



**Instituto Politecnico Nacional
Escuela Superior De Cómputo**



Reporte de resultados práctica 1
**“Introducción a la optimización en espacios continuos y
metaheurísticas de búsqueda local.”**

Clase de la profesora Dra. Miriam Pescador Rojas
de la materia de Algoritmos bioinspirados
ESCOM-IPN

Alumnos
Fuentes López Eva Melisa
Orozco Solorio Kevin Adrián

Fecha de entrega: 26 septiembre 2024

Introducción.

En ingeniería, el área de optimización es fundamental para mejorar el rendimiento, reducir costos, aumentar la eficiencia y cumplir con las restricciones de diseño y operación.

Definición de un problema de optimización.

Un problema de optimización busca la mejor solución factible para maximizar o minimizar una función, sujeta a ciertas restricciones. Por ejemplo, es posible maximizar el beneficio de una empresa eligiendo una cantidad adecuada de productos para producir, o minimizar el costo de construcción de una estructura eligiendo adecuadamente los materiales y las dimensiones.

Matemáticamente, el problema de optimización se define de la siguiente manera:

$$\begin{array}{ll} \text{Encontrar} & [Equation] \text{ tal que} \\ \\ \text{maximice (o minimice)} & [Equation] \quad (1) \\ \\ \text{sujeto a} & [Equation] \\ & [Equation] \end{array}$$

Donde:

- $f(x)$ es la función objetivo que se desea maximizar o minimizar.
- x es el vector de variables de decisión.
- $g_i(x)$ son las funciones de restricción de desigualdad.
- $h_i(x)$ son las funciones de restricción de igualdad.
- m es el número de restricciones de desigualdad.
- p es el número de restricciones de igualdad.

Búsqueda global y local

En el proceso de optimización, debe diferenciarse los conceptos de óptimo global y óptimos locales basándose en la definición de un vecindario. Es decir, un óptimo local es un punto en el cual la función objetivo toma un valor extremo (máximo o mínimo) en comparación con puntos cercanos en su *vecindad*. No necesariamente es el valor extremo más alto o más bajo en todo el dominio de la función, sino solo en una determinada región. Pueden existir varios óptimos locales para una función. Por otro lado, un óptimo global, es el punto en el cual la función objetivo toma su valor extremo (máximo o mínimo) en todo su dominio. Se trata del *mejor valor* posible de la función. Dependiendo de la función, puede existir un único óptimo global o múltiples puntos que alcancen ese valor extremo global (*multimodalidad*).

Una de las dificultades en un problema de optimización, especialmente cuando se usan técnicas heurísticas o métodos de búsqueda local, es la posibilidad de quedar *atrapado* en un óptimo local sin alcanzar el óptimo global.

En esta práctica estudiaremos, los métodos de búsqueda para funciones de una variable, los cuales requieren de dos fases:

1. **Fase de acotamiento.** Se realiza una búsqueda inicial de grano grueso para acotar el óptimo.
2. **Fase de refinamiento del intervalo.** Se realiza una secuencia finita de reducciones de intervalo o refinamientos para reducir el intervalo inicial de búsqueda a una cierta precisión deseable.

El término local utilizado frecuentemente en las metaheurísticas de búsqueda se asocia al uso de

estructuras de entorno, reflejando el concepto de proximidad o vecindad entre las soluciones alternativas de un problema.

Las soluciones creadas, en un determinado entorno, por un operador o mecanismo de generación se denominan soluciones vecinas. Un mecanismo de búsqueda local se usa para hacer eficiente el refinamiento de las soluciones. Una vez que se ha ubicado una zona factible, se recomienda utilizar este mecanismo para que agilice la convergencia al óptimo global. El Buscador local debe considerar lo siguiente:

- Establecer una dirección de búsqueda.
- Realizar movimientos que mejoren la solución inicial.
- En problemas de optimización continua, los movimientos deben generar soluciones con una

alta precisión.

Algunas posibles soluciones:

- Permitir movimientos de empeoramiento de la solución actual. p. ej. Recocido simulado y búsqueda tabú' (que veremos más adelante).
- Modificar la estructura de entornos. p. ej. búsqueda descendente basada en entornos variables.
- Inicializar la búsqueda desde otra solución, p. ej. búsqueda multiarranque.

Objetivo.

El objetivo de esta práctica es que el alumno entienda el funcionamiento de algoritmos heurísticos para la exploración y búsqueda de soluciones óptimas en problemas de optimización continua.

Objetivos particulares

Para cumplir con el objetivo general se plantean los siguientes objetivos particulares.

- Analizar el funcionamiento de algoritmos no deterministas como son las metaheurísticas.
- Conocer las características que hacen complejo un problema de optimización ya sea por ser no diferenciables, su alta dimensionalidad o por ser multimodales.
- Comparar el desempeño de dos algoritmos de búsqueda local en problemas de optimización continua con diferentes características.

Funciones de prueba

Los algoritmos se probarán con tres funciones de prueba clásicas, cada una con 2 variables de decisión (x_1, x_2):

- **Ackley.** La función de Ackley es ampliamente usada para probar algoritmos de optimización. El espacio definido por esta función se caracteriza por una región externa casi plana y un gran agujero en el centro. La función está definida como:

[Equation] (2)

Unable to load the textbox

Donde: $a = 20$, $b = 0.2$, $c = 2\pi$, d es la dimensionalidad del problema, para esta práctica $d = 2$

El dominio de las variables de decisión es: $x_i \in [-32.768, 32.768]$, para toda i

$= 1, \dots, d$. El mínimo global se encuentra en $f(\vec{x}^*) = 0$ en $\vec{x}^* = (0, 0, \dots, 0)$.

