Práctica 1. “Introducción a la optimización en espacios continuos y metaheurísticas de búsqueda local.”

Dra. Miriam Pescador Rojas
Departamento de Ciencias e Ingeniería de la Computación
Algoritmos bioinspirados, ESCOM- IPN

(email: mpescadorr@ipn.mx)

11 de septiembre 2024

1 Introdución.

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

1.1 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:

$$\begin{array}{ll} \text{Encontrar} & \vec{x} = \{x_1, x_2, \ldots, x_n\} \text{ tal que} \\ \text{maximice (o minimice)} & f(x) \\ \text{sujeto a} & g_i(x) \leq b_i,\ i = 1, \ldots, m \\ & h_j(x) = c_j,\ j = 1, \ldots, p \end{array} \tag{1}$$

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 d erestricciones de desigualdad.
- p es el número de restricciones de igualdad.

2 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.

3 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.

3.1 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.

4 Consignas

4.1 Funciones de prueba

Los algoritmos se probarán con tres funciones de prueba clásicos, 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:

$$f(\vec{x}) = -ae^{-b\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}} - e^{\frac{1}{d}\sum_{i=1}^{d} cos(cx_i)} + a + e^1 \tag{2}$$

  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: $x_i \epsilon [-32.768, 32.768]$, para toda $i = 1, ..., d$.

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

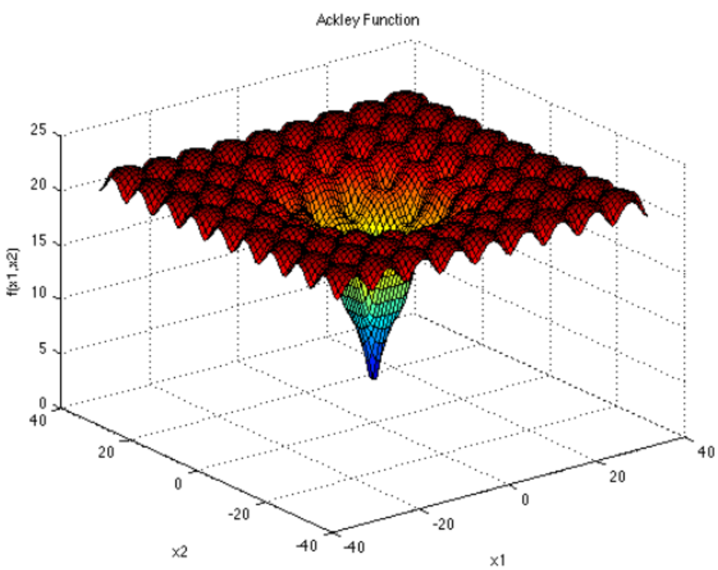

Figure 1: Función de Ackley

- **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:

$$f(\vec{x}) = 10d + \sum_{i=1}^{d} x_i^2 - 10cos(2\pi x_i) \tag{3}$$

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

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

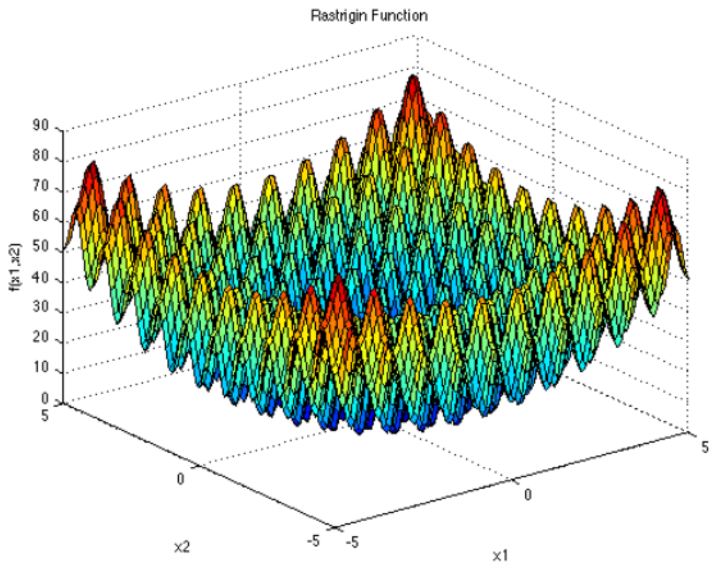

Figure 2: Función de Rastrigin

- **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:

