

---

# INSTITUTO POLITÉCNICO NACIONAL

Centro de Investigación en Computación

---



ASIGNATURA:

Metaheurísticas

Actividad #17:

Guía Taller Laboratorio No.8

"Solución de problemas mediante Recocido Simulado"

PROFESORA:

Dra. Yenny Villuendas Rey

PRESENTA:

Juan René Hernández Sánchez

Adriana Montserrat García Carrillo



Centro de Investigación  
en Computación  
Instituto Politécnico Nacional

## 1. Introducción

Muchos problemas de ingeniería, planificación y fabricación pueden ser modelados como minimizar o maximizar una función de coste sobre un conjunto finito de variables discretas. Esta clase de problemas, llamados de optimización combinatoria, ha recibido mucha atención en las dos últimas décadas y se han conseguido importantes logros en su análisis. Uno de estos logros es la separación de esta clase en dos subclases. La primera contiene los problemas que pueden resolverse de forma eficiente, es decir, los problemas para los que se conocen algoritmos que resuelven cada instancia de forma óptima en tiempo polinómico. La segunda subclase contiene los problemas que son notoriamente difíciles, denominados formalmente NP-duros.

Para un algoritmo NP-duro se cree que no existe ningún algoritmo que resuelva cada instancia en tiempo polinómico. En consecuencia, hay instancias que requieren un tiempo súper polinomial o exponencial para ser resueltas de forma óptima. Está claro que los problemas difíciles deben tratarse en la práctica, lo cual puede hacerse mediante dos tipos de algoritmos: de optimización, que encuentran soluciones óptimas posiblemente utilizando grandes cantidades de tiempo de cómputo, o algoritmos heurísticos que encuentran soluciones aproximadas en pequeñas cantidades de tiempo de cálculo.

Los algoritmos de búsqueda local son del tipo heurísticos. Constituyen un enfoque general muy utilizado para los problemas de optimización combinatoria. Suelen ser instancias de varios esquemas de búsqueda general, pero todas tienen la misma característica de una función de vecindad subyacente, que se utiliza para guiar la búsqueda de una buena solución.

El recocido simulado, es uno de los algoritmos de búsqueda local más conocidos, ya que tiene un rendimiento bastante bueno y es ampliamente aplicable.

En la física de la materia condensada, el recocido simulado se conoce como un proceso térmico para obtener estados de baja energía de un sólido en un baño de calor. El proceso consiste en los siguientes dos pasos [1]:

- aumentar la temperatura del baño de calor hasta un valor máximo en el que el sólido se funde
- disminuir cuidadosamente la temperatura del baño de calor hasta que las partículas se organicen en el estado básico del sólido.

A continuación, se presentará una explicación del recocido simulado, sus características, aplicaciones e implementación

## 2. Desarrollo

**Asignatura:** Metaheurísticas

**Actividad No.17**

**Guía Taller Laboratorio No.8**

**Título:** Solución de problemas mediante Recocido Simulado

**Contenido:**

- Métodos heurísticos de solución de problemas.
- Recocido Simulado.

**Objetivo:** Implementar algoritmos de Recocido Simulado, en lenguajes de alto nivel, para la solución de problemas de competencia.

**1. Analice detalladamente las seis funciones definidas en el documento “Funciones de prueba.pdf”.**

Nombre de la función	Ref.	Fórmula	Punto mínimo	Valor mínimo
Alpine 1 Function	[1]	$f_1(x) = \sum_{i=1}^D  x_i \sin(x_i) + 0.1x_i $	$x^* = f(0,0)$	$f(x^*)=0$
Dixon & Price Function	[2]	$f_2(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2 \sin(x_i) - x_{i-1})^2$	$x^* = f(2(\frac{2^i-2}{2^i}))$	$f(x^*)=0$
Quintic Function	[3]	$f_3(x) = \sum_{i=1}^D  x_i^5 - 3x_i^4 + 4x_i^3 - 2x_i^2 - 10x_i - 4 $	$x^* = f(-1 \text{ or } 2)$	$f(x^*)=0$
Schwefel 2.23 Function	[4]	$f_4(x) = \sum_{i=1}^D x_i^{1.0}$	$x^* = f(0,0)$	$f(x^*)=0$
Stretched V Sine Wave Function	[5]	$f_5(x) = \sum_{i=1}^{D-1} (x_{i+1}^2 + x_i^2)^{0.25} \left[ \sin^2 \left\{ 50(x_{i+1}^2 + x_i^2)^{0.1} \right\} + 0.1 \right]$	$x^* = f(0,0)$	$f(x^*)=0$
Sum Squares Function	[6]	$f_6(x) = \sum_{i=1}^D ix_i^2$	$x^* = f(0,0)$	$f(x^*)=0$

