



DEPARTAMENTO  
DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

## TP3

9 de julio de 2025

Introducción a la Investigación Operativa y Optimización

Integrante	LU	Correo electrónico
Laks, Joaquín	425/22	laksjoaquin@gmail.com
Szabo, Jorge	1683/21	jorgecszabo@gmail.com
Wilders Azara, Santiago	350/19	santiago199913@gmail.com



**Facultad de Ciencias Exactas y Naturales**  
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón Cero + Infinito)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Conmutador: (+54 11) 5285-9721 / 5285-7400

<https://dc.uba.ar>

## 1. Introducción

En este trabajo práctico se implementaron distintos algoritmos para encontrar la ubicación óptima de un centro de servicio médico que responda a zonas afectadas con distintos niveles de atención necesarios. Se busca encontrar un punto que minimice la suma de las distancias euclidianas ponderadas respecto a un conjunto discreto de puntos, es decir, encontrar la mediana geométrica de dicho conjunto.

Formalmente, dado un conjunto de puntos  $\mathbf{P} = \{p_1, \dots, p_m\} \subset \mathbb{R}^n$  y pesos  $w_1, \dots, w_m > 0$ , el problema se reduce a encontrar:

$$\mathbf{x}^* = \arg \min_{\mathbf{x} \in \mathbb{R}^n} W(\mathbf{x}) = \sum_{i=1}^m w_i \|\mathbf{x} - p_i\|$$

## 2. Algoritmos

Se evaluaron tres algoritmos distintos para resolver el problema de minimizar  $M$ .

### 2.1. Weiszfeld

Se implementó la variante 1 del algoritmo de Weiszfeld (operador  $\tilde{T}$ ). Es un algoritmo iterativo que busca el punto fijo de:

$$T(\mathbf{x}) = \frac{\sum_{i=1}^n \frac{w_i p_i}{\|\mathbf{x} - p_i\|}}{\sum_{i=1}^n \frac{w_i}{\|\mathbf{x} - p_i\|}}$$

$T(\mathbf{x})$  converge al mínimo global de  $W$  si  $x^{(k)} \notin \mathbf{P} \quad \forall k$ .

La variante 1 consiste en encontrar un punto alternativo en las iteraciones donde se caiga en el caso de  $T(p_j)$  con  $p_j \in \mathbf{P}$ .

La implementación de la variante 1 es correcta porque en el caso de tener un punto  $p_j \in \mathbf{P}$  no óptimo, se calcula un punto  $S(p_j) = p_j + d_j t_j$  con un cierto paso y dirección de descenso que en la próxima iteración  $T$  no se evalúe en un punto de  $\mathbf{P}$ , así asegurando la eventual convergencia al punto fijo.

### 2.2. Método de Hooke y Jeeves

Aplicar el método de Hooke y Jeeves para este problema es correcto porque  $W$  es una función convexa. Esto quiere decir que un  $x^*$  mínimo local de  $W$  implica que es un posible mínimo global. Si los puntos de  $\mathbf{P}$  no están alineados  $W$  es estrictamente convexa y este  $x^*$  es único.

Este método no requiere del cálculo del diferencial de  $W$ , pero con la condición de convexidad se puede asegurar que el método no va a converger en mínimos locales, su convergencia va a ser en un mínimo global.

### 2.3. Descenso de gradiente

El descenso de gradiente consiste en encontrar un punto tal que  $0 \in \partial M(x^*)$ .

$M$  es una función convexa por lo que se puede asegurar que  $x^*$  es un mínimo global sii  $0 \in \partial M(x^*)$ . El resultado de aplicar el método de descenso de gradiente a  $M$  es una solución a el problema planteado.

### 3. Comparación de tiempos entre algoritmos

En esta sección se presenta la comparación en rendimientos de los distintos algoritmos, tanto en cantidad de iteraciones como en tiempo de procesamiento. Además en la Figura 1 se pueden ver ejemplos visuales de instancias de dos dimensiones para cada algoritmo y las diferencias en los pasos entre iteraciones.