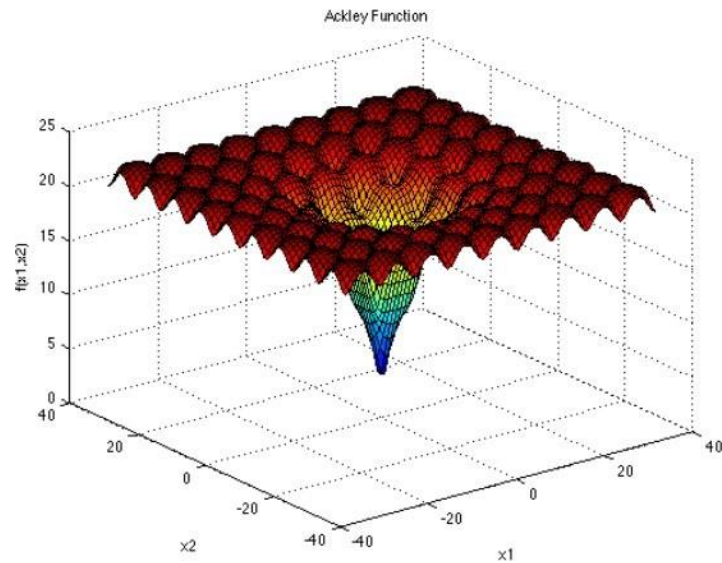


Figure 1: Función de Ackley

- **Rastrigin.** La función de Rastrigin tiene muchos mínimos locales, lo que significa que es altamente multimodal, sin embargo, estos están igualmente distribuidos. La figura muestra el espacio de búsqueda de esta función, para dos variables de decisión.

Matemáticamente la función es:

Unable to load the textbox

Unable to load the textbox

$$f(\vec{x}) = 10d + \sum_{i=1}^d x_i^2 - 10\cos(2\pi x_i)$$

$$(3)$$

$$i=1$$

Donde: $x_i \in [-5.12, 5.12]$, para todo $i = 1, \dots, d$.

El mínimo global se encuentra en $f(\vec{x}^*) = 0$ en $\vec{x}^* = (0, 0, \dots, 0)$.

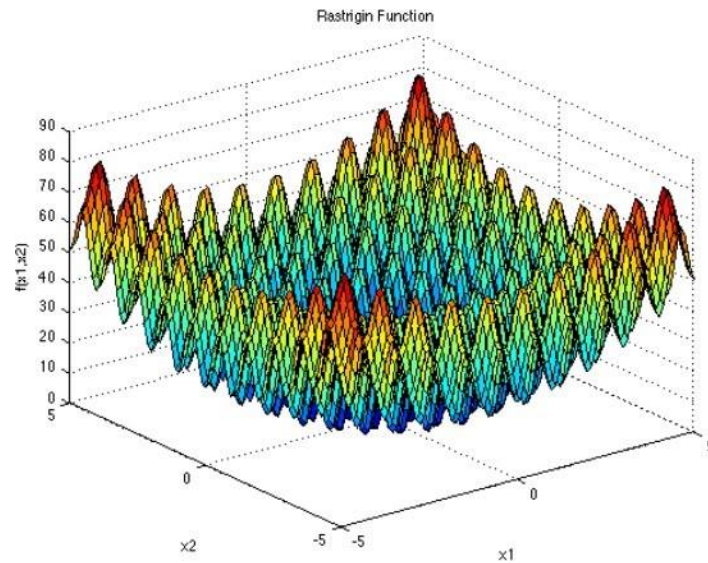


Figure 2: Función de Rastrigin

- **Schwefel.** La función de Schwefel también es multimodal, a diferencia de la función anterior los límites del espacio de búsqueda tienen una mayor amplitud. Ver figura. La figura muestra el espacio de búsqueda de esta función, para dos variables de decisión.

Matemáticamente la función es:

$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^d x_i \sin(\sqrt{|x_i|}) \quad (4)$$

Donde: $x_i \in [-500.0, 500.0]$, para todo $i = 1, \dots, d$.

El mínimo global se encuentra en $f(\vec{x}^*) = 0$ en $\vec{x}^* = (420.9687, 420.9687, \dots, 420.9687)$.

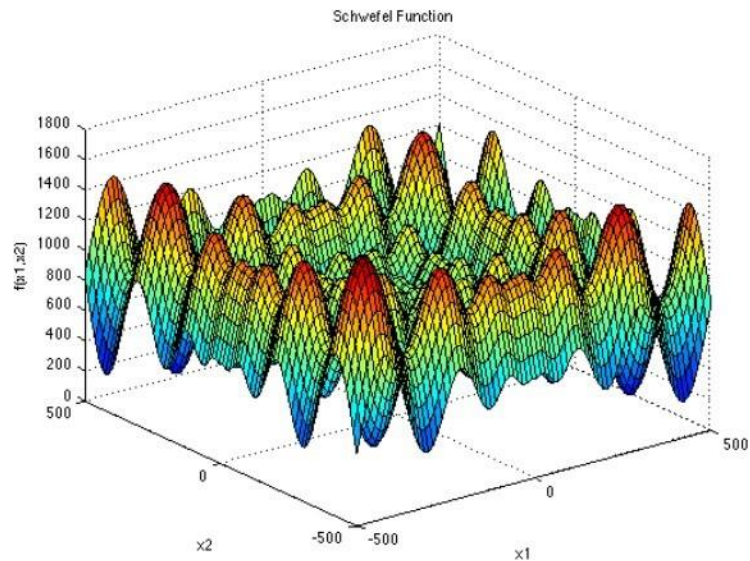


Figure 3: Función de Schwefel

Implementación de algoritmos de búsqueda heurísticas.