**Tabla 1. Descripción de las 6 funciones**

**2. Implemente dichas funciones & 3. Implemente el algoritmo de Recocido Simulado para la solución de los problemas de minimización de las funciones anteriores. Considere D=10 y posteriormente D=30 dimensiones**

```
use rand::Rng;
use std::time::Instant;

const NUM_ELEMENTS: usize = 30;
const TEMP_MIN: f64 = 0.01;
const TEMP_MAX: f64 = 300.0;
const ALPHA: f64 = 0.8;
const K: f64 = 1.0;
```

```

fn alpine_cost(function: &[f64]) -> f64 {
    let mut valmax: f64 = 0.0;
    function.iter().for_each(|&x| {
        valmax += ((x * x.to_radians().sin()) + (0.1 * x)).abs(); // El sin
esta en Radianes
    });
    valmax
}

fn dixon_price_cost(function: &[f64]) -> f64 {
    let mut valmax: f64 = (function[0] - 1.0).powi(2);
    for i in 1..function.len() {
        valmax += (i as f64) * (2.0 * function[i].to_radians().sin() -
function[i - 1]).powi(2);
    }
    valmax
}

fn quintic_cost(function: &[f64]) -> f64 {
    let mut valmax: f64 = 0.0;
    function.iter().for_each(|&x| {
        valmax += (x.powi(5) - (3.0 * x.powi(4)) + (4.0 * x.powi(3))
- (2.0 * x.powi(2))
- (10.0 * x)
- 4.0)
        .abs();
    });
    valmax
}

fn schwefel_cost(function: &[f64]) -> f64 {
    let mut valmax: f64 = 0.0;
    function.iter().for_each(|&x| {
        valmax += x.powi(10);
    });
    valmax
}

fn stretched_v_sine_wave_cost(function: &[f64]) -> f64 {
    let mut valmax: f64 = 0.0;
    for i in 0..(function.len() - 1) {
        valmax += (function[i + 1].powi(2) + function[i].powi(2)).powf(0.25)
* ((50.0 * (function[i + 1].powi(2) +
function[i].powi(2)).powf(0.1))

```

```

        .to_radians()
        .sin()
        .powi(2)
        + 0.1);
    }
    valmax
}

fn sum_squares_cost(function: &[f64]) -> f64 {
    let mut valmax: f64 = 0.0;
    function.iter().enumerate().for_each(|(i, &x)| {
        valmax += (i as f64) * x.powi(2);
    });
    valmax
}

fn vecindad(function: &[f64]) -> Vec<f64> {
    let mut vecindad: Vec<f64> = function.to_vec();
    let locus = rand::thread_rng().gen_range(0..vecindad.len());
    vecindad[locus] = rand::thread_rng().gen_range(-10.0..=10.0);
    vecindad
}

fn simulated_annealing() -> String {
    let mut function_x: Vec<f64> = Vec::with_capacity(NUM_ELEMENTS);

    for _ in 0..NUM_ELEMENTS {
        function_x.push(rand::thread_rng().gen_range(-10.0..=10.0));
    }

    let mut edo_anterior: f64 = sum_squares_cost(&function_x); //
    *****

    let mut temp: f64 = TEMP_MAX;
    let vecinos: usize = function_x.len() - 1;
    let mut llamadas_costo: usize = 1;
    let mut valor_mas_bajo: f64 = edo_anterior;
    let mut iteracion_encontrada: usize = 0;
    let mut temperatura_encantrada: f64 = 0.0;

    while temp >= TEMP_MIN && llamadas_costo <= 500 {
        let mut vecinos_revisados: usize = 0;
        while vecinos_revisados < vecinos && llamadas_costo <= 500 {
            let sucesor: Vec<f64> = vecindad(&function_x);

