

Centro de Investigación en Cómputo
Instituto Politécnico Nacional
Metaheurísticas
Actividad No. 10

Solución de problemas mediante heurísticas de trayectoria simple
Curso impartido por: Dra Yenny Villuendas Rey

Adrian González Pardo

1 de noviembre de 2020

1. Algoritmos de RMHC y SA

1.1. RMHC

Es un algoritmo en el cual a partir de una solución aleatoria, modifica bajo una muestra llamada locus la cual constara de un valor aleatorio de 0 a N bits los cuales dependiendo del problema a analizar es como actúa, ya que esta cadena aleatoria de bits nos puede representar un número de casillas a modificar o en su defecto un índice para modificar su valor, finalmente este algoritmo nos permite tener una idea clara donde que también existe un segundo valor aleatoria que puede ir de 0 a $locus - 1$ para obtener un tercer valor aleatorio para modificar y obtener una nueva solución y comparar con respecto a la anterior solución, con ello podremos realizarla una M cantidad de iteraciones, con la esperanza de que se llegue a una mejor solución. Finalmente matemáticamente este algoritmo busca explorar un espacio de soluciones de una función costo y lo que se busca es el optimizar la función con la intención de llegar a un mínimo local o mínimo global.

1.2. SA

Es un algoritmo bioinspirado en el que se basa en un proceso físico donde se busca bajar la temperatura de los metales, el cual es mejor conocido como Recocido de Metales. En este algoritmo podemos realizar casi el mismo procedimiento de búsqueda de solución como lo realiza RMHC, pero con la gran diferencia que en este algoritmo tenemos un factor de probabilidad el cual nos ayuda a tomar una solución la cual no sera factible, pero que nos permitirá seguir buscando otro mínimo local o en el mejor de los casos llegar a un mínimo global.

1.3. Ventajas de ambos algoritmos

- Ambos algoritmos obtienen una solución aleatoria aproximada a la solución determinista del problema
- Utilizan pocos recursos computacionales, con respecto a una implementación determinista
- Permite encontrar soluciones aproximadas a problemas NP-Completo y NP

1.3.1. Desventajas de ambos algoritmos

- RMHC puede quedar atascado en una región y no poder salir de ella, mientras que SA puede llegar al mínimo global pero si en la siguiente iteración se obtiene una probabilidad que acepte la nueva solución esta pueda caer en otra región que no sea el mínimo global.
- Dependiendo de la implementación puede tener una buena solución o simplemente no se llegue a nada
- Estos algoritmos son implementados en problemas de trayectoria simple, es decir, faltaría analizar si puede ser implementado en problemas más complicados.

2. Funciones a optimizar con SA

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	$f_6(x) = \sum_{i=1}^D ix_i^2$

3. 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.

3.1. 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

```

3.2. 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

```

3.3. 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

```

3.4. 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

```

3.5. Función Schwefel

```

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

```

3.6. 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

```

3.7. 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

```

3.8. Programación de las heurísticas

```

1 #!/usr/bin/env ruby
2 require "./punto.rb"
3 class Metaheuristic
4   attr_accessor :lista_x,:imax,:dimension,:max_temp,:max_iter
5   def initialize(dimension=1,imax=100,max_temp=1000,max_iter=500)
6     @dimension=dimension
7     @imax=imax
8     @max_temp=max_temp
9     @max_iter=max_iter
10    @lista_x=Punto.new(@dimension)
11  end
12
13  # Algorithm Random Mutation Hill Climbing

```

```

14 def rmhc(index=0)
15     lista_res=[]
16     lista_iter=[-1]
17     value=0
18     @lista_x.asigna_all
19     f_best=@lista_x.evaluar(index)
20     i=0
21     best_solution=rand(@lista_x.lista_xi.length)
22     lista_res.push("#{sprintf("%.7f",f_best)}")
23     while i<@max_iter
24         locus=rand(best_solution-1)
25         new_best=rand(locus..best_solution).to_i
26         values=@lista_x.lista_xi[new_best]
27         @lista_x.asigna_p(new_best)
28         f_new=@lista_x.evaluar(index)
29         if f_new<=f_best
30             lista_iter.push(i)
31             if f_new==f_best
32                 lista_iter.pop
33             end
34             best_solution=new_best
35             f_best=f_new
36         else
37             @lista_x.lista_xi[new_best]=values
38         end
39         i+=1
40     end
41     lista_res.push("#{lista_iter[-1]}")
42     lista_res.push("#{sprintf("%.7f",f_best)}")
43     lista_res
44 end
45
46 # Algorithm Simulated Annealling
47 def sa(index=0)
48     lista_res=[]
49     lista_iter=[-1]
50     # Genera una solucion aleatoria en todos los puntos
51     @lista_x.asigna_all
52     e_old=@lista_x.evaluar(index)
53     lista_res.push("#{sprintf("%.7f",e_old)}")
54     t=@max_temp
55     t_min=0.0001
56     @alpha=rand(0.88..0.99)
57     j=0
58     mejor=Punto.new(@dimension)
59     lista_res.push("#{sprintf("%.7f",@alpha)}")
60     lista_res.push("#{@max_temp}")
61     lista_res.push("#{t_min}")
62
63     while t>t_min && j<@max_iter
64         i=0
65         while i<@imax && j<@max_iter
66             vecino=rand(@lista_x.dimension-1)
67             old_v=@lista_x.lista_xi[vecino]
68             @lista_x.asigna_p(vecino)
69             e_new=@lista_x.evaluar(index)
70             delta=e_new-e_old
71             if delta>0
72                 if rand(0.0..1.0)>= Math.exp(-delta/t)
73                     @lista_x.lista_xi[vecino]=old_v
74                 else
75                     e_old=e_new
76                 end
77             else
78                 e_old=e_new
79                 @lista_x.lista_xi.each_with_index{|i,j|
80                     mejor.lista_xi[j]=i
81                 }
82                 lista_iter.push(j)
83                 if delta==0
84                     lista_iter.pop