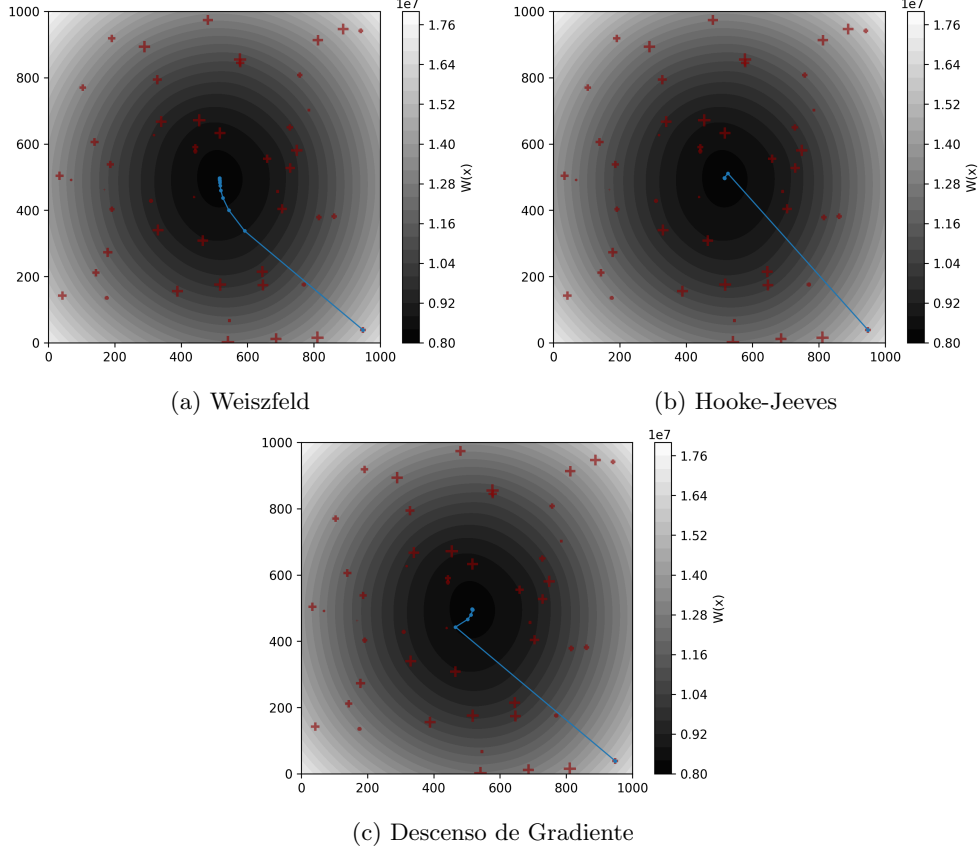
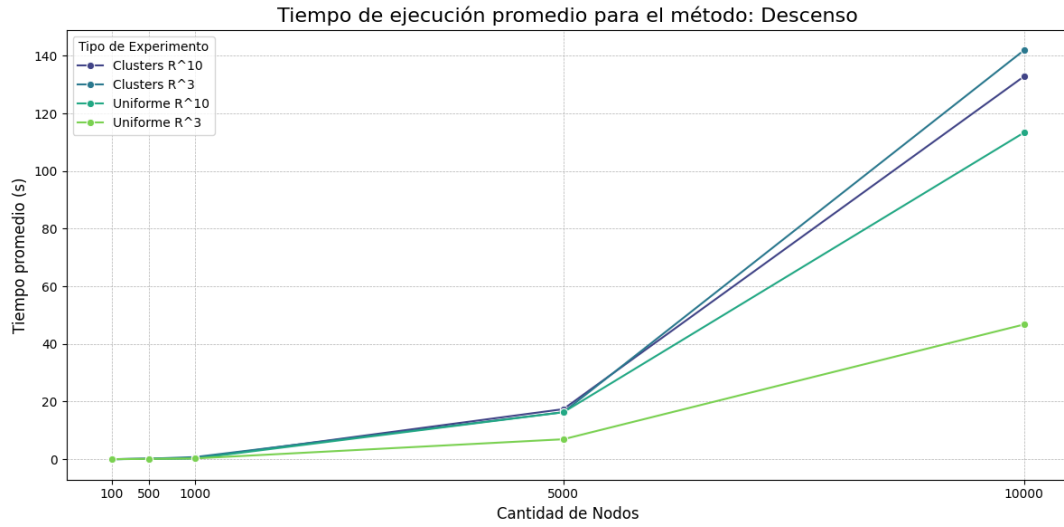