```

```

        let edo_nuevo: f64 = sum_squares_cost(&sucesor); //
*****
        llamadas_costo += 1;
        let delta = edo_nuevo - edo_anterior;
        if delta > 0.0 {
            if rand::thread_rng().gen_range(0.0..1.0) < (-delta / (K *
temp)).exp() {
                edo_anterior = edo_nuevo;
                funcion_x = sucesor;
            }
        } else {
            edo_anterior = edo_nuevo;
            funcion_x = sucesor;
            if edo_nuevo < valor_mas_bajo {
                valor_mas_bajo = edo_nuevo;
                iteracion_encontrada = llamadas_costo;
                temperatura_encantrada = temp;
            }
        }
        vecinos_revisados += 1;
    }
    temp *= ALPHA;
}
let res: String = format!(
    "| {} | {} | {} | {} | {} | {} | {} |",
    edo_anterior,
    temp,
    valor_mas_bajo,
    temperatura_encantrada,
    iteracion_encontrada,
    edo_anterior == valor_mas_bajo
);
res
}

fn main() {
    println!("## D=30 Funcion 6: sum_squares_cost\n");
    println!("| Costo Final | Temp Min | Valor Mejor Solución | Temperatura
Mejor Solución | Iteración Mejor Solución | Se Conserva (?) | Tiempo |");
    println!("|-----|-----|-----|-----|
-----|-----|-----|-----|");
    for _ in 0..20 {
        let start: Instant = std::time::Instant::now();
        println!(
            "{} {} |",

```

```

        simulated_annealing(),
        start.elapsed().as_secs_f64()
    );
}
}

```

4. Reporte los resultados obtenidos. Para ello, realice 20 ejecuciones independientes, con la siguiente configuración:

a. Considere un total de 500 evaluaciones de la función objetivo.

b. Muestre el mejor, peor, promedio, mediana y desviación estándar de los resultados en las 20 ejecuciones.

Tabla de Costo para D = 10					
Función	Mejor	Peor	Promedio	Mediana	Desviación Estándar
F1	0.7036	0.2288	0.4563	0.4712	0.1332
F2	33.4467	2.6885	10.3252	9.7182	7.2672
F3	34.3277	3.3382	17.4788	16.8258	8.1133
F4	7.4888	0.0125	0.9877	0.3765	1.7927
F5	7.9450	2.9633	5.4107	5.1949	1.1162
F6	11.2354	1.9190	6.0890	6.1752	2.8386
<b>Promedios Globales</b>	<b>15.8579</b>	<b>1.8584</b>	<b>6.7913</b>	<b>6.4603</b>	<b>3.5435</b>

**Tabla 2. Costos con promedios globales D=10**

Tabla de Costo para D = 30					
Función	Mejor	Peor	Promedio	Mediana	Desviación Estándar
F1	25.1673	8.7683	16.3795	17.0539	4.3401
F2	718.3550	361.2193	495.7782	466.2917	109.7471
F3	615.5488	193.9260	333.4955	316.3965	121.3671
F4	37182.0678	127.5891	6605.7022	2724.2832	9721.9047
F5	93.0818	67.5016	78.4858	78.6814	6.0067
F6	700.5744	208.8296	494.3630	465.2775	131.4393
<b>Promedios Globales</b>	<b>6555.7992</b>	<b>161.3056</b>	<b>1337.3674</b>	<b>677.9974</b>	<b>1682.4675</b>

**Tabla 3. Costos con promedios globales D=30**

c. Muestre el mejor, peor, promedio, mediana y desviación estándar de los tiempos de ejecución (en segundos) en las 20 ejecuciones.

Tabla de Tiempos (segundos) para D = 10					
Función	Mejor	Peor	Promedio	Mediana	Desviación Estándar
F1	0.0041	0.0019	0.0022	0.0020	0.0005
F2	0.0046	0.0022	0.0025	0.0023	0.0006
F3	0.0042	0.0027	0.0032	0.0031	0.0004
F4	0.0033	0.0021	0.0024	0.0023	0.0003
F5	0.0048	0.0034	0.0038	0.0036	0.0004
F6	0.0037	0.0021	0.0024	0.0022	0.0004
<b>Promedios Globales</b>	<b>0.0041</b>	<b>0.0024</b>	<b>0.0028</b>	<b>0.0026</b>	<b>0.0004</b>