```

```

85         end
86     end
87     i+=1
88     j+=1
89 end
90 t*=@alpha
91 end
92 lista_res.push("#{sprintf("%.7f",t)}")
93 lista_res.push("#{lista_iter[-1]}")
94 lista_res.push("#{sprintf("%.7f",@lista_x.evaluar(index))}")
95 lista_res.push("#{sprintf("%.7f",mejor.evaluar(index))}")
96 lista_res
97 end
98 end

```

4. Características de Hardware

- Procesador Intel Core i5-3210M CPU 2.50GHz
- Memoria RAM DDR3 12 GB
- Sistema Operativo Fedora 32 x86_64
- Ruby 2.7.2p137 para la ejecución

5. Características de ejecución:

- Dimensiones de la función 10 y 30
- I_{max} de SA 10 y 30
- Temperatura máxima 1000
- Máximo número de iteraciones 500
- Temperatura mínima 0,0001

6. Tablas de ejecución

6.1. RMHC

	Evaluación					Tiempo					
Función	Mejor	Peor	Promedio	Mediana	Desviación estandar	Mejor	Peor	Promedio	Mediana	Desviación estandar	D
Alpine	10,6546074	38,0650436	22,45185659	21,4266734	7,317188933	1,9	2,85	2,0295	1,955	0,2144877619	10
Dixon	580,2225102	2675,618473	1556,619701	1589,587055	680,6552386	2,55	3,27	2,6735	2,6	0,1705494218	10
Quintic	2961,573091	261383,1696	118564,9147	121364,3623	80168,04266	4,4	6,26	4,621	4,495	0,4161464823	10
Schwefel	67295438,64	17553938963	5546196795	4409635691	4504604501	1,47	1,7	1,5355	1,54	0,05661271942	10
Stretched	4,2979669	17,9891769	11,64175408	11,42784745	3,230141438	4,76	5	4,838	4,815	0,06477491392	10
Sum Squares	744,7350542	3060,491847	1795,133703	1810,795214	660,3979749	2,09	2,39	2,1565	2,13	0,06853927034	10
	Evaluación					Tiempo					
Función	Mejor	Peor	Promedio	Mediana	Desviación estandar	Mejor	Peor	Promedio	Mediana	Desviación estandar	D
Alpine	47,635283	100,007354	74,4015984	72,4880505	15,3912869	4,5	8,52	4,952	4,71	0,9431447955	30
Dixon	9669,079992	19353,47996	15608,73731	16003,05266	2703,497668	6,39	11,94	7,2575	6,63	1,614147567	30
Quintic	172649,2192	820202,4285	439468,938	436666,5758	155822,1818	11,78	18,08	12,8165	12,255	1,77258442	30
Schwefel	5548452894	40561600490	19485101298	17937449488	10298995909	2,86	3,29	3,0285	3,005	0,1265857313	30
Stretched	30,2496385	56,4925898	42,5988064	41,98725925	7,189081593	13,47	20,38	14,741	13,775	2,242716633	30
Sum Squares	8999,574866	20944,01477	15124,77833	14597,84675	2785,408396	4,36	8,52	4,8005	4,535	0,9202258018	30

6.2. SA

	Evaluación					Tiempo					Temperatura alcanzada					
Función	Mejor	Peor	Promedio	Mediana	Desviación estandar	Mejor	Peor	Promedio	Mediana	Desviación estandar	Mejor	Peor	Promedio	Mediana	Desviación estandar	D
Alpine	18,4779094	45,2344021	27,85765654	25,8091568	7,622193245	2,6	2,94	2,6545	2,625	0,08068620178	2,542292	259,507425	111,5242172	103,16259	97,45703358	10
Dixon	43,0759922	1627,060002	563,834918	542,4811398	464,0398665	3,06	4,39	3,2675	3,205	0,2776001479	3,0429593	406,3332736	153,8742594	125,020127	137,3768293	10
Quintic	46,5076923	127500,9721	24318,49909	8683,267801	39182,06631	4,89	6,37	5,0865	4,96	0,3904690162	1,7835797	437,134951	100,2799507	24,05913295	144,1847743	10
Schwefel	2,2452699	7482654899	718944528,7	601144,8021	1888996557	1,82	2,28	1,931	1,885	0,1205644619	2,8704253	507,7691181	125,3414382	49,86910535	151,3372328	10
Strechd	9,2331228	20,3208	14,28142163	13,9713069	3,404048341	5,46	5,91	5,5685	5,52	0,1220127258	2,5470265	501,3785139	138,2548441	52,0193613	167,2216597	10
Sum Squares	25,989397	2372,779031	687,7717971	606,732258	542,4064061	2,54	2,86	2,6305	2,6	0,07917169882	2,21434	540,4244659	103,6747841	25,4268719	153,3851957	10
	Evaluación					Tiempo					Temperatura alcanzada					
Función	Mejor	Peor	Promedio	Mediana	Desviación estandar	Mejor	Peor	Promedio	Mediana	Desviación estandar	Mejor	Peor	Promedio	Mediana	Desviación estandar	D
Alpine	68,143043	112,4999176	91,20472396	93,1181025	11,36325677	5,71	11,92	6,4775	5,88	1,573343323	118,1728169	722,8941978	315,20808	217,5504376	198,6773466	30
Dixon	1985,911179	10503,34127	5144,297399	4816,437091	2320,141151	7,23	12,75	8,3735	7,59	1,740009906	121,1598771	835,4908671	388,9527187	340,8057764	227,5389676	30
Quintic	465,5830539	86773,07495	16820,1839	4658,684987	27030,82626	12,54	15,39	13,219	13,055	0,6859330948	125,0373318	743,6180894	347,9874207	307,200305	189,3302366	30
Schwefel	988,4203779	5214641936	793415984,1	1494542,94	1764220651	3,43	4,74	3,6465	3,54	0,3017540825	122,0761572	839,431176	336,4815582	252,7888108	218,8429581	30
Strechd	35,671384	62,5609186	48,54872318	48,01247325	7,15089101	14,86	26,96	16,8895	15,165	3,408122612	115,8722122	738,6450409	334,0648567	311,5499471	189,2431628	30
Sum Squares	2503,366881	10547,22422	5982,697158	5702,70166	1873,791458	5,25	9,91	5,9445	5,53	1,185555499	117,2162961	837,7447155	374,9062726	281,2870295	255,0079772	30