Figura 1: Recorridos para los distintos métodos en una misma instancia con 50 puntos ponderados y el contorno de  $W(x)$ .

Para el análisis implementamos dos tipos de instancias, una con distribución uniforme y otra con clusters definidos que generaban una mayor densidad alrededor de cinco centroides distintos. Para cada tipo generamos cinco replicas de instancias con 100, 500, 1000, 5000 y 10.000 nodos en dos dimensiones distintas,  $\mathbb{R}^3$  y  $\mathbb{R}^{10}$ . Cada instancia se resolvió con los tres algoritmos utilizando un mismo criterio de parada, que la norma de la diferencia entre una iteración y la siguiente sea menor a  $10^{-6}$ , esto es válido porque la función  $W$  que estamos evaluando es convexa, por lo que nos aseguramos estar convergiendo al mínimo global.

El tamaño de la grilla era de  $10n \times 10n$  siendo  $n$  la cantidad de puntos. Los pesos se generaron aleatoriamente de manera uniforme muestreando el intervalo  $(0, 10n)$ .

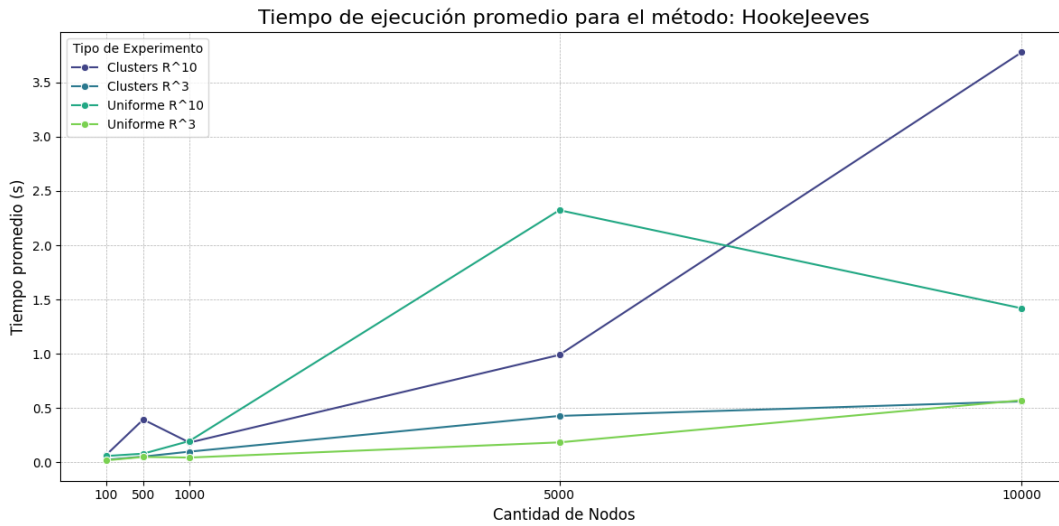
### 3.1. Método de descenso



(a) Tiempos de convergencia para el método de descenso

Como se puede observar en la figura 2a el tiempo de convergencia del algoritmo crece exponencialmente con la cantidad de nodos. Se puede observar que consistentemente los puntos distribuidos en clusters resultaban en un mayor tiempo de convergencia. Incluso en dimensiones más altas donde el costo de cada iteración aumenta por tener dimensiones de matrices mayores.