$$f(\vec{x}) = 10d + \sum_{i=1}^{d} x_i^2 - 10cos(2\pi x_i) \tag{4}$$

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

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

Para consultar más detalles de las funciones de prueba vea el siguiente enlace
https://www.sfu.ca/ ssurjano/optimization.html

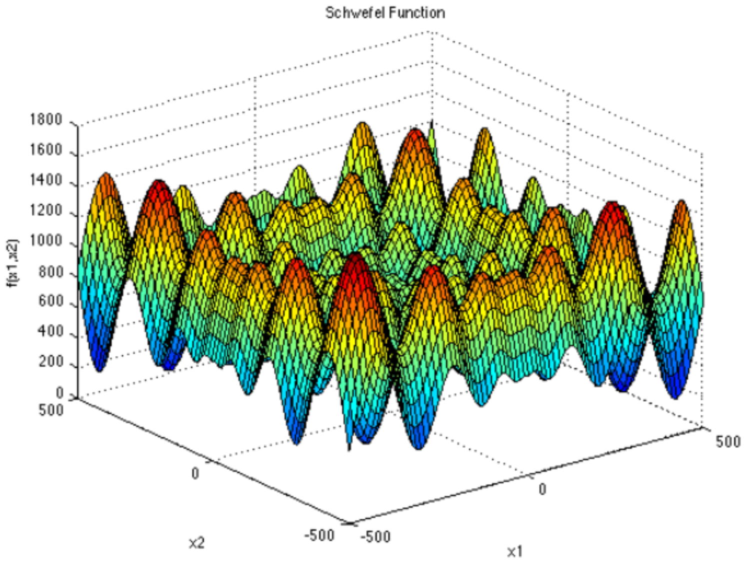

Figure 3: Función de Schwefel

4.2 Implementación de algoritmos de búsqueda heurística.

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.

Input: S_0: solución inicial
$maxT$: número máximo de iteraciones
Output: S_{act}: mejor solución visitada
begin
    $S_0 \leftarrow$ genera solución inicial
    $cont \leftarrow 1$
    while $cont \geq maxT$ **do**
        $S_{cand} \leftarrow$ Genera vecino aleatorio (S_0)
        if $f(S_{cand}) < f(S_0)$ **then**
            $S_0 \leftarrow S_{cand}$
        $cont++$
    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 temperatura
L: tiempo de cada nivel de temperatura (nota: para esta práctica L = 20)
Tf: temperatura final
Output: S_{act}: mejor solución visitada

begin
  $T \leftarrow T_0$
  S$_{act} \leftarrow generasolucióninicial$ **while** $T \geq T_f$ **do**
    for $count \leftarrow 1$ *to* $L(T)$ **do**
      $S_{cand} \leftarrow$ Genera vecino aleatorio (S_{act})
      $\delta \leftarrow cost(S_{cand}) - cost(S_{act})$
      if $U(0,1) < e^{\frac{-\delta}{T}}$ *or* $(\delta < 0)$ **then**
        $S_{act} \leftarrow S_{cand}$
    $T \leftarrow \alpha(T)$
  Devuelve la mejor solución S_{act} visitado

  Revise como referencia el código proporcionado en el enlace de colab.

  ```
  https://colab.research.google.com/drive/
  1G_YGOHVeZTDRhOxqpM_UliH_YINqmHZ6?
  usp=sharing
  ```

- Implementar al menos 3 diferentes funciones de enfriamiento mostradas en la tabla (figura 1).