6.3. Discusión respecto a la implementación RMHC y SA

Si bien podemos ejecutar ambos algoritmos con algunas condiciones compartidas, como es el caso de compartir el máximo número de iteraciones que admite el programa y como fue indicado, la obtención de sus soluciones y por ende de los valores que retorna el programa son distintos respecto a una iteración, debido a que una vez entra una nueva iteración en la llamada del algoritmo RMHC o SA este genera una solución aleatoria que esta mostrada al menos en las tablas de SA particulares de cada función, por ello podemos que el hecho de implementar alguno de los dos algoritmos nos es conveniente saber que al menos los algoritmos actúan de forma en que la iteración 0 genere una solución aleatoria, y sobre ella podamos evolucionar aleatoriamente o bajo el esquema de recocido, obteniendo soluciones aceptables donde al momento de ser evaluadas por la computadora estas no demoren en mostrar su valor numérico y tengan proximidad a su mínimo global o local.

6.4. Tablas particulares

6.4.1. Alpine

Iteracion	Sol_al	Alpha	T_max	T_min	T_final	Iteracion_donde_se_encontro_un_minimo	Sol_final_algoritmo	Sol_mejor	Tiempo ms	Dimension	Se modifiko?
1	26,6049199	0,9316372	1000	0,0001	28,9962209	496	29,7634716	29,5319995	2,94	10	FALSE
2	14,4191841	0,8933262	1000	0,0001	3,5522572	498	25,2555121	19,013183	2,7	10	FALSE
3	27,8049893	0,8930025	1000	0,0001	3,488455	495	25,0717629	20,8036838	2,64	10	FALSE
4	38,0049398	0,9728044	1000	0,0001	251,9274302	499	22,0863319	22,0863319	2,6	10	TRUE
5	26,66529	0,9064725	1000	0,0001	7,374472	495	36,1013728	25,0096642	2,62	10	FALSE
6	29,3893254	0,9630085	1000	0,0001	151,8820772	498	45,2344021	42,9950386	2,69	10	FALSE
7	25,9244696	0,9535364	1000	0,0001	92,6540495	499	34,4368761	34,4368761	2,66	10	TRUE
8	23,8669392	0,8874127	1000	0,0001	2,5484841	495	18,4779094	13,7899013	2,61	10	FALSE
9	32,1978103	0,9714061	1000	0,0001	234,4448292	495	37,3395152	26,6659636	2,63	10	FALSE
10	24,8390271	0,9574432	1000	0,0001	113,6711305	499	20,2977838	20,2977838	2,66	10	TRUE
11	33,5245937	0,8873695	1000	0,0001	2,542292	499	20,723457	20,723457	2,6	10	TRUE
12	21,5325376	0,960705	1000	0,0001	134,7417004	498	20,1661058	18,7046025	2,61	10	FALSE
13	36,1993463	0,9480784	1000	0,0001	69,5364868	495	34,6385311	26,2565437	2,63	10	FALSE
14	25,6893653	0,9436756	1000	0,0001	55,0981656	499	26,3628015	26,3628015	2,78	10	TRUE
15	33,4434821	0,9180471	1000	0,0001	13,9074722	497	39,4556689	36,1060375	2,62	10	FALSE
16	23,8938525	0,9699941	1000	0,0001	217,9988948	498	28,755826	24,6305393	2,6	10	FALSE
17	36,8410066	0,9733813	1000	0,0001	259,507425	499	21,1334128	21,1334128	2,6	10	TRUE
18	25,5097727	0,9581628	1000	0,0001	118,0227932	498	27,2466975	26,3169995	2,67	10	FALSE
19	46,7604584	0,9726242	1000	0,0001	249,6049788	499	21,0261177	21,0261177	2,62	10	TRUE
20	28,237827	0,9700816	1000	0,0001	218,98473	498	23,5795746	20,4822748	2,61	10	FALSE
1	98,2472953	0,8973088	1000	0,0001	158,4938964	499	83,0274031	83,0274031	5,96	30	TRUE
2	95,7417024	0,9066139	1000	0,0001	188,8775993	498	98,9218127	94,5643517	5,76	30	FALSE
3	94,052753	0,940147	1000	0,0001	350,209638	497	112,4999176	107,3693987	5,71	30	FALSE
4	94,1844541	0,8819464	1000	0,0001	118,1728169	497	82,7807313	77,6424757	5,78	30	FALSE
5	101,224568	0,9660727	1000	0,0001	556,1181847	499	96,9601232	96,9601232	5,72	30	TRUE
6	92,5782381	0,8891397	1000	0,0001	135,6718236	498	105,2901604	101,1655603	5,79	30	FALSE
7	95,2859105	0,9810932	1000	0,0001	722,8941978	497	90,8814716	86,8743212	5,98	30	FALSE
8	102,649193	0,9188728	1000	0,0001	237,3242088	498	78,107488	77,9823226	6,18	30	FALSE
9	72,1689146	0,8969326	1000	0,0001	157,3682271	499	86,4214696	86,4214696	6,57	30	TRUE
10	83,3754601	0,9361947	1000	0,0001	326,0054279	499	95,4342397	95,4342397	5,84	30	TRUE
11	81,167929	0,9534709	1000	0,0001	444,8634634	499	95,914884	95,914884	5,83	30	TRUE
12	119,8440833	0,9041605	1000	0,0001	180,3740318	495	95,429798	87,6252566	5,96	30	FALSE
13	75,3569531	0,9738204	1000	0,0001	637,0023243	499	71,5459249	71,5459249	5,88	30	TRUE
14	113,2487179	0,9090725	1000	0,0001	197,7766663	498	97,7055522	94,6015245	11,92	30	FALSE
15	110,5288744	0,9362765	1000	0,0001	326,4903993	497	68,143043	66,1305739	6,2	30	FALSE
16	97,1181521	0,9747489	1000	0,0001	647,4064785	499	85,3460859	85,3460859	5,88	30	TRUE
17	82,9235746	0,8981685	1000	0,0001	161,0953952	498	87,9713164	87,2324257	9,78	30	FALSE
18	99,567801	0,8918333	1000	0,0001	142,8311361	499	88,2129912	88,2129912	5,83	30	TRUE
19	98,2078318	0,9587317	1000	0,0001	488,4844327	497	108,145333	104,1359725	5,8	30	FALSE
20	86,9744378	0,885569	1000	0,0001	126,7012509	499	95,3547334	95,3547334	7,18	30	TRUE