En esta práctica se solicita la implementación de dos estrategias de búsqueda heurística:

- **Hill climbing.** El algoritmo hill climbing, también llamado algoritmo de Escalada Simple o ascenso de colinas, es una técnica de optimización matemática que pertenece a la familia de los algoritmos de búsqueda local. Es un algoritmo iterativo que comienza con una solución arbitraria a un problema, luego intenta encontrar una mejor solución variando incrementalmente un único elemento de la solución. Si el cambio produce una mejor solución, otro cambio incremental se le realiza a la nueva solución, repitiendo este proceso hasta que no se puedan encontrar mejoras. Suele llamarse a esta búsqueda algoritmo voraz local, porque toma un estado vecino “bueno” sin pensar en la próxima acción.

El algoritmo 1 muestra el funcionamiento de esta técnica de búsqueda local.

Algorithm 1: Algoritmo de Hill Climbing.

Unable to load the graphic

Input: S_0 : solución inicial

maxT : número máximo de iteraciones

Output: S_{act} : mejor solución visitada

```
1  begin
2     $S_0 \leftarrow$  genera solución inicial
3     $\text{cont} \leftarrow 1$ 
4    while  $\text{cont} \geq \text{maxT}$  do
5       $\text{Scand} \leftarrow$  Genera vecino aleatorio ( $S_0$ )
6      if  $f(\text{Scand}) < f(S_0)$  then
7         $S_0 \leftarrow \text{Scand}$ 
8       $\text{cont}++$ 
9  Devuelve la mejor solución  $S$  visitada
```

• Simulated annealing

El algoritmo de recocido simulado fue propuesto por Kirkpatrick, Gellat y Vechi en 1983. Esta técnica se inspira en el proceso de recocido del acero y cerámicas. La técnica consiste en calentar y luego enfriar lentamente el material para variar sus propiedades físicas. Esta técnica requiere principalmente de 2 etapas:

1. Calentamiento: causa que los átomos aumenten su energía y que puedan así desplazarse de sus posiciones iniciales (un mínimo local de energía);
2. Enfriamiento: permite cristalizar en configuraciones con menor energía que la inicial (mínimo global) El algoritmo 2 muestra el funcionamiento de la estrategia de recocido simulado.

Algorithm 2: Algoritmo de recocido simulado.**Input:** T_0 : temperatura inicial α : función para decrementar la temperaturaL: tiempo de cada nivel de temperatura (nota: para esta práctica $L = 20$) T_f : temperatura final**Output:** S_{act} : mejor solución visitada

```

1  begin
2   $T \leftarrow T_0$ 
3   $S_{act} \leftarrow \text{genera solución inicial}$  while  $T \geq T_f$  do
4    for  $count \leftarrow 1$  to  $L(T)$  do
5       $Scand \leftarrow \text{Genera vecino aleatorio}(S_{act})$ 
6       $\delta \leftarrow \text{cost}(Scand) - \text{cost}(S_{act})$ 
7      if  $U(0, 1) < e^{-\delta/T}$  or  $(\delta < 0)$  then
8         $S_{act} \leftarrow Scand$ 
9       $T \leftarrow \alpha(T)$ 
10 Devuelve la mejor solución  $S_{act}$  visitado

```

Denotation	General functional form	Iterative formula
Linear	$T_k = T_0 - k\beta$	$T_k = T_{k-1} - \beta$
Geometric	$T_k = \alpha^k T_0$	$T_k = \alpha T_{k-1}$
Logarithmic	$T_k = \frac{T_0}{\ln(k+1)}$	$T_k = \frac{\ln(k)}{\ln(k+1)} T_{k-1}$
Hybrid	$T_k = \begin{cases} \frac{T_0}{1+k} & \text{if } k \leq \beta \\ \alpha^k T_0 & \text{if } k > \beta \end{cases}$	$T_k = \begin{cases} \frac{k}{k+1} T_{k-1} & \text{if } k \leq \beta \\ \alpha T_{k-1} & \text{if } k > \beta \end{cases}$
Exponential	$T_k = \frac{T_0}{1 + k\beta T_0}$	$T_k = \frac{T_{k-1}}{1 + \beta T_{k-1}}$

Figure 4: Tabla de funciones de enfriamiento para el algoritmo de recocido simulado

Desarrollo de la metodología experimental.

Con la finalidad de observar el efecto estocástico de los algoritmos implementados se realizó 2 ejecuciones con diferente semilla para generación de números aleatorios. El número máximo de iteraciones del algoritmo fue de 1,000 y se reporta el valor mínimo encontrado cada 200 iteraciones.

Para la implementación del algoritmo de recocido simulado se utilizaron la función lineal, la función geométrica y la función exponencial, para poder contrastar los resultados, ya que la función lineal disminuiría más lentamente la temperatura a diferencia de la función exponencial.

A continuación, se muestran los resultados obtenidos en la siguiente tabla comparativa:

Table 1: Tabla que muestra los resultados de prueba

Función	semilla	# iteraciones	Hill climbing		SA-1 (lineal)		SA-2 (Geometrica)		SA-3 (exponencial)	
			x	f(x)	x	f(x)	x	f(x)	x	f(x)
Ackley	571	200	-6.082106, -3.882123	13.275077	-5.40276	13.722146	-5.468327, -3.268344	14.016215	-6.091231, -3.891248	13.279232
		400	-6.081594, -3.881610	13.275017	-4.64557	12.739237	-1.743332, 0.456650	6.612805	-6.085679, -3.885696	13.276004
		600	-6.081594, -3.881610	13.275017	-5.85928	14.140803	-1.914581, 0.285401	6.131608	-6.083612, -3.883629	13.275358
		800	-4.994898, -2.794915	11.921682	0.105769	7.039749	5.300716, 7.500699	16.743099	-5.311599, -3.111616	13.130872
		1000	-1.094870, 1.105112	4.423088	2.117421	11.407656	4.769487, 4.769487	13.883638	-1.089498, 1.110484	4.426214
	2122	200	8.995824, 3.995042	15.029645	10, 5.626186	17.6101	9.745972, 5.348181	17.827016	9.019457, 4.018675	15.069825
		400	8.991315, 3.990533	15.028756	8.1892148, 4.679148	16.477595	4.919592, 0.521801	11.839123	8.995903, 3.995120	15.02968
		600	8.992102, 3.991320	15.028755	7.3603874, 7.315424	17.51381	8.179338, 3.781547	15.758337	8.996287, 3.995505	15.02986
		800	8.992102, 3.991320	15.028755	-1.491197, -1.536160	7.569966	-0.166359, -4.564150	11.423121	8.985707, 3.984925	15.030667
		1000	3.049890, - 1.950892	8.141566	2.5256527, 2.480689	10.223969	-7.920378, -7.920378	16.210775	3.049844, - 1.950938	8.141313
Rastrigin	571	200	-17.944742, -10.735837	448.757369	-13.407192, -6.198287	243.326663	- 18.024229, -10.815324	447.96957 9	- 14.943546, -7.734641	294.72029 5
		400	-3.117777, 4.091128	30.668192	-9.232645, -2.023740	98.360049	- 11.449306, -4.240401	177.96189 8	-4.085005, 3.123899	30.718986
		600	-4.100228, 3.108676	30.636156	-15.894358, -8.685452	344.136294	- 14.093106, -6.884201	250.20294 9	-3.109105, 4.099799	30.636497
		800	-4.100228, 3.108676	30.636156	20.263192, 27.472097	1195.98784 7	- 21.002929, -13.794024	638.66886 1	-3.109105, 4.099799	30.636497
		1000	-4.100228, 3.108676	30.636156	4.278773, 4.278773	60.211965	-3.298860, 3.910044	40.748141	-4.099655, 3.109249	30.636637
	2122	200	15.975909, -0.410653	273.977875	29.002188, 12.615625	1017.75696 8	32.529363, 17.540007	1405.3273 5	16.646006, 0.259442	303.82878 6
		400	8.176110, - 8.210452	147.323366	19.037528, 2.650965	385.560514	24.799539, 9.810183	724.50225 2	8.210178, - 8.176384	147.32206 3

Schwefel		600	8.178689, - 8.207873	147.311938	11.848378, -4.538184	184.897631	25.007300, 15.840483	880.91261 7	8.188623, - 8.197939	147.28523 7
		800	8.193688, - 8.192875	147.282229	-8.769284, -25.155847	722.931604	11.527894, 2.361078	174.74013 3	8.188623, - 8.197939	147.28523 7
		1000	8.193688, - 8.192875	147.282229	-32.768, - 32.768	2165.22652	-1.053902, -1.053902	3.357545	8.188623, - 8.197939	147.28523 7
	571	200	-1.145550, -0.019159	838.973488	-2.669783, -1.543392	842.091143	-2.949801, -1.823409	842.66319 3	-0.965366, 0.1610254	838.70599 6
		400	5.12, 5.12	830.080935	-1.357917, -0.231526	839.320752	-1.637935, -0.511543	839.87034 5	5.005147, 5.0051475	830.09732 4
		600	5.12, 5.12	830.080935	-1.209320, -0.082929	839.066938	-1.489338, -0.362946	839.57030 1	5.12, 5.12	830.08093 5
		800	5.12, 5.12	830.080935	-1.126656, -0.000265	838.949441	-1.406673, -0.280282	839.41117 9	5.12, 5.12	830.08093 5
		1000	5.12, 5.12	830.080935	0.684317, 1.810708	835.6971	0.404300, 1.530691	836.27939 1	5.12, 5.12	830.08093 5
	2122	200	5.12, 4.803221	830.116444	4.872441, 2.938695	831.139157	4.872441, 2.794747	831.26714 3	5.086475, 4.151435	830.31775 6
		400	5.12, 5.12	830.080935	3.962802, 3.371636	831.093635	3.962802, 3.227688	831.20145	5.12, 5.12	830.08093 5
		600	5.12, 5.12	830.080935	3.254666, 2.663501	832.140775	3.254666, 2.519552	832.28007 6	5.1057369 7, 5.1057369	830.08239 1
		800	5.12, 5.12	830.080935	2.180711, 1.589545	834.280965	2.180711, 1.445597	834.44616 1	5.12, 5.12	830.08093 5
		1000	5.12, 5.12	830.080935	-2.405935, -2.997101	843.329889	-2.405935, -3.141049	843.44876 4	5.12, 5.12	830.08093 5

Estas son algunas gráficas de dichos resultados:

```
[9.50274993 4.50196765]
X : [8.995824485223736, 3.9950422054236943] FUNCTION VALUE ==> 15.029645991593886
X : [8.991315405245716, 3.9905331254456744] FUNCTION VALUE ==> 15.028756530820228
X : [8.992102489044326, 3.991320209244284] FUNCTION VALUE ==> 15.028755504235882
X : [8.992102489044326, 3.991320209244284] FUNCTION VALUE ==> 15.028755504235882
X : [3.0498901436334003, -1.9508921361666416] FUNCTION VALUE ==> 8.141566481272777
```

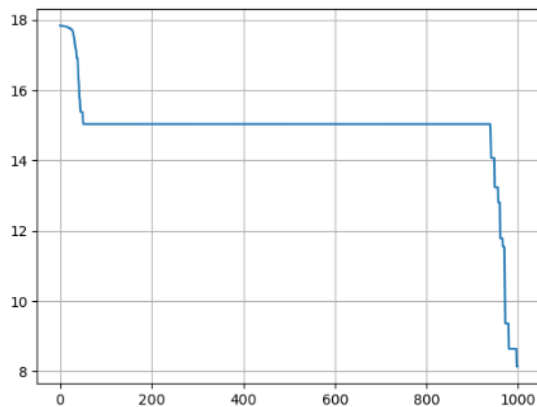


Figure 5. Hill climbing con función Ackley semilla fijada en 2122

```
[31.13861096 14.75204758]
X : [15.975909945901908, -0.4106534285468495] FUNCTION VALUE ==> 273.97787599525094
X : [8.176110536257031, -8.210452838191724] FUNCTION VALUE ==> 147.32336606237982
X : [8.178689652448316, -8.20787372200044] FUNCTION VALUE ==> 147.31193802235836
X : [8.193688300757412, -8.192875073691344] FUNCTION VALUE ==> 147.28222913648202
X : [8.193688300757412, -8.192875073691344] FUNCTION VALUE ==> 147.28222913648202
```