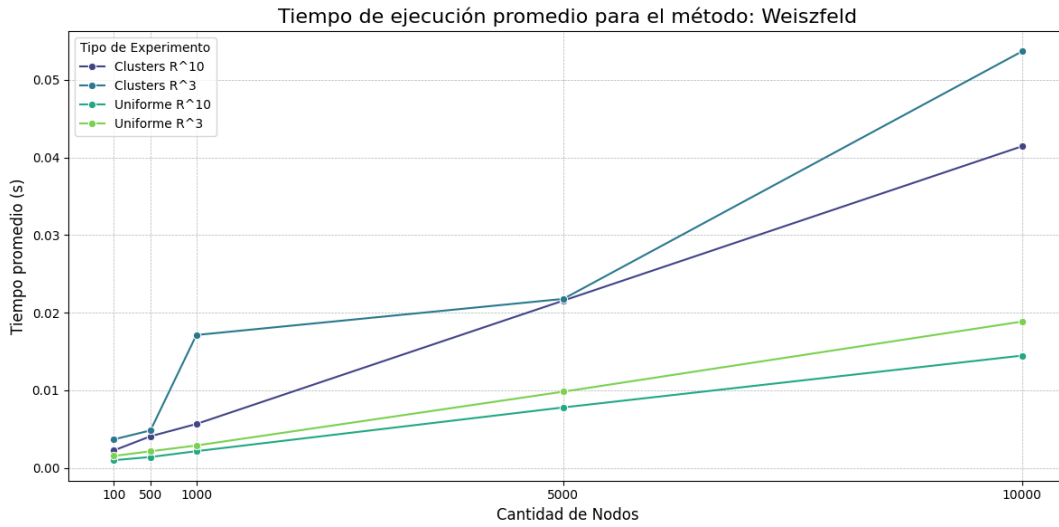
### 3.2. Método de Hooke-Jeeves



(a) Tiempos de convergencia para el método de Hooke-Jeeves

En la figura 3a se puede observar que la tasa de crecimiento del tiempo de convergencia de Hooke-Jeeves no es exponencial con la cantidad de nodos. Tampoco hay una clara relación entre densidad de puntos y el tiempo de convergencia. Si se puede observar que un aumento en la dimensión del espacio consistentemente resulta en un aumento en el tiempo de convergencia.

### 3.3. Algoritmo de Weiszfeld



(a) Tiempos de convergencia para el algoritmo de Weiszfeld

En la figura 4a se puede observar los tiempos de convergencia de la variante 1 del algoritmo de Weiszfeld. En regla general es bastante estable el tiempo de convergencia del algoritmo, y no crece desproporcionadamente con la cantidad de puntos.

En el caso de las instancias con puntos uniformemente distribuidos en la grilla el tiempo de convergencia es muy estable. Las instancias con dimensión más alta consistentemente convergen más rápido que las instancias con dimensión más baja.

Luego las instancias con puntos distribuidos en clusters convergen más lento que las distribuidas uniformemente. Acá se vuelve a repetir que las instancias de dimensión más alta convergen antes que las de dimensión más baja.

### 3.4. Conclusión

De las implementaciones en Python de los algoritmos se puede concluir que el algoritmo de Weiszfeld es la mejor alternativa para resolver el problema computacionalmente. La única desventaja de este algoritmo es que la dificultad de implementar un algoritmo ad-hoc es mayor a la de usar una solución genérica para optimizar funciones. Los tiempos de convergencia para las mismas instancias siempre fueron consistentemente órdenes de magnitud más rápidas que los métodos de Hooke-Jeeves y descenso.

Está abierta la posibilidad de explorar si se puede seleccionar un mejor criterio del paso en el método de descenso y si existen posibles optimizaciones utilizando un lenguaje distinto a Python y bibliotecas de cálculo numérico potencialmente mejores optimizadas que NumPy y SciPy.