distribución de probabilidad. Si $f(\hat{\mathbf{x}}) < f(\mathbf{x}_k)$, definir $\mathbf{x}_{k+1} \leftarrow \hat{\mathbf{x}}$. En caso contrario, $\mathbf{x}_{k+1} \leftarrow \mathbf{x}_k$.

Paso 2 Detener el algoritmo hasta alcanzar K iteraciones. En caso contrario, ir al **Paso 1**. Definir $k \leftarrow k + 1$.

Búsqueda Aleatoria Simple

```r
# Funcion a optimizar
f1 <- function(x){
  x^2+ 2*exp(-x)
}

# Funcion del optimizador
BusqAleatoria <- function(iteraciones){

  # INICIALIZACION x = 2
  mejorX <- 2
  mejorFx <- f1(mejorX)

  for(i in 1:iteraciones){
    # Generar un nuevo x entre 0 y 2
    nuevoX <- runif(1,0,2)

    # reemplazar mejorX por nuevoX si este ultimo es mejor
    if(f1(nuevoX) < f1(mejorX)){
      mejorX <- nuevoX
      mejorFx <- f1(nuevoX)
    }
  }

  # imprimir resultado
  cadena <- cat('valor de x = ', mejorX,' con fx = ', mejorFx,'\n')
  return(cadena)
}

# ejecutar el programa
set.seed(19) # hacer reproducible el ejercicio
BusqAleatoria(20)
```


$$\text{mín } f_1(x) = x^2 + 2e^{-x}$$




```
##
## valor de x =  0.5837632  con fx =  1.45637
```


Resultados de cada iteración.

```
##           x                  f(x)               Estado
## historial "2"                "4.27067056647323" "Aceptado"
##           "0.23426116630435" "1.6371886257201"  "Aceptado"
##           "0.968059466220438" "1.69677788165315" "No"
##           "1.30241405358538" "2.24003172728446" "No"
##           "0.136793352663517" "1.76301329148501" "No"
##           "0.730595606379211" "1.49701403502543" "Aceptado"
##           "0.447807408403605" "1.47858696213054" "Aceptado"
##           "0.583763214293867" "1.45637010764265" "Aceptado"
##           "1.14656746480614" "1.9500679588002"  "No"
##           "1.67324066068977" "3.17501032494648" "No"
##           "1.45210416382179" "2.57676096798158" "No"
##           "0.812931420747191" "1.54796931398813" "No"
##           "0.899435093160719" "1.62258228402193" "No"
##           "1.58191019529477" "2.91360390815354" "No"
##           "1.848307013046"   "3.73124599897314" "No"
##           "0.456405003555119" "1.47541991170174" "No"
##           "1.80990367475897" "3.6030911150783"  "No"
##           "0.795028073713183" "1.53520675306905" "No"
##           "1.13535565696657" "1.93164811909391" "No"
##           "0.854732232168317" "1.5813613556832"  "No"
##           "1.67648362182081" "3.18465831627926" "No"
```


Actividad

1. Resolver el siguiente problema de optimización (con $D = 2$):

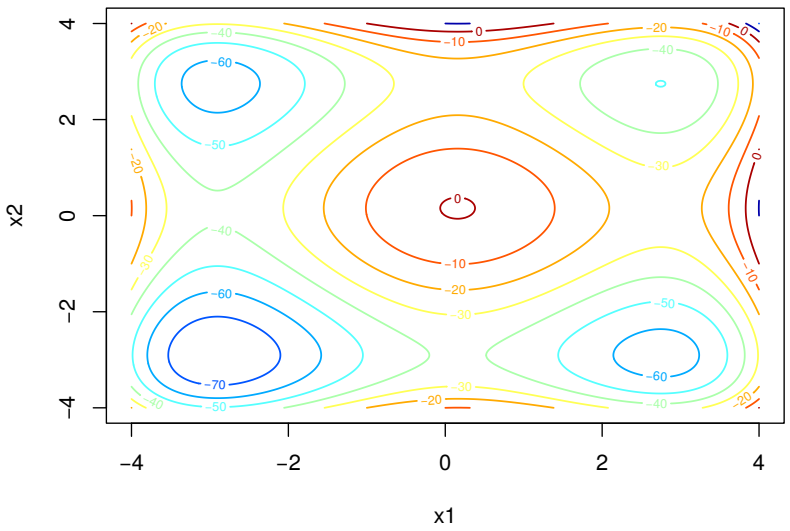
$$\text{min } f(x_1, x_2) = \frac{1}{D}\sum_{i=1}^{D}(x_i^4 - 16x_i^2 + 5x_i)$$

donde $x_1, x_2 \in [-8, 8]$.

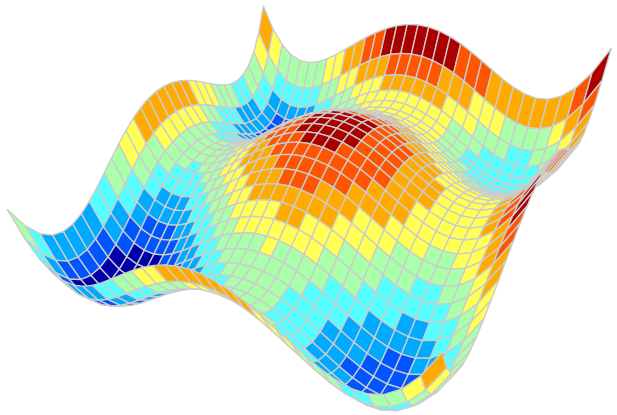
La solución óptima es: $\mathbf{x}^* = (-2.9035, -2.9035)$ y $f(\mathbf{x}^*) = -78.33$.

2. La búsqueda aleatoria simple deberá inicializar en $\mathbf{x}_0 = (4.0, 6.4)$.
3. Considere la distribución uniforme para generar nuevas soluciones, $\mathcal{U}(-8, 8)$.

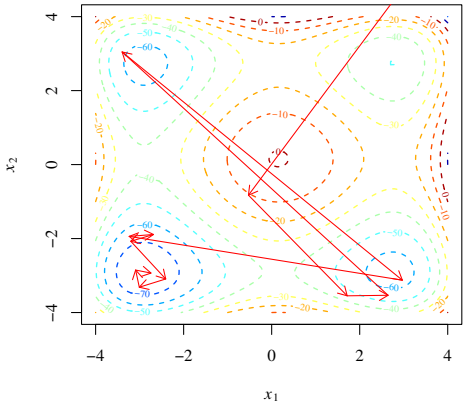
$\frac{1}{2}((x_1^4 - 16x_1^2 + 5x_1) + (x_2^4 - 16x_2^2 + 5x_2))$



$\frac{1}{2}((x_1^4 - 16x_1^2 + 5x_1) + (x_2^4 - 16x_2^2 + 5x_2))$



Resultados




```
iter   x1         x2         f(x)
 0  4.0000000  6.4000000 537.18080
 2 -0.5276512 -0.8271382 -10.81477
17  1.7155351 -3.5474330 -45.28561
53  2.6448358 -3.5334269 -55.65854
55 -3.3975480  3.0449880 -57.79487
82  2.9688612 -3.1249831 -62.50037
132 -3.2379002 -1.9413034 -64.91076
231 -2.6851474 -1.8894810 -65.31270
254 -3.2009208 -2.0729422 -67.80686
256 -2.4072120 -3.0927669 -74.09327
320 -3.0076445 -3.3151985 -74.78846
892 -3.0906568 -2.8516098 -77.64240
7307 -2.7422467 -2.9327247 -77.89166
```


Resultados

Gráfico de convergencia.