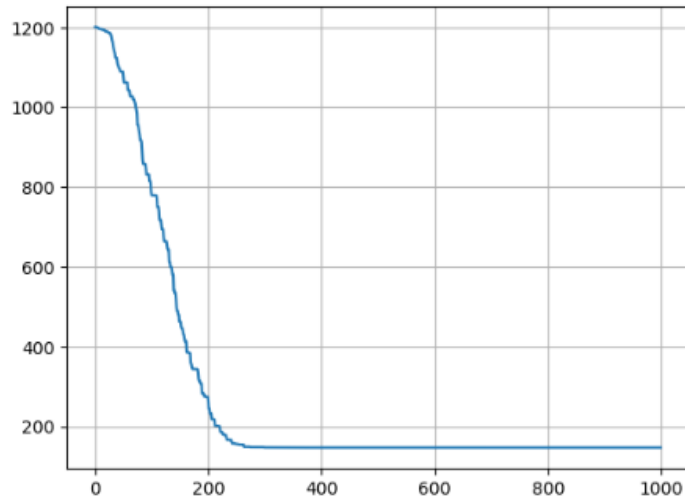


Figure 7. Hill climbing con función Rasrigin semilla fijada en 2122

```
[4.86540796 2.30500744]
X : [5.12, 4.803221963788928] FUNCTION VALUE ==> 830.1164442990289
X : [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197
X : [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197
X : [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197
X : [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197
```

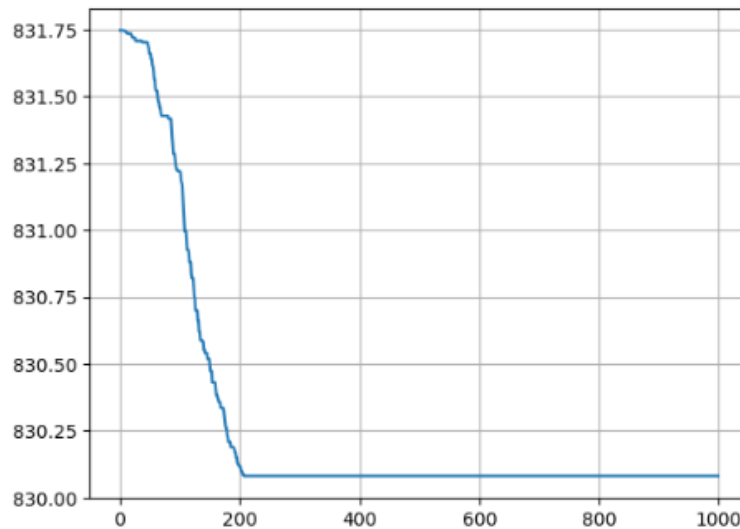


Figure 8. Hill climbing con función Schwefel con semilla fijada en 2122

```

[-21.70057062 -14.49166543]
X ( 200 ): [-13.40719227759542, -6.198287090634368] FUNCTION VALUE ==> 243.32666368156086 Temp: -17.5
X ( 400 ): [-9.232645422606042, -2.0237402356450085] FUNCTION VALUE ==> 98.36004919763197 Temp: -67.5
X ( 600 ): [-15.894358031192404, -8.68545284423137] FUNCTION VALUE ==> 344.1362946497104 Temp: -117.5
X ( 800 ): [20.26319279686514, 27.472097983826163] FUNCTION VALUE ==> 1195.9878478757814 Temp: -167.5
X ( 1000 ): [4.278773788108931, 4.278773788108931] FUNCTION VALUE ==> 60.21196585322262 Temp: -217.5

```

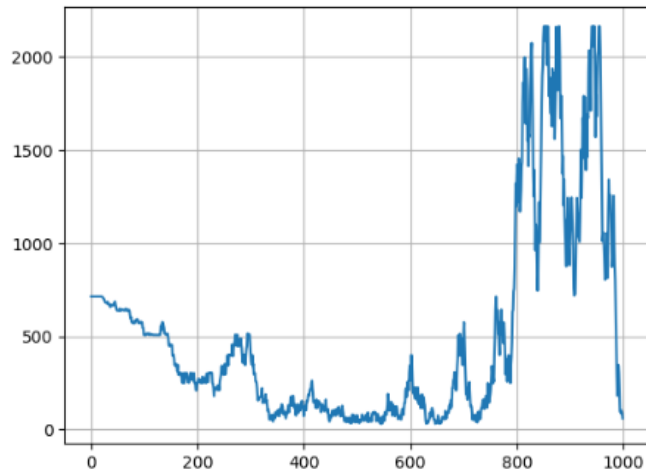


Figure 9. SA Ackley con función lineal y semilla fijada en 571

```

[-3.39071416 -2.26432272]
X ( 200 ): [-2.6697836961496773, -1.5433922606870096] FUNCTION VALUE ==> 842.091143292553 Temp: -17.5
X ( 400 ): [-1.3579177855525333, -0.23152635008986283] FUNCTION VALUE ==> 839.3207529599681 Temp: -67.5
X ( 600 ): [-1.2093207574364193, -0.08292932197374911] FUNCTION VALUE ==> 839.0669389411587 Temp: -117.5
X ( 800 ): [-1.126656556195975, -0.00026512073330575703] FUNCTION VALUE ==> 838.9494418952576 Temp: -167.5
X ( 1000 ): [0.6843174869751751, 1.8107089224378459] FUNCTION VALUE ==> 835.6971001261842 Temp: -217.5

