

Centro de Investigación en Cómputo
Instituto Politécnico Nacional
Metaheurísticas
Actividad No. 5
Solución de problemas mediante Ascensión de Colinas

Por: Adrian González Pardo

15 de octubre de 2020

1. Funciones a optimizar

Función a evaluar	Forma o descripción de la función
Alpine Function	$f_1(x) = \sum_{i=1}^D x_i \sin(x_i) + 0,1x_i $
Dixon & Price Function	$f_2(x) = (x_1 - 1)^2 \sum_{i=2}^D i (2 \sin(x_i) - x_{i-1})^2$
Quintic Function	$f_3(x) = \sum_{i=1}^D x_i^5 - 3x_i^4 + 4x_i^3 - 2x_i^2 - 10x_i - 4 $
Schwefel 2.23 Function	$f_4(x) = \sum_{i=1}^D x_i^{10}$
Stretched V Sine Wave Function	$f_5(x) = \sum_{i=1}^{D-1} (x_{i+1}^2 + x_i^2)^{0,25} [\sin^2 \{50(x_{i+1}^2 + x_i^2)^{0,1}\} + 1]$
Sum Squares Function	$f_6(x) = \sum_{i=1}^D ix_i^2$

2. Código de implementación

El código fue implementado en lenguaje Ruby para el cálculo de las funciones en D dimensiones y a través de 1 índice se determina la selección de que función se trabajara.

2.1. Función que selecciona que $f(x)$ trabajara

```
1 # Recibe indice de cual funcion trabajara
2 def evaluar(index)
3   if index==0
4     return evaluar_alpine
5   elsif index==1
6     return evaluar_dixon
7   elsif index==2
8     return evaluar_quintic
9   elsif index==3
10    return evaluar_schwefel
11  elsif index==4
12    return evaluar_stretched
13  elsif index==5
14    return evaluar_sum_squares
15  else
16    return 0
17  end
18 end
```

2.2. Función Alpine

```
1 def evaluar_alpine()
2   func=0
3   @lista_xi.each{|i|
4     func+= (i*Math.sin(i)+0.1*i).abs
5   }
6   func
7 end
```

2.3. Función Dixon & Price

```
1 def evaluar_dixon()
2   func=(@lista_xi[0]-1)**2
3   for i in 1..(@dimension-1)
4     func+=(i+1)*((2*Math.sin(@lista_xi[i])-@lista_xi[i-1])**2)
5   end
6   func
7 end
```

2.4. Función Quintic

```
1 def evaluar_quintic()
2   func=0
3   @lista_xi.each{|i|
4     func+=((i**5)-3*(i**4)+4*(i**3)-2*(i**2)-10*i-4).abs
5   }
6   func
7 end
```

2.5. Función Schwefel

```
1 def evaluar_schwefel()
2   func=0
3   @lista_xi.each{|i|
4     func+=i**10
5   }
6   func
7 end
```

```
5 }  
6 func  
7 end
```

2.6. Función Stretched

```
1 def evaluar_stretched()  
2     func=0  
3     for i in 0..(@dimension-2)  
4         func+=(((@lista_xi[i+1]**2)+(@lista_xi[i]**2))*0.25)*((Math.sin(50*((@lista_xi[i+1]**2)+(  
5             @lista_xi[i]**2))*0.1))**2+0.1)  
6     end  
7     func  
8 end
```

2.7. Función Sum Squares

```
1 def evaluar_sum_squares()  
2     func=0  
3     @lista_xi.each_with_index{|i,j|  
4         func+= (j+1)*(i**2)  
5     }  
6     func  
7 end
```

3. Tabla de resultados

	Valor					Tiempo (ms)					
Funcion	Mejor	Peor	Promedio	Mediana	Desviación	Mejor	Peor	Promedio	Mediana	Desviación	D
f1	9,3	39,07	26,245	25,86	7,89	2,96	4,07	26,58	3,16	0,26	10
f1	61,21	104,23	81,79	77,335	11,44	5,53	7,82	6,23	5,94	0,67	30
f2	326,12	3067,91	1856,593	2034,85	806,16	3,77	5,17	1936,91	4,00	0,44	10
f2	10743,86	22784,04	16616,91	15800,955	3145,21	7,84	14,55	8,77	8,37	1,41	30
f3	21577,91	339235,54	121370,027	105829,945	75541,27	5,63	6,97	125094,00	5,84	0,41	10
f3	265655,09	805727,73	511519,31	514130,375	128346,48	13,60	17,76	14,52	14,20	0,96	30
f4	798149999,8	13044318392	6053668361	5450094674	3824123894,37	2,34	4,34	5685739411,80	2,59	0,45	10
f4	6657143215	38914164708	21851371861,68	21751101566	8900476848,01	3,73	5,68	4,42	4,16	0,57	30
f5	6,17	17,5	12,214	11,905	2,79	6,08	12,47	12,09	6,51	1,42	10
f5	32,04	53,93	43,24	42,93	5,61	15,41	21,54	16,71	16,31	1,61	30
f6	666,77	2404,51	1531,797	1518,025	478,62	3,01	4,57	1522,13	3,18	0,41	10
f6	9714,17	20698,17	14935,60	14762,62	2715,56	5,33	7,58	6,00	5,75	0,63	30
Promedio	621300167,2	4329973103	2325475685	2266820873	1060400983,62	6,27	9,38	473822338,35	6,67	0,77	

	Valor					Tiempo (ms)				
Funcion	Mejor	Peor	Promedio	Mediana	Desviación	Mejor	Peor	Promedio	Mediana	Desviación
f1	9,3	39,07	26,245	25,86	7,89	2,96	4,07	26,58	3,16	0,26
f2	326,12	3067,91	1856,593	2034,85	806,16	3,77	5,17	1936,91	4,00	0,44
f3	21577,91	339235,54	121370,027	105829,945	75541,27	5,63	6,97	125094,00	5,84	0,41
f4	798149999,8	13044318392	6053668361	5450094674	3824123894,37	2,34	4,34	5685739411,80	2,59	0,45
f5	6,17	17,5	12,214	11,905	2,79	6,08	12,47	12,09	6,51	1,42
f6	666,77	2404,51	1531,797	1518,025	478,62	3,01	4,57	1522,13	3,18	0,41
Prom	133028764,3	2174110526	1008965526	908367349,1	637366788,52	3,97	6,27	947644667,25	4,21	0,56
	Valor D	10								

	Valor					Tiempo (ms)				
Funcion	Mejor	Peor	Promedio	Mediana	Desviación	Mejor	Peor	Promedio	Mediana	Desviación
f1	61,21	104,23	81,79	77,335	11,44	5,53	7,82	6,23	5,94	0,67
f2	10743,86	22784,04	16616,91	15800,955	3145,21	7,84	14,55	8,77	8,37	1,41
f3	265655,09	805727,73	511519,31	514130,375	128346,48	13,60	17,76	14,52	14,20	0,96
f4	6657143215	38914164708	21851371861,68	21751101566	8900476848,01	3,73	5,68	4,42	4,16	0,57
f5	32,04	53,93	43,24	42,93	5,61	15,41	21,54	16,71	16,31	1,61
f6	9714,17	20698,17	14935,60	14762,62	2715,56	5,33	7,58	6,00	5,75	0,63
Prom	1109571570	6485835679	3641985843,09	3625274397	1483435178,72	8,57	12,49	9,44	9,12	0,97
	Valor D	30								