6.4.2. Dixon

Iteracion	Sol_al	Alpha	T_max	T_min	T_final	Iteracion_donde_se_encontro_un_minimo	Sol_final_algoritmo	Sol_mejor	Tiempo ms	Dimension	Se modifico?
1	2228,997202	0,9640971	1000	0,0001	160,7084023	496	574,4675197	510,0417926	4,39	10	FALSE
2	1828,169795	0,8905656	1000	0,0001	3,0429593	483	43,0759922	43,0759922	3,15	10	TRUE
3	1278,908571	0,9751522	1000	0,0001	284,1977323	496	677,3768967	631,7095582	3,21	10	FALSE
4	2173,557592	0,9821496	1000	0,0001	406,3332736	499	278,3323475	278,3323475	3,22	10	TRUE
5	3443,147072	0,9336467	1000	0,0001	32,2944047	498	275,3095042	275,3095042	3,14	10	TRUE
6	3159,304494	0,9291379	1000	0,0001	25,3518624	499	50,7655132	50,7655132	3,12	10	TRUE
7	2865,491742	0,958149	1000	0,0001	117,9378776	496	337,0164699	337,0164699	3,2	10	TRUE
8	954,1068477	0,932138	1000	0,0001	29,7859056	490	201,0751952	201,0751952	3,16	10	TRUE
9	1254,858616	0,960325	1000	0,0001	132,1023764	498	790,7743766	710,9276769	3,28	10	FALSE
10	1343,438238	0,9621176	1000	0,0001	145,0133709	498	760,1548163	760,1548163	3,34	10	TRUE
11	2820,466687	0,975788	1000	0,0001	293,6113768	496	925,980507	537,4348066	3,25	10	FALSE
12	2515,713649	0,979095	1000	0,0001	347,7290615	494	1548,616844	1042,603405	3,38	10	FALSE
13	3461,787607	0,9776269	1000	0,0001	322,5947438	498	906,1723624	905,0508034	3,25	10	FALSE
14	1531,923191	0,9537164	1000	0,0001	93,5324603	498	510,4947598	422,3369562	3,34	10	FALSE
15	3815,245766	0,9162662	1000	0,0001	12,6206989	484	81,0881499	70,7898884	3,11	10	FALSE
16	2967,07085	0,9033159	1000	0,0001	6,1941661	476	84,9761349	79,1982637	3,06	10	FALSE
17	2957,732233	0,9453638	1000	0,0001	60,2486651	496	658,5950536	658,5950536	3,17	10	TRUE
18	1752,608609	0,979379	1000	0,0001	352,8091743	499	837,833341	837,833341	3,27	10	TRUE
19	1097,330438	0,9717679	1000	0,0001	238,8514476	496	1627,060002	1284,513303	3,2	10	FALSE
20	2592,129305	0,9161271	1000	0,0001	12,5252291	499	107,532573	107,532573	3,11	10	TRUE
1	15927,41222	0,8875729	1000	0,0001	131,6645684	499	3054,160803	3054,160803	7,74	30	TRUE
2	16457,64518	0,989483	1000	0,0001	835,4908671	497	5567,052219	5108,254348	7,5	30	FALSE
3	10771,72721	0,8832424	1000	0,0001	121,1598771	492	1985,911179	1797,580357	12,35	30	FALSE
4	16914,11476	0,9471766	1000	0,0001	397,4899704	497	4117,629982	4117,629982	7,46	30	TRUE
5	22055,20018	0,8937737	1000	0,0001	148,2069189	496	5000,310315	5000,310315	7,37	30	TRUE
6	16388,05008	0,9044094	1000	0,0001	181,2201111	497	3604,396304	3562,635614	7,6	30	FALSE
7	18471,68596	0,9765817	1000	0,0001	668,4149316	499	10503,34127	10503,34127	7,43	30	TRUE
8	17877,6394	0,9569314	1000	0,0001	473,1227038	496	4632,563867	4490,075993	7,67	30	FALSE
9	9284,684807	0,9756396	1000	0,0001	657,53743	495	8531,878294	7608,588173	10,02	30	FALSE
10	19430,29398	0,8878306	1000	0,0001	132,3158843	492	3219,713331	2687,102281	7,23	30	FALSE
11	19951,53665	0,9099941	1000	0,0001	201,213046	498	2616,13402	2616,13402	7,55	30	TRUE
12	18141,1528	0,9377463	1000	0,0001	335,3136352	496	6553,089508	5760,634834	7,75	30	FALSE
13	19029,80314	0,9861814	1000	0,0001	789,3428525	499	4286,557089	4286,557089	7,58	30	TRUE
14	17540,63629	0,9590045	1000	0,0001	490,8526334	497	5435,93187	5267,774379	12,75	30	FALSE
15	17954,427	0,9376897	1000	0,0001	334,9696369	499	5995,783095	5995,783095	7,63	30	TRUE
16	11873,14641	0,9301899	1000	0,0001	292,2249758	496	3879,199975	3879,199975	7,57	30	TRUE
17	12629,81729	0,9682492	1000	0,0001	577,8048456	498	9403,055409	9403,055409	11,36	30	TRUE
18	12844,41684	0,9601651	1000	0,0001	501,0494175	495	6135,556273	5803,606842	7,46	30	FALSE
19	16165,74243	0,939526	1000	0,0001	346,2979176	494	6138,036082	6061,195769	7,55	30	FALSE
20	18766,0275	0,8989071	1000	0,0001	163,3621506	499	2225,647089	2225,647089	7,9	30	TRUE