```

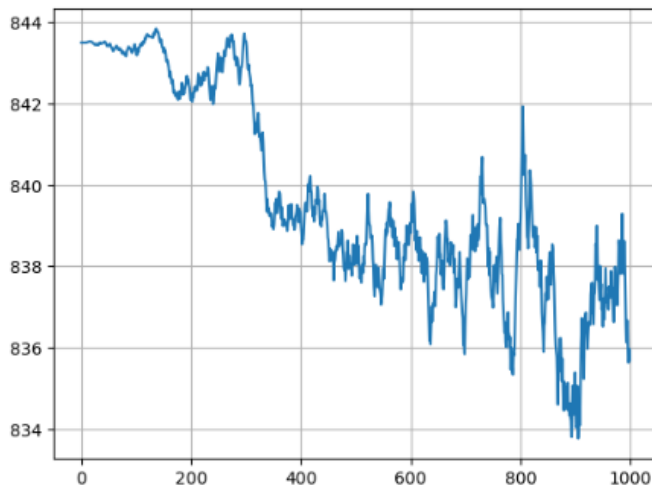


Figure 10. SA Schwefel con función lineal y semilla fijada en 571

```
[9.50274993 4.50196765]
X ( 200 ): [9.745972132841562, 5.348181060005811] FUNCTION VALUE ==> 17.827016560121834 Temp: 1091393642.127514
X ( 400 ): [4.919592954992667, 0.5218018821569099] FUNCTION VALUE ==> 11.839123252881294 Temp: 9.92616735063633e+16
X ( 600 ): [8.179338864753436, 3.781547791917675] FUNCTION VALUE ==> 15.758337447262262 Temp: 9.027796614315164e+24
X ( 800 ): [-0.16635971144997685, -4.564150784285743] FUNCTION VALUE ==> 11.423121782436047 Temp: 8.210733189403225e+32
X ( 1000 ): [-7.920378570810499, -7.920378570810499] FUNCTION VALUE ==> 16.210775589877695 Temp: 7.467618333433372e+40
```

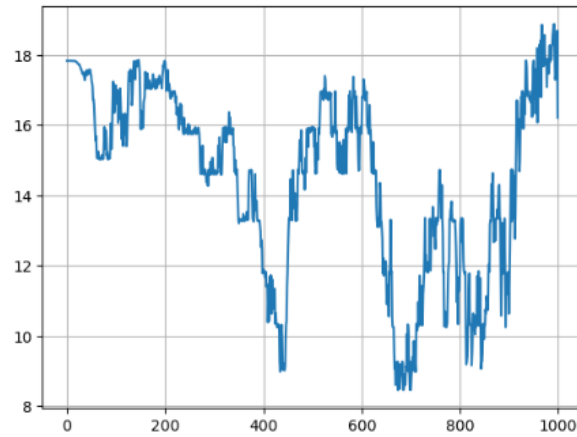


Figure 10. SA2 Ackley con función geométrica para temperatura con semilla fijada en 2122

```
[31.13861096 14.75204758]
X ( 200 ): [32.5293635578178, 17.540007323716623] FUNCTION VALUE ==> 1405.327354665943 Temp: 1091393642.127514
X ( 400 ): [24.799539781176758, 9.810183547075578] FUNCTION VALUE ==> 724.5022528354029 Temp: 9.92616735063633e+16
X ( 600 ): [25.00730050409207, 15.840483703951412] FUNCTION VALUE ==> 880.9126171184324 Temp: 9.027796614315164e+24
X ( 800 ): [11.527894879436507, 2.3610780792958566] FUNCTION VALUE ==> 174.7401333301389 Temp: 8.210733189403225e+32
X ( 1000 ): [-1.0539027070447702, -1.0539027070447702] FUNCTION VALUE ==> 3.3575455220823436 Temp: 7.467618333433372e+40
```

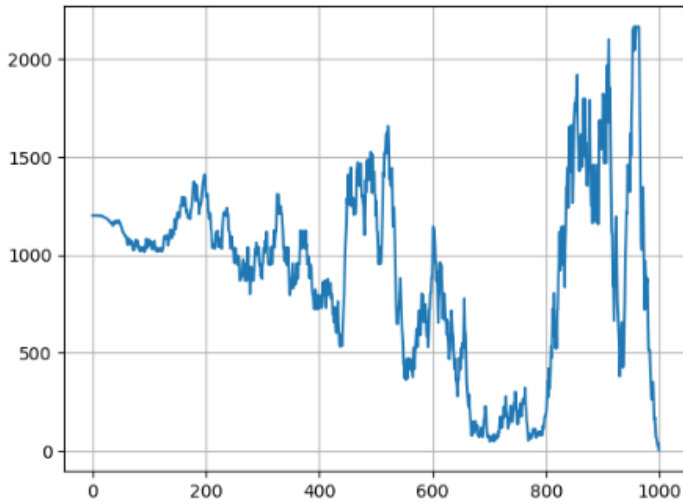


Figure 11. SA2 Rasgrign con función geométrica para temperatura con semilla fijada en 2122

```
[4.86540796 2.30500744]
X ( 200 ): [4.872441319738535, 2.7947472921581693] FUNCTION VALUE ==> 831.2671437114883 Temp: 1091393642.127514
X ( 400 ): [3.962802642042237, 3.2276888028538218] FUNCTION VALUE ==> 831.2014501655501 Temp: 9.92616735063633e+16
X ( 600 ): [3.2546668368840206, 2.519552997695602] FUNCTION VALUE ==> 832.2800764625742 Temp: 9.027796614315164e+24
X ( 800 ): [2.180711006466466, 1.445597167278046] FUNCTION VALUE ==> 834.4461614286681 Temp: 8.210733189403225e+32
X ( 1000 ): [-2.405935299581542, -3.1410491387699597] FUNCTION VALUE ==> 843.4487641320395 Temp: 7.467618333433372e+40
```