  Para revisar detalles de implementación, revise el artículo: Amine, Khalil, 2019. Multiobjective Simulated Annealing: Principles and Algorithm Variants," Advances in Operations Research, Hindawi, vol. 2019, pages 1-13, May.

- Implemente 2 gráficas de convergencia: una para visualizar el valor de la solución óptima global por cada iteración y otra que muestre como baja la temperatura respecto a las iteraciones (grafique por separado para que se observe claramente las diferencias.)

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	$T_k = \dfrac{T_0}{\ln(k+1)}$	$T_k = \dfrac{\ln(k)}{\ln(k+1)} T_{k-1}$
Hybrid	$T_k = \begin{cases} \dfrac{T_0}{1+k} & \text{if } k \leq \beta \\ \alpha^k T_0 & \text{if } k > \beta \end{cases}$	$T_k = \begin{cases} \dfrac{k}{k+1} T_{k-1} & \text{if } k \leq \beta \\ \alpha T_{k-1} & \text{if } k > \beta \end{cases}$
Exponential	$T_k = \dfrac{T_0}{1 + k\beta T_0}$	$T_k = \dfrac{T_{k-1}}{1 + \beta T_{k-1}}$

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

4.3 Metodología experimental.

Con la finalidad de observar el efecto estocástico de los algoritmos implementados se deberá realizar al menos 2 ejecuciones con diferente semilla para generación de números aleatorios. El número máximo de iteraciones del algoritmo será 1,000 y se reportarán el valor mínimo encontrado cada 200 iteraciones. De tal manera que se genere una tabla como la que se muestra (veáse Tabla 1).

Table 1: Tabla ejemplo de los resultados de prueba

Función	**Semilla**	**# iteraciones**	**Hill climbing**		**SA-1**		**SA-2**		**SA-3**	
			x	f(x)	x	f(x)	x	f(x)	x	f(x)
Ackley	1	200								
		400								
		600								
		800								
		1000								
	2	200								
		400								
		600								
		800								
		1000								

5 Reporte

El reporte es preferible que se elabore en latex, este debe incluir los siguientes elementos.

- Análisis de la lectura sobre definición de heurísticas y metaheurísticas donde se muestre un resumen, que enfatice la diferencia entre uno y otro. La extensión mínima de este resumen es de una página con letra de tamaño 12pt.
- Sección con explicación de la implementación de la práctica con el código principal o más relevante, aquí debe indicar cuáles fueron las versiones de decremento de temperatura elegidas y por qué.

- Experimentos y resultados. Se refiere a la tabla con los resultados obtenidos y que se aprecie claramente la comparativa entre métodos y funciones de optimización que se probaron.
- Discusión de resultados. Indicar cuáles fueron las técnicas a las que les fue mejor, si hay algún patrón o tendencia.

  Como parte del análisis responda a las siguientes preguntas:

  – ¿En todos los métodos se encontró el óptimo global de la función de prueba?

  – ¿Cuál fue la estrategia que tuvo mejor desempeño?

  – ¿Cuáles son las recomendaciones que daría para la resolver problemas multimodales con recocido simulado?

  – ¿Qué propuestas de mejoras cree conveniente implementar para encontrar el óptimo global de los problemas de optimización continua?
- Conclusiones. Agregue conclusiones individuales respecto a los conocimientos adquiridos.

6 Requerimientos de entrega.

Se realizará una revisión de avance en laboratorio el día martes 17 de septiembre. Y la entrega final del reporte y código será el próximo **26 de septiembre de 2024** a más tardar a las 11:50 hrs.

El desarrollo de la práctica es individual o por parejas (solo envía uno de los integrantes), enviar por archivos separados el reporte en PDF y el código fuente si es más de un archivo también por separado. Deberá nombrar al reporte con el primer apellido de cada integrante. Por ejemplo: $Hernandez\_Perez.pdf$.

El envío de la práctica(enlace) es a través de Teams y habrá penalización de 10% sobre la calificación obtenida por cada día de retraso.