6.4.3. Quintic

Iteracion	Sol_al	Alpha	T_max	T_min	T_final	Iteracion_donde_se_encontro_un_minimo	Sol_final_algoritmo	Sol_mejor	Tiempo ms	Dimension	Se modifico?
1	153353,9439	0,9732446	1000	0,0001	257,6909128	498	127500,9721	127500,9721	6,37	10	TRUE
2	211913,5427	0,8930027	1000	0,0001	3,4885055	486	495,8938451	495,8938451	4,94	10	TRUE
3	154698,9353	0,9266665	1000	0,0001	22,1908554	493	34980,14884	34951,4385	4,97	10	FALSE
4	249475,2726	0,9133373	1000	0,0001	10,7537314	468	73639,51282	73624,82187	4,91	10	FALSE
5	161979,901	0,9671093	1000	0,0001	187,8353144	496	551,4469982	551,4469982	4,92	10	TRUE
6	75722,32371	0,9835859	1000	0,0001	437,134951	496	14060,19994	13933,46788	4,95	10	FALSE
7	190987,1076	0,8993336	1000	0,0001	4,9663914	479	13165,45285	13156,35749	4,89	10	FALSE
8	188535,88	0,9324186	1000	0,0001	30,2374878	481	1079,517211	1068,107197	4,92	10	FALSE
9	51776,51775	0,9180272	1000	0,0001	13,8924284	497	156,1637568	156,1637568	6,04	10	TRUE
10	104265,6202	0,8979523	1000	0,0001	4,5990094	483	29709,75584	29707,14916	4,89	10	FALSE
11	167846,7752	0,9737908	1000	0,0001	265,0230604	495	1195,34777	1184,369294	5	10	FALSE
12	191277,9212	0,8811013	1000	0,0001	1,7835797	496	1801,442432	1801,442432	5,04	10	TRUE
13	174626,0154	0,9396581	1000	0,0001	44,5135951	495	15343,05793	15339,02893	5,05	10	FALSE
14	89167,44402	0,8955699	1000	0,0001	4,0269342	482	46,5076923	38,4953435	5,09	10	FALSE
15	93182,40533	0,9387768	1000	0,0001	42,4734513	460	5097,212974	5097,212974	4,96	10	TRUE
16	187086,4941	0,9830337	1000	0,0001	425,0309848	494	1113,84577	1102,015147	4,96	10	FALSE
17	189609,4116	0,8983386	1000	0,0001	4,698963	494	26192,78973	26192,43893	4,9	10	FALSE
18	179450,5077	0,9295552	1000	0,0001	25,9274105	490	2558,173097	2552,630589	5,01	10	FALSE
19	117112,5936	0,9697122	1000	0,0001	214,8541118	495	12269,32263	12269,32263	5,02	10	TRUE
20	269350,2882	0,8974709	1000	0,0001	4,4773354	478	125413,2176	125411,7556	4,9	10	FALSE
1	595958,148	0,9241399	1000	0,0001	261,5420579	493	4080,927475	4077,438227	12,65	30	FALSE
2	445074,4538	0,9015969	1000	0,0001	171,8742928	499	2095,071085	2095,071085	12,54	30	TRUE
3	446699,0858	0,8916743	1000	0,0001	142,3987528	497	1123,891719	1123,891719	13,23	30	TRUE
4	891539,6529	0,9739639	1000	0,0001	638,5998135	494	4541,093821	4145,403619	12,86	30	FALSE
5	207520,8398	0,9827258	1000	0,0001	743,6180894	489	16963,14404	13410,5955	12,79	30	FALSE
6	639424,8087	0,9393218	1000	0,0001	345,0202277	496	4125,467006	3914,142464	13,6	30	FALSE
7	725016,6787	0,9392786	1000	0,0001	344,7510751	482	9712,894657	9664,191837	12,97	30	FALSE
8	556322,5374	0,8848806	1000	0,0001	125,0373318	494	892,9661366	892,9661366	13,39	30	TRUE
9	474055,998	0,9564098	1000	0,0001	468,7576231	497	9772,56122	9772,56122	13,41	30	TRUE
10	593687,5357	0,9595989	1000	0,0001	496,0499599	491	82979,65499	82944,85823	12,96	30	FALSE
11	270616,6496	0,9422593	1000	0,0001	363,828933	499	3089,485599	3089,485599	13,92	30	TRUE
12	696812,3917	0,9718967	1000	0,0001	615,9444266	499	6278,762798	6278,762798	14,04	30	TRUE
13	633803,5125	0,8876745	1000	0,0001	131,9209496	496	1301,626899	1292,761154	13,19	30	FALSE
14	766948,6046	0,9110146	1000	0,0001	205,0834124	475	86773,07495	86255,46463	15,39	30	FALSE
15	656380,8859	0,9029993	1000	0,0001	176,4760811	497	24206,97748	24206,97748	12,71	30	TRUE
16	586497,2141	0,9639288	1000	0,0001	535,5058822	499	62008,86317	62008,86317	12,59	30	TRUE
17	329289,9614	0,9086934	1000	0,0001	196,379269	494	4776,276154	4514,786659	13,73	30	FALSE
18	332059,1627	0,925801	1000	0,0001	269,6495348	489	2458,79517	2222,603344	12,73	30	FALSE
19	376954,8392	0,9605686	1000	0,0001	504,6411773	480	8756,560539	7704,074533	12,54	30	FALSE
20	717651,292	0,9154341	1000	0,0001	222,6695247	497	465,5830539	465,5830539	13,14	30	TRUE

