

Inteligencia Artificial 2022

Tarea 03

27.febrero.2022

1. Implementar el algoritmo de descenso máximo o alguna de sus variantes para encontrar el mínimo de las funciones que se indican a continuación. Para cada una de ellas, el algoritmo debe recibir como inputs los siguientes:

En cada uno de los casos, su función debe recibir los siguientes argumentos: la función objetivo f , el gradiente de la función objetivo df , un punto inicial $\mathbf{x}_0 \in \mathbb{R}^n$, el tamaño de paso $\alpha > 0$, el número máximo de iteraciones $maxIter$, la tolerancia ε , así como el criterio de paro que se va a utilizar.

Como resultado, sus algoritmos deben devolver: la mejor solución encontrada \mathbf{x} (la última de las aproximaciones calculadas); la secuencia de iteraciones \mathbf{x}_k ; la secuencia de valores $f(\mathbf{x}_k)$; la secuencia de errores en cada paso (según el error de su criterio de paro).

Además, es deseable indicar el número de iteraciones efectuadas por el algoritmo, y si se obtuvo o no convergencia del método. Para cada función, se debe reportar también la evolución de la convergencia (en una gráfica). De manera opcional, para las funciones en \mathbb{R}^2 pueden mostrar una gráfica con la trayectoria seguida por el método.

- a) La función $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, dada por

$$f(x, y) = x^4 + y^4 - 4xy + \frac{1}{2}y + 1.$$

- b) La función de Rosenbrock 2-dimensional $f : \mathbb{R}^2 \rightarrow \mathbb{R}$, dada por

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

- c) La función de Rosenbrock 10-dimensional $f : \mathbb{R}^{10} \rightarrow \mathbb{R}$,

$$f(x_1, x_2, \dots, x_{10}) = \sum_{i=1}^9 [100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2].$$

En cada uno de los casos, hallar un tamaño de paso α que garantice la convergencia de los métodos, y elabore una tabla con las primeras 4 y las últimas 4 aproximaciones \mathbf{x}_k obtenidas.

2. Implementar el algoritmo de enfriamiento simulado para resolver el problema TSP (*Traveling Salesman Problem*) para un conjunto de ciudades con coordenadas (x_k, y_k) , $1 \leq k \leq n$. Como input, el algoritmo debe recibir un archivo de texto en el siguiente formato:

```
n
x1 y1
x2 y2
...
xn yn
```

donde la primer línea contiene un entero positivo n , que indica el número de ciudades a visitar. Las siguientes n líneas del archivo, cada una contiene los x_k, y_k , los cuales representan las coordenadas de la localización de cada ciudad.

Además de este archivo, el algoritmo debe recibir como entrada un nodo inicial $1 \leq k \leq n$, el cual indicará la ciudad donde inicia y termina el recorrido.

Como resultado, su algoritmos debe indicar (en pantalla o en archivo) la secuencia de ciudades a visitar, comenzando con k y terminando con k . Adicionalmente, se puede desplegar el recorrido visual en pantalla.