*Tabla 4. Tiempos con promedios globales D=10*

Tabla de Tiempos (segundos) para D = 30					
Función	Mejor	Peor	Promedio	Mediana	Desviación Estándar
F1	0.0040	0.0025	0.0030	0.0028	0.0005
F2	0.0049	0.0035	0.0039	0.0037	0.0004
F3	0.0077	0.0060	0.0065	0.0065	0.0004
F4	0.0059	0.0034	0.0038	0.0037	0.0005
F5	0.0103	0.0087	0.0095	0.0094	0.0005
F6	0.0049	0.0035	0.0039	0.0036	0.0004
<b>Promedios Globales</b>	<b>0.0063</b>	<b>0.0046</b>	<b>0.0051</b>	<b>0.0050</b>	<b>0.0005</b>

*Tabla 5. Tiempos con promedios globales D=30*

d. Muestre el mínimo, máximo, promedio, mediana y desviación estándar de la temperatura mínima alcanzada por el algoritmo en las 20 ejecuciones.

Tabla de Temperatura Mínima para D = 10					
Función	Mejor	Peor	Promedio	Mediana	Desviación Estándar
F1	0.0084	0.0084	0.0084	0.0084	0.0000
F2	0.0084	0.0084	0.0084	0.0084	0.0000
F3	0.0084	0.0084	0.0084	0.0084	0.0000
F4	0.0084	0.0084	0.0084	0.0084	0.0000
F5	0.0084	0.0084	0.0084	0.0084	0.0000
F6	0.0084	0.0084	0.0084	0.0084	0.0000
<b>Promedios Globales</b>	<b>0.0084</b>	<b>0.0084</b>	<b>0.0084</b>	<b>0.0084</b>	<b>0.0000</b>

*Tabla 6. Temperatura mínima con promedios globales D=10*



Tabla de Temperatura Mínima para D = 30					
Función	Mejor	Peor	Promedio	Mediana	Desviación Estándar
F1	5.4043	5.4043	5.4043	5.4043	0.0000
F2	5.4043	5.4043	5.4043	5.4043	0.0000
F3	5.4043	5.4043	5.4043	5.4043	0.0000
F4	5.4043	5.4043	5.4043	5.4043	0.0000
F5	5.4043	5.4043	5.4043	5.4043	0.0000
F6	5.4043	5.4043	5.4043	5.4043	0.0000
Promedios Globales	5.4043	5.4043	5.4043	5.4043	0.0000

**Tabla 7. Temperatura mínima con promedios globales D=30**

5. Reporte, además, de forma independiente, para cada función y cada ejecución: mejor solución encontrada por el algoritmo, la temperatura a la que se encontró, el número de evaluaciones de la función objetivo a la que se encontró, y si fue conservada o no por el algoritmo (fue conservada si coincide con la solución devuelta, NO fue conservada si el algoritmo la desechó durante su ejecución).

#### Tablas para D=10

Ejecución	F1				F2			
	Valor de la mejor solución	Temperatura	Evaluación	Conservada	Valor de la mejor solución	Temperatura	Evaluación	Conservada
1	0.3502	0.0105	419	false	2.6885	0.0105	423	true
2	0.4155	0.0105	417	true	6.8949	0.0255	381	true
3	0.2337	0.0131	415	true	6.2094	0.0163	402	true
4	0.2748	0.0319	379	false	12.0127	0.0105	417	true
5	0.5508	0.0163	398	false	3.8048	0.0163	403	true
6	0.5858	0.0105	419	true	9.5738	0.0163	402	true
7	0.4341	0.0105	419	true	21.9189	0.0105	419	true
8	0.2288	0.0105	420	true	3.9144	0.0255	381	true
9	0.3303	0.0105	417	true	4.4991	0.0163	406	true
10	0.5115	0.0105	423	true	3.2074	0.0105	421	true
11	0.4826	0.0105	424	true	12.9886	0.0105	421	true
12	0.4599	0.0105	424	true	14.8064	0.0131	414	true
13	0.5976	0.0105	422	true	12.2194	0.0399	368	false
14	0.7036	0.0131	411	true	33.4467	0.0163	404	true
15	0.3542	0.0319	377	false	10.5314	0.0974	334	true
16	0.4972	0.0131	411	false	14.2154	0.0105	424	true
17	0.3082	0.0105	424	true	9.8627	0.0204	394	true
18	0.5557	0.0105	422	true	8.4824	0.0204	389	true
19	0.6286	0.0105	424	true	9.9869	0.0204	396	true
20	0.4899	0.0319	374	false	5.2228	0.0779	343	true