6.4.4. Schwefel

Iteracion	Sol_al	Alpha	T_max	T_min	T_final	Iteracion_donde_se_encontro_un_minimo	Sol_final_algoritmo	Sol_mejor	Tiempo ms	Dimension	Se modifiko?
1	4493528062	0,9073486	1000	0,0001	7,7394247	468	2821785,168	2821785,168	1,9	10	TRUE
2	24316728270	0,967242	1000	0,0001	189,1284086	498	9688189,992	9688189,992	1,98	10	TRUE
3	10587662054	0,9776995	1000	0,0001	323,7953845	490	5764,803705	5764,803705	1,84	10	TRUE
4	11717178310	0,9254239	1000	0,0001	20,7508454	496	87012881,41	87012881,41	1,85	10	TRUE
5	8944908347	0,9865369	1000	0,0001	507,7691181	480	37,6710105	37,670982	1,95	10	FALSE
6	14893041238	0,9306167	1000	0,0001	27,4500012	471	1981459808	1981459794	2,28	10	FALSE
7	19025317335	0,9410689	1000	0,0001	47,9812288	499	450030,6383	450030,6383	1,9	10	TRUE
8	7027462940	0,8974463	1000	0,0001	4,4711883	499	2,2452699	2,2452699	1,84	10	TRUE
9	10739220107	0,9000926	1000	0,0001	5,1803521	494	752258,966	752258,966	1,87	10	TRUE
10	2486529163	0,9791837	1000	0,0001	349,309016	483	21,7683423	13,8297988	1,83	10	FALSE
11	8248722116	0,9597913	1000	0,0001	128,4817019	486	5445809,43	5445808,917	2,05	10	FALSE
12	4258639259	0,8958695	1000	0,0001	4,0948513	477	95784846,39	95784842,86	1,85	10	FALSE
13	5717327160	0,8895266	1000	0,0001	2,8704253	491	215,1241594	215,1241594	1,82	10	TRUE
14	12551605300	0,960072	1000	0,0001	130,3739681	497	472359165,9	472359165,9	1,86	10	TRUE
15	8808133149	0,9808602	1000	0,0001	380,5005364	493	402,5632011	402,5632011	1,85	10	TRUE
16	6769557511	0,9424957	1000	0,0001	51,7569819	483	55589,40787	55589,40787	1,83	10	TRUE
17	17234092151	0,9665536	1000	0,0001	182,514043	496	77,4129032	77,4129032	2,03	10	TRUE
18	16733294501	0,9346768	1000	0,0001	34,1250625	497	213908,1301	213907,7135	1,99	10	FALSE
19	10807655156	0,9547533	1000	0,0001	98,7546674	458	7482654899	7482654899	1,96	10	TRUE
20	11551256189	0,9116081	1000	0,0001	9,7815584	499	4240184878	4240184878	2,14	10	TRUE
1	23320435900	0,883634	1000	0,0001	122,0761572	484	583901,8079	583821,6136	3,49	30	FALSE
2	47857408366	0,9226643	1000	0,0001	254,5323735	493	145242,5212	145242,5212	3,66	30	FALSE
3	30090066853	0,9384118	1000	0,0001	339,382047	497	2163,08753	2163,08753	3,46	30	TRUE
4	35405194154	0,9897569	1000	0,0001	839,431176	492	2228471,246	2228471,246	3,57	30	TRUE
5	24220374832	0,9348495	1000	0,0001	318,1330673	468	86566930,33	86566910,52	3,53	30	FALSE
6	27544535272	0,9219159	1000	0,0001	251,0452481	499	5653362,261	5653362,261	3,56	30	TRUE
7	27746899849	0,8836898	1000	0,0001	122,2073812	499	283586,997	283586,997	3,76	30	TRUE
8	26980938980	0,9057842	1000	0,0001	185,9606026	491	68076,81912	67975,39187	4,74	30	FALSE
9	30635426466	0,9618666	1000	0,0001	516,3601418	481	4122402503	4122402474	3,51	30	FALSE
10	26545481139	0,8914321	1000	0,0001	141,7427547	497	21739401,7	21739400,44	3,59	30	FALSE
11	27596564328	0,9852346	1000	0,0001	776,5585959	484	1004065417	1004065417	3,82	30	TRUE
12	36089347768	0,9020487	1000	0,0001	173,3444703	487	5214641936	5214641936	3,47	30	TRUE
13	22437472529	0,955071	1000	0,0001	457,7262365	485	5146805700	5146805700	3,43	30	FALSE
14	34155828969	0,9156446	1000	0,0001	223,5412001	496	988,4203779	763,0179492	3,97	30	FALSE
15	26563658987	0,9337066	1000	0,0001	311,5852816	480	760614,6338	760597,517	3,51	30	FALSE
16	23713770864	0,9017476	1000	0,0001	172,3634286	479	145238677,4	145238474,5	3,53	30	FALSE
17	29501180319	0,8951286	1000	0,0001	152,0729505	496	303188,904	303188,904	3,91	30	TRUE
18	21235580419	0,9695798	1000	0,0001	591,4531238	492	116816536,9	116816341,9	3,43	30	FALSE
19	13727867649	0,9691726	1000	0,0001	587,244295	489	2102,112117	1880,591334	3,44	30	FALSE
20	17807354568	0,9077303	1000	0,0001	192,8706324	491	10879,93139	10752,57936	3,55	30	FALSE