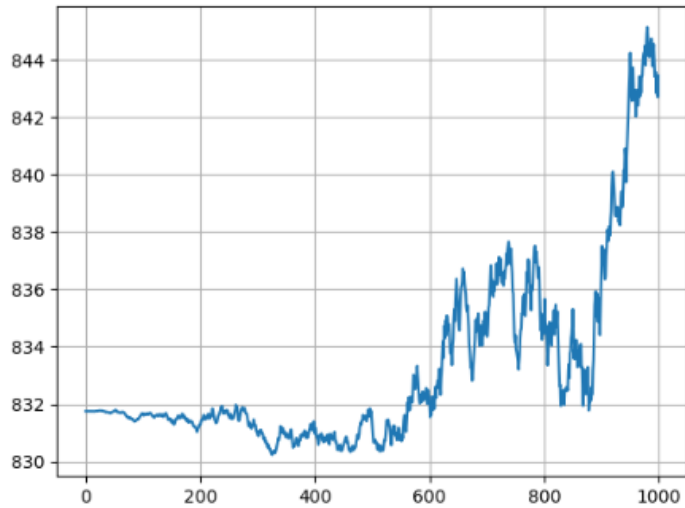


Figure 12. SA Schwefel con función geométrica para temperatura con semilla fijada en 2122

```
[-6.62248859 -4.42250532]
X ( 200 ): [-6.091231844143522, -3.891248571755502] FUNCTION VALUE ==> 13.279232331419959 Temp: 0.02103786816269284
X ( 400 ): [-6.0856794676823665, -3.885696195294347] FUNCTION VALUE ==> 13.276004431588408 Temp: 0.01025290498974709
X ( 600 ): [-6.083612863430865, -3.8836295910428453] FUNCTION VALUE ==> 13.27535897110062 Temp: 0.00677812923633077
X ( 800 ): [-5.311599816517539, -3.1116165441295207] FUNCTION VALUE ==> 13.130872658178442 Temp: 0.005062436719541004
X ( 1000 ): [-1.0894985411065077, 1.1104847312815103] FUNCTION VALUE ==> 4.426214946074779 Temp: 0.00403985995152168
```

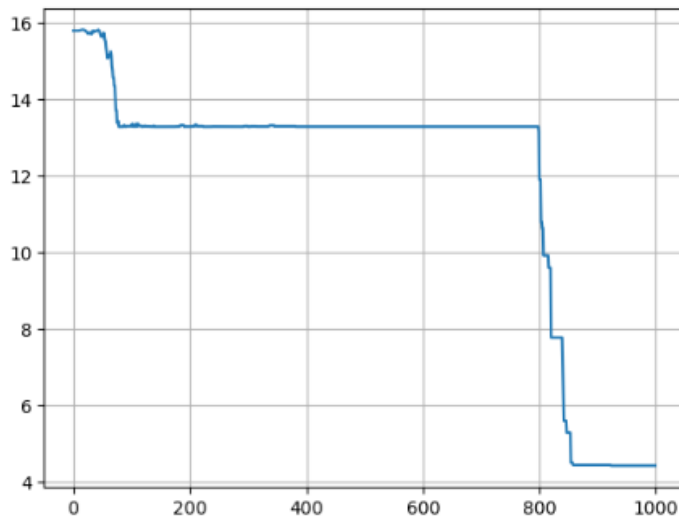


Figure 13. SA 3 Ackley con función exponencial para temperatura con semilla fijada en 571

```

[-21.70057062 -14.49166543]
X ( 200 ): [-14.94354649595155, -7.734641308990476] FUNCTION VALUE ==> 294.720295926866 Temp: 0.02103786816269284
X ( 400 ): [-4.085005951302586, 3.1238992356584903] FUNCTION VALUE ==> 30.71898616880275 Temp: 0.01025290498974709
X ( 600 ): [-3.1091053740553862, 4.099799812905689] FUNCTION VALUE ==> 30.63649719936968 Temp: 0.00677812923633077
X ( 800 ): [-3.1091053740553862, 4.099799812905689] FUNCTION VALUE ==> 30.63649719936968 Temp: 0.005062436719541004
X ( 1000 ): [-4.099655593284486, 3.10924959367659] FUNCTION VALUE ==> 30.63663768131871 Temp: 0.00403985995152168

```

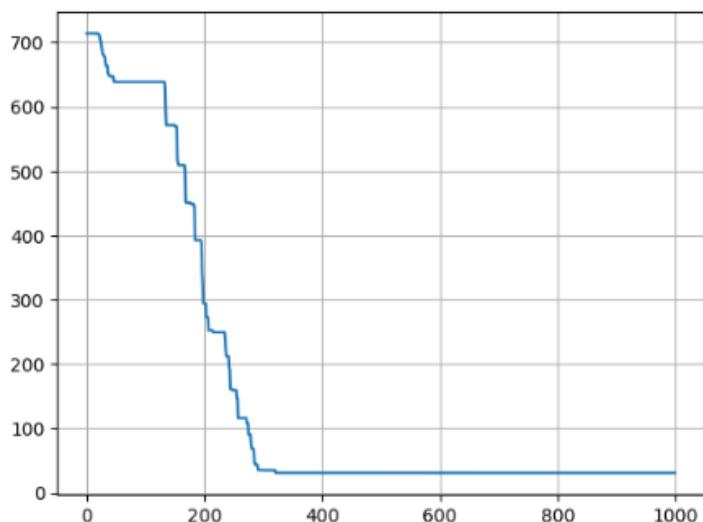


Figure 13. SA 3 Rasgrin con función exponencial para temperatura con semilla fijada en 571

```

[-3.39071416 -2.26432272]
X ( 200 ): [-0.9653660132295464, 0.16102542223311933] FUNCTION VALUE ==> 838.7059960442787 Temp: 0.02103786816269284
X ( 400 ): [5.005147595228821, 5.005147595228821] FUNCTION VALUE ==> 830.0973246913558 Temp: 0.01025290498974709
X ( 600 ): [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197 Temp: 0.00677812923633077
X ( 800 ): [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197 Temp: 0.005062436719541004
X ( 1000 ): [5.12, 5.12] FUNCTION VALUE ==> 830.0809359575197 Temp: 0.00403985995152168