**Tabla 8. Fragmento 1 de Resultados con 20 Ejecuciones para D=10**

	F3				F4			
Ejecución	Valor de la mejor solución	Temperatura	Evaluación	Conservada	Valor de la mejor solución	Temperatura	Evaluación	Conservada
1	12.4216	0.2377	295	true	0.0929	0.4642	263	false
2	13.1360	0.0319	374	true	0.0712	0.0105	424	true
3	16.5530	0.0105	422	true	0.0177	0.0105	419	true
4	13.0456	0.0105	419	true	0.0282	0.0163	400	true
5	29.5454	0.0131	411	true	3.7914	0.0399	366	false
6	11.8707	0.0623	352	true	0.8279	10.5553	137	false
7	24.9785	0.0204	392	true	0.2902	0.0204	390	false
8	21.8977	0.0131	412	true	0.4611	0.0623	352	false
9	19.9171	0.0163	403	true	0.2197	0.0163	402	false
10	17.1792	0.0131	414	true	0.7927	0.0319	375	true
11	23.4825	0.0255	384	true	2.3868	0.0131	412	false
12	3.3382	0.0163	401	true	0.3632	0.0105	418	false
13	17.0986	0.0498	360	true	0.0291	0.0163	405	true
14	27.2206	0.0163	402	true	0.0299	0.0105	419	false
15	7.0740	0.0105	420	true	0.4788	0.0163	406	true
16	24.7610	0.0498	356	true	0.0083	0.0255	383	false
17	34.3277	0.0105	420	true	0.2427	2.2136	204	false
18	8.5193	0.0131	415	true	7.4888	0.0105	423	true
19	12.4298	0.0319	377	true	0.2385	0.0255	387	false
20	10.7787	0.0105	423	true	0.7629	0.0163	402	true

**Tabla 9. Fragmento 2 de Resultados con 20 Ejecuciones para D=10**

	F5				F6			
Ejecución	Valor de la mejor solución	Temperatura	Evaluación	Conservada	Valor de la mejor solución	Temperatura	Evaluación	Conservada
1	6.1077	0.0131	414	true	5.6732	0.0105	424	true
2	5.5212	0.0105	418	true	8.4034	0.0163	399	true
3	5.2830	0.0105	418	true	3.2733	0.0204	393	true
4	4.5032	0.0399	370	true	3.1563	0.0163	400	true
5	5.0592	0.0105	420	true	1.9190	0.0105	424	true
6	6.8445	0.0131	413	true	8.3972	0.0498	354	true
7	4.5877	0.0105	422	true	3.6590	0.0163	402	true
8	5.1068	0.0131	415	true	3.1958	0.0163	400	true
9	5.4694	0.0255	383	true	7.2178	0.0131	408	true
10	4.2823	0.0131	407	true	11.2354	0.0204	389	true
11	4.7540	0.0319	374	true	5.9587	0.0255	388	true
12	2.9633	0.0163	403	true	10.1821	0.0204	397	true
13	5.0370	0.0255	383	true	2.8144	0.0105	420	true
14	7.9450	0.0163	401	true	6.6681	0.0105	421	true
15	6.0513	0.0399	366	true	7.0709	0.0779	335	false
16	4.5357	0.0399	367	true	10.5892	0.0163	404	true
17	6.8075	0.0163	406	true	4.5765	0.0319	374	true
18	4.7598	0.0319	374	true	3.0002	0.0163	406	true
19	6.3170	0.0105	419	true	8.3962	0.0105	420	true
20	6.2784	0.0204	396	true	6.3917	0.0498	358	true