6.4.5. Streched

Iteracion	Sol_al	Alpha	T_max	T_min	T_final	Iteracion_donde_se_encontro_un_minimo	Sol_final_algoritmo	Sol_mejor	Tiempo ms	Dimension	Se modifiko?
1	13,7346433	0,9421053	1000	0,0001	50,6959316	499	9,4323843	9,4323843	5,91	10	TRUE
2	10,2076972	0,9139639	1000	0,0001	11,12888	499	15,9677152	15,9677152	5,52	10	TRUE
3	13,7055444	0,9116282	1000	0,0001	9,7923727	499	9,9633234	9,9633234	5,57	10	TRUE
4	15,4503927	0,9372093	1000	0,0001	39,0687652	498	15,0457459	14,7262131	5,51	10	FALSE
5	10,8318418	0,8911148	1000	0,0001	3,1382239	495	9,2331228	9,2331228	5,47	10	TRUE
6	6,5845494	0,8922518	1000	0,0001	3,3448192	497	15,1998431	14,0715736	5,53	10	FALSE
7	13,7510457	0,9038355	1000	0,0001	6,3748439	499	14,2772402	14,2772402	5,8	10	TRUE
8	9,57982	0,9732241	1000	0,0001	257,4208074	497	13,0519413	10,4946444	5,57	10	FALSE
9	13,7986051	0,8874025	1000	0,0001	2,5470265	496	12,0209375	9,6748691	5,67	10	FALSE
10	14,8117878	0,986287	1000	0,0001	501,3785139	499	17,123913	17,123913	5,52	10	TRUE
11	16,5535302	0,9531154	1000	0,0001	90,6305741	499	12,1471565	12,1471565	5,49	10	TRUE
12	13,1323444	0,9831807	1000	0,0001	428,2211835	499	17,4639592	17,4639592	5,58	10	TRUE
13	14,2281109	0,9430648	1000	0,0001	53,342791	499	13,4592834	13,4592834	5,46	10	TRUE
14	12,0924605	0,978447	1000	0,0001	336,4073192	499	12,6715487	12,6715487	5,6	10	TRUE
15	16,7011818	0,9826681	1000	0,0001	417,1989379	499	9,5249055	9,5249055	5,49	10	TRUE
16	16,5058272	0,9461109	1000	0,0001	62,6761286	498	13,6653736	13,4079544	5,49	10	FALSE
17	16,8517223	0,974564	1000	0,0001	275,75137	498	17,9408886	14,7775646	5,74	10	FALSE
18	11,1339504	0,8954489	1000	0,0001	3,9998082	494	16,8088746	11,6222069	5,47	10	FALSE
19	19,3421274	0,9648396	1000	0,0001	167,014858	498	20,3094757	20,2772854	5,5	10	FALSE
20	15,8467952	0,9398472	1000	0,0001	44,9637275	495	20,3208	9,4949546	5,48	10	FALSE
1	58,7086838	0,9461088	1000	0,0001	389,9407603	497	52,5087029	47,9380703	15,07	30	FALSE
2	48,4157714	0,9339976	1000	0,0001	313,2406062	499	45,4365687	45,4365687	14,9	30	TRUE
3	49,3161784	0,9738387	1000	0,0001	637,2049325	498	47,5087265	46,8955521	15,14	30	FALSE
4	49,863121	0,9232922	1000	0,0001	257,4931514	496	62,5609186	53,0052302	14,86	30	FALSE
5	26,9830841	0,982338	1000	0,0001	738,6450409	498	37,9591614	37,7269068	15,19	30	FALSE
6	47,0937386	0,9443382	1000	0,0001	377,7189405	499	45,31622	45,31622	15,2	30	TRUE
7	46,3802081	0,8875343	1000	0,0001	131,5671665	497	57,9458861	48,8787963	14,98	30	FALSE
8	41,7788537	0,9060165	1000	0,0001	186,7727344	499	48,6970918	48,6970918	20,88	30	TRUE
9	46,6668425	0,8812771	1000	0,0001	116,6573663	498	41,8845599	37,1897646	15,05	30	FALSE
10	44,1813663	0,9360514	1000	0,0001	325,1583652	499	47,5736866	47,5736866	15,38	30	TRUE
11	50,4094175	0,9594019	1000	0,0001	494,3221124	498	48,059889	45,8724271	21,74	30	FALSE
12	45,1109106	0,8966051	1000	0,0001	156,3942842	495	58,5304132	47,232214	15,62	30	FALSE
13	57,1619227	0,92714	1000	0,0001	276,356951	498	47,9650575	45,0068571	15,9	30	FALSE
14	54,1173976	0,9338151	1000	0,0001	312,2017684	499	49,8595013	49,8595013	26,96	30	TRUE
15	46,6915005	0,8809271	1000	0,0001	115,8722122	495	57,9676481	51,9946897	15,02	30	FALSE
16	38,4575535	0,9801063	1000	0,0001	710,6306163	499	35,671384	35,671384	14,94	30	TRUE
17	50,120455	0,9079925	1000	0,0001	193,819805	498	48,4587111	47,5156007	14,91	30	FALSE
18	46,7301629	0,9072314	1000	0,0001	191,0765882	499	39,6396151	39,6396151	15	30	TRUE
19	42,2000674	0,9335853	1000	0,0001	310,8981257	499	54,0347283	54,0347283	22,33	30	TRUE
20	43,093739	0,9535292	1000	0,0001	445,3256063	498	43,3959934	40,281618	18,72	30	FALSE