```

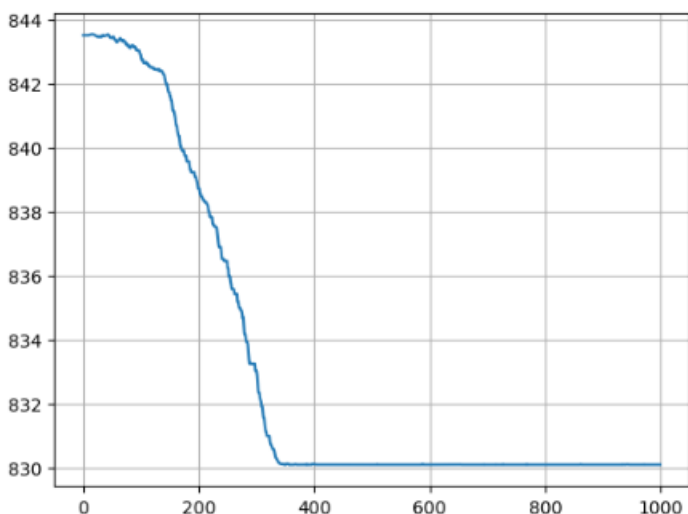


Figure 14. SA 3 Schwefel con función exponencial para temperatura con semilla fijada en 571

Análisis de lectura “Una aproximación a la heurística y metaheurísticas”

El artículo en cuestión inicia hablando sobre optimización y lo complicado que puede ser encontrar soluciones óptimas ya sea un mínimo o un máximo dado un problema, además de la complejidad, otra dificultad es que las herramientas analíticas ya no son suficientes, por lo que es necesario desarrollar métodos que nos permitan abordar este tipo de problemas.

Una heurística es un procedimiento en el que se espera encontrar soluciones óptimas con un coste computacional razonable, aunque no sea factible o exacto, y los problemas que se abordan bajo este lente usualmente son problemas de tipo NP que recordemos son problemas a los cuales no se puede garantizar encontrar una solución en un tiempo polinómico razonable.

Se asocian las siguientes características a los métodos heurísticos:

- 1) Al problema a resolver por este método no tiene solución bajo algún método exacto
- 2) O en caso contrario, si es que existiese un método exacto, es computacionalmente costoso o inviable.
- 3) El modelo matemático es grande o no lineal o muy complejo desde el punto de vista lógico.
- 4) Al intentar simplificar el problema, se tiende a destruir la estructura del modelo lo que hace que la solución encontrada no sea viable.
- 5) El método heurístico proporciona una buena solución inicial de partida o participa en un paso intermedio del procedimiento.

Se da el ejemplo de que el problema del viajero (TSP) se puede resolver con métodos heurísticos para un número grande de ciudades.

El artículo muestra una clasificación de los métodos heurísticos:

- Métodos de descomposición, el objetivo es dividir un problema complejo en partes más pequeñas y manejables.
- Métodos Inductivos, la idea es generalizar soluciones de casos simples a problemas más complejos.
- Métodos de Reducción, se busca reducir la complejidad del problema poniendo restricciones.
- Métodos Constructivos, se busca construir una solución paso a paso tomando las mejores decisiones en cada iteración.
- Métodos de búsqueda local, se busca mejorar una solución inicial a través de pequeños cambios iterativos.

Se menciona que la ventaja de abordar estos problemas bajo métodos heurísticos es que se tiene una mayor flexibilidad y robustez, mas sin embargo debido a que los métodos heurísticos pueden no arrojar resultados exactos se deben evaluar los resultados, por ello, debe ser eficiente, bueno (que la solución esté cerca del óptimo) y robusto.

Se puede evaluar la calidad por los siguientes procedimientos:

- a. Comparando con la solución óptima, Se calcula la desviación porcentual entre el costo de la solución encontrada por la heurística (ch) y el costo de la solución óptima ($copt$). Esto indica qué tan lejos está la solución de la heurística de la mejor solución posible: $(ch - copt) / copt * 100$.
- b. Comparación con una cota, se recurre a comparar la solución proporcionada por el heurístico con una cota, que es un valor que limita superiormente o inferiormente el valor de la solución óptima.
- c. Comparación con un método exacto truncado, se busca comparar la solución del heurístico con la mejor solución encontrada por un método exacto al que se le ha limitado el tiempo de ejecución o la cantidad de iteraciones
- d. Análisis del peor caso, se busca estudiar el comportamiento del heurístico en los casos más desfavorables para determinar la máxima desviación posible respecto a la solución óptima

Por otra parte, las metaheurísticas son estrategias para mejorar los procedimientos heurísticos orientados a tener un alto rendimiento, y de forma similar a los heurísticos se evalúa su rendimiento buscando las siguientes propiedades: simple, precisa, coherente, eficaz, eficiente, general, adaptable, robusta, interactiva, múltiple.

el artículo concluye describiendo una clasificación dentro de las metaheurísticas:

- Metaheurísticas Inspiradas en la física
- Metaheurísticas Inspiradas en la evolución
- Metaheurísticas Inspiradas en la biología

Finalmente concluye recordando que los problemas del mundo real muchas veces no se pueden modelar mediante modelos analíticos por lo que las heurísticas y metaheurísticas son un camino para poder abordar este tipo de problemas.