**Tabla 10. Fragmento 3 de Resultados con 20 Ejecuciones para D=10**

### Tablas para D=30

	F1				F2			
Ejecución	Valor de la mejor solución	Temperatura	Evaluación	Conservada	Valor de la mejor solución	Temperatura	Evaluación	Conservada
1	11.9012	62.9146	220	false	351.5265	10.5553	464	false
2	12.8874	240.0000	57	false	712.9056	6.7554	498	true
3	10.0560	10.5553	441	false	406.4749	8.4442	477	false
4	8.2139	25.7698	348	false	618.3851	6.7554	496	true
5	9.8957	78.6432	198	false	514.0673	8.4442	492	true
6	8.7384	6.7554	500	false	383.3285	8.4442	475	false
7	7.1440	20.6158	364	false	458.8711	8.4442	493	false
8	10.0163	13.1941	425	false	718.3550	8.4442	484	true
9	9.1158	16.4927	400	false	636.7497	6.7554	498	true
10	10.5379	25.7698	329	false	380.1697	6.7554	498	true
11	7.3472	8.4442	467	false	523.3561	8.4442	494	true
12	11.3751	78.6432	192	false	406.1953	6.7554	499	true
13	10.2047	13.1941	408	false	374.7486	8.4442	494	false
14	12.4708	62.9146	212	false	573.2667	8.4442	491	true
15	8.5962	6.7554	498	false	472.8177	8.4442	492	true
16	12.3359	50.3316	257	false	506.8709	8.4442	489	true
17	9.3311	25.7698	349	false	553.2417	6.7554	499	false
18	10.7709	98.3040	152	false	428.6653	8.4442	494	true
19	9.8217	300.0000	26	false	438.2344	8.4442	481	false
20	8.8695	62.9146	225	false	417.3514	8.4442	490	true

**Tabla 11. Fragmento 1 de Resultados con 20 Ejecuciones para D=30**

	F3				F4			
Ejecución	Valor de la mejor solución	Temperatura	Evaluación	Conservada	Valor de la mejor solución	Temperatura	Evaluación	Conservada
1	312.2156	6.7554	498	true	3064.7301	6.7554	498	true
2	224.1925	8.4442	481	false	2752.6545	6.7554	499	true
3	168.8585	10.5553	438	false	127.5891	8.4442	491	false
4	613.8412	8.4442	493	true	21583.2861	6.7554	499	true
5	238.7284	8.4442	484	true	455.9425	8.4442	494	true
6	300.2145	6.7554	498	true	4567.4699	6.7554	500	true
7	324.1093	8.4442	492	false	2734.2573	6.7554	501	true
8	257.9690	8.4442	479	false	1604.6023	10.5553	465	false
9	446.5613	6.7554	500	true	1281.7483	8.4442	488	false
10	325.3715	10.5553	462	false	7533.5603	10.5553	455	false
11	203.9402	6.7554	499	true	417.9259	8.4442	488	false
12	225.3163	8.4442	466	false	37182.0678	8.4442	493	true
13	317.6329	10.5553	444	false	1201.3171	10.5553	453	false
14	310.7885	8.4442	494	true	21286.6832	6.7554	499	true
15	459.0751	8.4442	480	false	2714.3091	8.4442	492	true
16	341.5495	8.4442	487	true	202.8781	10.5553	444	false
17	206.1185	6.7554	498	true	772.4000	8.4442	478	false
18	314.5914	10.5553	459	false	10636.9872	8.4442	480	false
19	594.6420	8.4442	481	false	11723.9941	8.4442	470	false
20	337.7146	8.4442	488	false	224.7272	8.4442	489	false

**Tabla 12. Fragmento 2 de Resultados con 20 Ejecuciones para D=30**

	F5				F6			
Ejecución	Valor de la mejor solución	Temperatura	Evaluación	Conservada	Valor de la mejor solución	Temperatura	Evaluación	Conservada
1	61.4023	8.4442	477	false	366.2522	10.5553	455	false
2	65.6354	20.6158	378	false	408.4069	8.4442	467	false
3	64.9825	10.5553	440	false	402.3012	8.4442	471	false
4	67.8092	10.5553	437	false	449.4508	6.7554	499	true
5	62.5648	122.8800	135	false	643.7001	8.4442	488	true
6	70.5465	153.6000	90	false	693.2707	6.7554	499	false
7	62.7020	10.5553	453	false	552.9158	8.4442	494	false
8	69.6962	98.3040	160	false	547.2150	8.4442	477	true
9	65.2431	240.0000	56	false	658.8074	8.4442	469	false
10	63.0457	122.8800	123	false	596.7631	8.4442	485	true
11	63.0003	153.6000	99	false	481.1042	8.4442	492	true
12	59.0292	10.5553	455	false	298.2957	6.7554	496	true
13	67.7243	122.8800	138	false	516.4709	8.4442	489	true
14	60.0006	98.3040	171	false	632.8962	8.4442	489	true
15	65.1608	78.6432	201	false	669.5333	8.4442	493	true
16	69.5088	78.6432	203	false	448.7239	8.4442	484	true
17	63.9920	153.6000	115	false	396.8502	8.4442	492	true
18	66.4387	40.2653	280	false	208.8296	6.7554	498	true
19	71.3014	16.4927	395	false	437.4917	8.4442	487	false
20	63.9633	78.6432	181	false	419.8599	8.4442	491	false

