Tarea 1 - Problemas de Optimización

Ejercicio 1. Funciones de prueba para optimización continua

1.1. Investiga algunas (al menos 3) funciones que se utilizan para probar métodos de optimización continua. Las funciones deben ser diferentes a las vistas en clase.
   a. Para cada función, da una breve descripción / discusión de sus características
   b. ¿Cuál podría ser más fácil y difícil de optimizar?, ¿Por qué?
   c. ¿La dimensión podría alterar la dificultad?

1.2. Implementar las funciones del inciso anterior
Cada función deberá estar implementada en un método (o función) diferente. En todos los casos, se debe considerar que el parámetro de entrada será el valor del punto a evaluar x; dependiendo del lenguaje y la implementación, se puede recibir también la dimensión del problema.

Ejemplo:

```
double sphere( double x [ ] ) {
        double res = 0;
        for(int i=0; i < x.len; i++) {
        res += ….
        }
        return res;
}
```

1.3. Genera un programa que te permita evaluar las funciones del ejercicio anterior. El programa debe poder ejecutarse desde consola, y recibir todos los parámetros al momento de la ejecución.
La función a evaluar puede pasarse como un número o una cadena, para elegir alguna de las descritas anteriormente

Por ejemplo:

```
$ evaluar sphere 2 2.1 -0.1
```

En el ejemplo:
- “evaluar” es el nombre del ejecutable
- “sphere” es el nombre de la función que queremos evaluar (de manera alternativa, pueden implementarlo como un parámetro numérico, en cuyo caso, la línea de ejecución sería algo como: $ evaluar 1 2 2.1 -0.1
[el 1 es un número arbitrario, que debe definirse en el reporte para especificar que con ese número indicaremos la ejecución de la función sphere]
- 2 es la dimensión del problema
- Después de la dimensión, siguen n números, correspondientes a los valores de x_i

Ejercicio 2. Búsqueda aleatoria

2.a) Implementar una búsqueda aleatoria para problemas de optimización continua.

El programa deberá recibir como parámetros:

i) Función objetivo [Esto puede pasarse como un número o una cadena, para elegir alguna de las descritas anteriormente]
ii) Dimensión, que se utilizará para definir el espacio de búsqueda correspondiente
iii) Número total de iteraciones a realizar
iv) Intervalo de búsqueda

El programa deberá devolver, e imprimir en pantalla, el resultado de la búsqueda imprimiendo el valor de x (la mejor solución encontrada), así como su evaluación.

Ejemplo de ejecución y resultado esperado:

$ busqueda_aleatoria sphere 2 1000
Función: sphere
Dimensión del problema: 2
Total de iteraciones: 1000
Mejor solución encontrada:
x = [0.1 1.2]
f(x) = 300.86

2. b) Ejecutar la búsqueda aleatoria para todas las funciones anteriores, considerando 1,000,000 iteraciones; se deberá probar la búsqueda en diferentes dimensiones. Probar al menos con dimensiones 2, 5, 10.

Ejemplo de tabla que deben incluir en el reporte:

Función	**Dimensión**	**Mejor valor f(x)**	**Valor promedio f(x)**	**Peor valor f(x)**
Sphere		13.12	15.6	19.6
Ackley		...	...	...
….				

Esta tabla contendrá 3 renglones para cada una de las funciones (1 por cada valor de dimensión).

Ejercicio 3. Preguntas de repaso

1. Mencionar algún ejemplo de problema de optimización combinatoria, diferente a los mencionados en clase.
   Indicar claramente:
   - Espacio de búsqueda
   - Función objetivo
   - Tamaño del espacio de búsqueda
   - Ejemplar concreto del problema
   - Ejemplo de una solución

Consideraciones Generales

- Para la lectura de parámetros desde consola, pueden apoyarse de bibliotecas auxiliares (pero se debe agregar la referencia y aclaración correspondiente en el reporte)

El entregable para esta tarea deberá ser un archivo zip con la siguiente estructura:

+ # cuenta / <--- Nombre de la carpeta
  - src / <--- carpeta con el código fuente de su implementación
  - README.txt <--- Archivo con instrucciones para compilar y ejecutar Se debe incluir el comando para ejecutar un ejemplo de cada inciso
  - makefile <-- [Opcional]
  - reporteT1.pdf <--- Reporte de la Tarea
  - ejecuciones.csv <-- [Opcional] Hoja de Cálculo con la información de las ejecuciones realizadas.

El reporte (reporteT1.pdf) deberá incluir al menos:
- Nombre completo
- Título y número de Tarea
- Respuestas a los ejercicios planteados
- Pseudocódigo de algoritmos implementados
- Capturas de pantalla de la ejecución
- Comentarios / Conclusiones

El formato para el reporte es libre. Pueden usar word, latex, o cualquier otro procesador; es obligatorio que se incluya el archivo pdf. No hay extensión mínima ni máxima, pero deben incluir las respuestas / comentarios que se piden en cada uno de los ejercicios.