6.4.6. Sum Squares

Iteracion	Sol_al	Alpha	T_max	T_min	T_final	Iteracion_donde_se_encontro_un_minimo	Sol_final_algoritmo	Sol_mejor	Tiempo ms	Dimension	Se modifico?
1	1353,598579	0,9400708	1000	0,0001	45,5018163	495	357,0394708	310,4318529	2,64	10	FALSE
2	2586,031008	0,930312	1000	0,0001	27,0042135	491	245,6818334	245,6818334	2,64	10	TRUE
3	2607,182445	0,9792726	1000	0,0001	350,8969524	499	1314,914931	1314,914931	2,7	10	TRUE
4	2172,170967	0,9705695	1000	0,0001	224,5594386	498	575,0425537	575,0425537	2,65	10	TRUE
5	1102,008549	0,941299	1000	0,0001	48,5713673	499	318,9090357	318,9090357	2,58	10	TRUE
6	1814,351255	0,9034596	1000	0,0001	6,2436261	482	258,5653154	258,5653154	2,59	10	TRUE
7	2830,318924	0,9181064	1000	0,0001	13,952495	491	285,0591714	264,7700532	2,58	10	FALSE
8	2341,48055	0,9358515	1000	0,0001	36,3368169	499	884,4727268	884,4727268	2,56	10	TRUE
9	867,3381886	0,8873151	1000	0,0001	2,5345056	499	126,2672197	126,2672197	2,55	10	TRUE
10	965,3854202	0,9579272	1000	0,0001	116,580367	499	797,6591002	797,6591002	2,59	10	TRUE
11	1209,246644	0,9253218	1000	0,0001	20,6366314	498	256,6419213	256,6419213	2,54	10	TRUE
12	1661,064647	0,8873784	1000	0,0001	2,5435603	482	25,989397	21,2465625	2,69	10	FALSE
13	1932,611039	0,9770009	1000	0,0001	312,4274197	499	1082,279602	1082,279602	2,72	10	TRUE
14	1420,295678	0,9280035	1000	0,0001	23,8495303	497	549,405694	549,405694	2,86	10	TRUE
15	1749,587107	0,9877674	1000	0,0001	540,4244659	498	2372,779031	2340,154517	2,74	10	FALSE
16	2333,001471	0,8973123	1000	0,0001	4,4379417	489	638,4219623	635,9999814	2,59	10	FALSE
17	2615,678337	0,8849218	1000	0,0001	2,21434	485	969,0707356	969,0707356	2,61	10	TRUE
18	1737,907196	0,9173858	1000	0,0001	13,4153248	486	657,8831867	657,8831867	2,57	10	TRUE
19	2297,453063	0,9747154	1000	0,0001	277,9019761	496	1231,687383	847,7599745	2,64	10	FALSE
20	2603,858248	0,8928711	1000	0,0001	3,4628927	493	807,6656699	803,5365454	2,57	10	FALSE
1	16469,07673	0,9094149	1000	0,0001	199,0469021	499	4282,316798	4282,316798	5,28	30	TRUE
2	17625,25805	0,9077856	1000	0,0001	193,0704882	497	6083,991628	6051,868812	5,34	30	FALSE
3	12988,22734	0,961659	1000	0,0001	514,4686696	499	10547,22422	10547,22422	5,45	30	TRUE
4	14684,39039	0,9818789	1000	0,0001	732,7983413	499	8827,815033	8827,815033	5,5	30	TRUE
5	18441,03245	0,9309058	1000	0,0001	296,0718603	499	5094,581681	5094,581681	5,46	30	TRUE
6	13425,06935	0,9733275	1000	0,0001	631,542753	499	6009,179312	6009,179312	5,57	30	TRUE
7	15639,88644	0,8931071	1000	0,0001	146,3389179	489	2503,366881	2451,283484	5,54	30	FALSE
8	18034,30528	0,8970753	1000	0,0001	157,7942498	495	5329,160143	5181,209458	7,84	30	FALSE
9	19984,61451	0,9315495	1000	0,0001	299,571961	499	5698,004235	5698,004235	5,39	30	TRUE
10	17343,04786	0,9855859	1000	0,0001	781,2796608	499	4691,821635	4691,821635	5,55	30	TRUE
11	14586,25141	0,951447	1000	0,0001	429,0802429	499	7137,202399	7137,202399	7,82	30	TRUE
12	15790,11193	0,9890821	1000	0,0001	829,7553244	498	9293,351207	9293,351207	5,62	30	TRUE
13	21027,26854	0,9461796	1000	0,0001	390,4367399	497	5268,515476	4870,390392	5,69	30	FALSE
14	18415,09154	0,9251618	1000	0,0001	266,5021986	498	6766,745336	6590,821291	5,52	30	FALSE
15	11740,32188	0,9180921	1000	0,0001	233,9196055	497	4814,772396	4814,772396	5,55	30	TRUE
16	18286,75447	0,9090057	1000	0,0001	197,5295995	493	5795,451824	5238,504216	5,25	30	FALSE
17	18868,11054	0,8839461	1000	0,0001	122,8112366	496	4846,918635	4846,918635	5,32	30	TRUE
18	14253,70825	0,9896398	1000	0,0001	837,7447155	498	6726,586698	6726,586698	5,47	30	TRUE
19	17695,94	0,8832364	1000	0,0001	121,1456892	496	4229,538545	4229,538545	9,91	30	TRUE
20	15120,84157	0,8815249	1000	0,0001	117,2162961	498	5707,399084	5707,399084	5,82	30	TRUE

6.5. Discusión respecto a las tablas del algoritmo SA

Finalmente de forma detallada podemos ver los resultados y la aleatoriedad de los valores programados y obtenidos de por lo que en algunas ocasiones el mínimo obtenido se ve modificado por uno distinto al valor que obtuvimos y que era mejor, en las tablas podemos ver que en están en el sentido inverso, es decir, un valor *False* es equivalente a decir que es un valor *Modificado* en la solución, mientras que un valor *True* es equivalente a *NoModificado*, por lo tanto dependiendo de nuestra función de probabilidad sea aceptable y de que el equipo genere un vecino cuya solución es peor a la solución vieja, esta modifique la solución y las siguientes iteraciones no se vean afectadas por las nuevas soluciones "mejores".^a la solución que fue aceptada, por otro lado también podemos ver la forma de actuar de la aleatoriedad del valor α actúa de forma en que la temperatura al final de la ejecución es aceptablemente baja o en su defecto es bastante alta considerada a otras temperaturas.

7. Conclusiones

Los algoritmos heurísticos de trayectoria simple como son RMHC y SA, nos permiten computar y obtener soluciones aproximadas al mínimo local o global de cualquier función de optimización cuyo espacio de soluciones cae en la categoría de problemas NP y NP-Complejos, por lo que podemos aproximar una solución aceptable en tiempo y recursos de memoria que nos permita comprender y analizar el espacio de soluciones de modo que al ser implementados de una forma, estos se puedan hibridar con otros algoritmos u otros problemas a optimizar. Para finalizar es importante reconocer que existen más heurísticas que permiten abordar otros problemas que no son de trayectoria simple, por lo que en un futuro, así como en los papers de investigación es posible realizar una implementación híbrida en distintos dominios (Discreto, Continuo o Discontinuo), de modo que podamos modelar y solucionar una problemática de planificación, reconocimiento de patrones y/u optimización en general.