**Tabla 13. Fragmento 3 de Resultados con 20 Ejecuciones para D=30**

### 3. Conclusiones

Al realizar la corrida manual del algoritmo SA, se pudo apreciar que los algoritmos de búsqueda local pueden ser muy útiles si estamos interesados en el estado de la solución, pero no en el camino hacia ese objetivo. Estos algoritmos operan sólo en el estado actual y se mueven a estados vecinos. Al permitir un ascenso ocasional en el proceso de búsqueda, se puede escapar de la trampa de los mínimos locales, pero también existe la posibilidad de pasar óptimos globales después de alcanzarlos.

Se puede aplicar el SA para generar una solución a los problemas de optimización combinatoria asumiendo una analogía entre ellos y los sistemas físicos de muchas partículas con las siguientes equivalencias:

- Las soluciones del problema son equivalentes a los estados de un sistema físico.
- El costo de una solución es equivalente a la "energía" de un estado

Por último, es importante mencionar que los algoritmos de búsqueda local tienen dos ventajas fundamentales: usan muy poca memoria y pueden encontrar soluciones razonables en espacios de estados grandes o infinitos (continuos).

### 4. Referencias

[1] Burke & Kendall. Search Methodologies – 2005. Capítulo 7

## 5. Apéndice

- **Resultados Costos:**

F1	F2	F3	F4	F5	F6
0.3511	2.6885	12.4216	0.3897	6.1077	5.6732
0.4155	6.8949	13.1360	0.0712	5.5212	8.4034
0.2337	6.2094	16.5530	0.0177	5.2830	3.2733
0.3282	12.0127	13.0456	0.0282	4.5032	3.1563
0.5799	3.8048	29.5454	3.7949	5.0592	1.9190
0.5858	9.5738	11.8707	1.4049	6.8445	8.3972
0.4341	21.9189	24.9785	0.2902	4.5877	3.6590
0.2288	3.9144	21.8977	0.4611	5.1068	3.1958
0.3303	4.4991	19.9171	0.2197	5.4694	7.2178
0.5115	3.2074	17.1792	0.7927	4.2823	11.2354
0.4826	12.9886	23.4825	2.4052	4.7540	5.9587
0.4599	14.8064	3.3382	0.3632	2.9633	10.1821
0.5976	12.2374	17.0986	0.0291	5.0370	2.8144
0.7036	33.4467	27.2206	0.0616	7.9450	6.6681
0.3764	10.5314	7.0740	0.4788	6.0513	7.0718
0.5003	14.2154	24.7610	0.0125	4.5357	10.5892
0.3082	9.8627	34.3277	0.4384	6.8075	4.5765
0.5557	8.4824	8.5193	7.4888	4.7598	3.0002
0.6286	9.9869	12.4298	0.2441	6.3170	8.3962
0.5136	5.2228	10.7787	0.7629	6.2784	6.3917

**Tabla 14. Resultados costo 6 funciones en las 20 ejecuciones D=10**

F1	F2	F3	F4	F5	F6
21.0001	361.2193	312.2156	3064.7301	81.3732	377.4662
17.2018	712.9056	225.7750	2752.6545	72.6392	414.0227
13.3384	419.7718	193.9260	127.5891	75.1015	417.8675
19.6462	618.3851	613.8412	21583.2861	81.2067	449.4508
13.2190	514.0673	238.7284	455.9425	77.6685	643.7001
8.7683	386.2046	300.2145	4567.4699	81.5333	700.5744
19.3134	459.7656	334.8990	2734.2573	72.6043	556.8773
19.8356	718.3550	267.7859	1604.7807	89.8460	547.2150
17.5915	636.7497	446.5613	1282.0924	81.2356	663.1991
12.0917	380.1697	332.6962	7534.3515	78.0702	596.7631
10.0787	523.3561	203.9402	438.5446	73.9477	481.1042
14.7197	406.1953	231.4074	37182.0678	70.5556	298.2957
16.2135	375.2953	353.2781	1201.3454	80.3043	516.4709
19.0803	573.2667	310.7885	21286.6832	67.5016	632.8962
10.1713	472.8177	467.9428	2714.3091	76.8660	669.5333
18.2203	506.8709	341.5495	214.5391	75.5685	448.7239
13.2807	555.3542	206.1185	774.0795	93.0818	396.8502
16.9060	428.6653	320.5774	10637.3377	81.4006	208.8296
25.1673	448.7972	615.5488	11732.9795	79.9184	445.5971
21.7461	417.3514	352.1155	225.0048	79.2927	421.8232

**Tabla 15. Resultados costo 6 funciones en las 20 ejecuciones D=30**

- **Resultados Tiempos:**

F1	F2	F3	F4	F5	F6
0.0041	0.0046	0.0042	0.0033	0.0048	0.0037
0.0025	0.0023	0.0034	0.0023	0.0036	0.0023
0.0019	0.0025	0.0029	0.0023	0.0034	0.0022
0.0020	0.0022	0.0030	0.0022	0.0036	0.0021
0.0020	0.0032	0.0028	0.0026	0.0037	0.0027
0.0021	0.0024	0.0034	0.0023	0.0042	0.0025
0.0027	0.0022	0.0034	0.0023	0.0037	0.0023
0.0029	0.0022	0.0028	0.0024	0.0035	0.0022
0.0021	0.0028	0.0028	0.0022	0.0040	0.0022
0.0019	0.0025	0.0038	0.0025	0.0040	0.0022
0.0020	0.0024	0.0027	0.0029	0.0035	0.0021
0.0021	0.0023	0.0034	0.0023	0.0035	0.0034
0.0020	0.0025	0.0031	0.0021	0.0037	0.0025
0.0019	0.0022	0.0031	0.0022	0.0042	0.0022
0.0019	0.0025	0.0029	0.0023	0.0034	0.0022
0.0023	0.0023	0.0031	0.0024	0.0034	0.0023
0.0022	0.0029	0.0031	0.0028	0.0036	0.0022
0.0019	0.0022	0.0029	0.0024	0.0044	0.0032
0.0019	0.0022	0.0035	0.0024	0.0037	0.0022
0.0020	0.0022	0.0028	0.0021	0.0034	0.0022

**Tabla 16. Resultados tiempo 6 funciones en las 20 ejecuciones D=10**

F1	F2	F3	F4	F5	F6
0.0036	0.0049	0.0071	0.0059	0.0103	0.0049
0.0033	0.0037	0.0062	0.0039	0.0094	0.0035
0.0027	0.0041	0.0065	0.0039	0.0094	0.0035
0.0025	0.0037	0.0061	0.0042	0.0094	0.0038
0.0025	0.0038	0.0068	0.0037	0.0091	0.0040
0.0031	0.0036	0.0077	0.0037	0.0097	0.0036
0.0029	0.0044	0.0062	0.0035	0.0091	0.0036
0.0026	0.0037	0.0069	0.0035	0.0097	0.0038
0.0025	0.0041	0.0063	0.0042	0.0098	0.0042
0.0025	0.0039	0.0065	0.0035	0.0087	0.0035
0.0036	0.0037	0.0060	0.0036	0.0102	0.0035
0.0038	0.0036	0.0063	0.0036	0.0088	0.0047
0.0034	0.0041	0.0066	0.0041	0.0094	0.0036
0.0030	0.0043	0.0065	0.0036	0.0097	0.0036
0.0025	0.0036	0.0063	0.0034	0.0094	0.0036
0.0039	0.0035	0.0068	0.0037	0.0101	0.0045
0.0040	0.0040	0.0060	0.0040	0.0087	0.0037
0.0027	0.0042	0.0069	0.0037	0.0098	0.0035
0.0025	0.0036	0.0060	0.0034	0.0100	0.0036
0.0025	0.0035	0.0065	0.0036	0.0089	0.0044

**Tabla 17. Resultados tiempo 6 funciones en las 20 ejecuciones D=30**

- **Resultados Temperaturas mínimas:**

F1	F2	F3	F4	F5	F6
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084
0.0084	0.0084	0.0084	0.0084	0.0084	0.0084

**Tabla 18. Resultados temperatura mínima 6 funciones en las 20 ejecuciones D=10**

F1	F2	F3	F4	F5	F6
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043
5.4043	5.4043	5.4043	5.4043	5.4043	5.4043

**Tabla 19. Resultados temperatura mínima 6 funciones en las 20 ejecuciones D=30**