



**DEPARTAMENTO
DE COMPUTACION**

Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico N°3

9 de Julio de 2025

Introducción a la Investigación Operativa y Optimización

Grupo 4

Integrante	LU	Correo electrónico
Suarez Ines	890/22	ine.suarez22@gmail.com
Navarro Solana	906/22	solanan3@gmail.com
Wittmund Montero, Lourdes	1103/22	lourdesmonterochiara@gmail.com



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires

Ciudad Universitaria - (Pabellón I/Planta Baja)

Intendente Güiraldes 2610 - C1428EGA

Ciudad Autónoma de Buenos Aires - Rep. Argentina

Tel/Fax: (+54 +11) 4576-3300

<http://www.exactas.uba.ar>

1. Introducción

En situaciones de desastre a gran escala (como terremotos, inundaciones o accidentes industriales) es fundamental brindar una respuesta médica rápida y efectiva para minimizar los daños y salvar vidas. Uno de los principales desafíos logísticos en estos contextos es decidir la ubicación óptima para establecer centros de servicios médicos temporales. La dificultad se agrava por la falta de información precisa sobre el estado de las rutas o el tráfico, por lo que se requiere una modelización robusta y eficiente.

Este trabajo se enfoca en resolver un problema clásico de localización conocido como problema de Fermat-Weber, cuya meta es determinar el punto óptimo que minimiza la suma de distancias ponderadas desde dicho punto a un conjunto de ubicaciones determinadas, cada una con un peso asociado que representa su nivel de prioridad o demanda. En este caso, se adopta como medida de distancia la distancia euclídea, que actúa como una aproximación razonable ante la incertidumbre sobre la red vial real.

La resolución de este problema se aborda desde el enfoque de la optimización no lineal, mediante la implementación y comparación de tres métodos: el algoritmo de Weiszfeld, un método de búsqueda directa (en nuestro trabajo utilizaremos el de descenso coordenado), y un método basado en direcciones de descenso (en nuestro trabajo utilizaremos el del gradiente con el paso elegido según el criterio de Armijo). Además, se analiza el comportamiento de cada algoritmo frente a diferentes configuraciones de entrada y dimensiones, evaluando su desempeño en términos de convergencia y eficiencia.

2. Formulación del Problema

El problema que se busca resolver es una instancia del conocido problema de *Fermat-Weber*, el cual consiste en determinar una ubicación óptima $x \in \mathbb{R}^n$ que minimice la suma de las distancias euclídeas ponderadas desde dicho punto a un conjunto de puntos dados $p_i \in \mathbb{R}^n$ con pesos asociados $w_i > 0$, para $i = 1, \dots, m$. Formalmente, el problema puede expresarse como:

$$x^* = \arg \min_{x \in \mathbb{R}^n} \sum_{i=1}^m w_i \|x - p_i\|_2$$

donde:

- p_i representa la posición del punto i en el espacio.
- w_i es un peso positivo asociado al punto p_i , que puede interpretarse como un nivel de prioridad o demanda de asistencia.
- $\|x - p_i\|_2$ denota la distancia euclídea entre el punto buscado x y el punto p_i .

Este modelo tiene múltiples aplicaciones en contextos reales, particularmente en problemas de localización de infraestructuras críticas. En el marco de este trabajo, se utiliza para decidir la ubicación de un centro de atención médica temporario que maximice la eficiencia en la asistencia frente a una catástrofe. Al utilizar la distancia euclídea como métrica y ponderarla por niveles de prioridad, se obtiene un enfoque razonable que permite sortear la incertidumbre sobre el estado real de las rutas o del tráfico en situaciones de emergencia.

3. Implementación

La implementación fue realizada en Python, encapsulada en una clase **FermatWeberSolver**, que contiene métodos para evaluar la función objetivo, calcular el gradiente, aplicar la fórmula de cada método y ejecutar el algoritmo completo.

Se consideraron diferentes configuraciones de distribución de puntos: uniforme, agrupada (clusters) y lineal, junto con dos esquemas de pesos (constantes y decrecientes exponencialmente). Se evaluó además la sensibilidad del algoritmo al punto inicial, comparando en la distribución de clusters el comportamiento al iniciarlo en el centroide, en el origen y en un punto aleatorio.

4. Algoritmos Implementados

4.1. Algoritmo de Weiszfeld

El siguiente esquema describe el algoritmo implementado, basado en la versión modificada del método de Weiszfeld, adaptado para evitar inestabilidades numéricas cuando el iterado coincide con uno de los puntos de datos. Sea $\mathbf{x}_0 \in \mathbb{R}^m$ un punto inicial.

Paso general: para $k = 0, 1, \dots$

$$\mathbf{x}_{k+1} = \tilde{T}(\mathbf{x}_k) = \begin{cases} T(\mathbf{x}_k), & \mathbf{x}_k \notin \mathcal{P}, \\ \mathbf{p}_j, & \mathbf{x}_k = \mathbf{p}_j \text{ y } \|\mathbf{R}_j\| \leq w_j, \\ S(\mathbf{p}_j), & \mathbf{x}_k = \mathbf{p}_j \text{ y } \|\mathbf{R}_j\| > w_j, \end{cases}$$

donde:

- $\mathcal{P} = \{\mathbf{p}_1, \dots, \mathbf{p}_m\}$ es el conjunto de puntos dados,
- $w_j > 0$ es el peso asociado al punto \mathbf{p}_j ,
- $T(\mathbf{x})$ es el operador clásico de Weiszfeld:

$$T(\mathbf{x}) = \frac{\sum_{i=1}^m \frac{w_i}{\|\mathbf{x} - \mathbf{p}_i\|} \mathbf{p}_i}{\sum_{i=1}^m \frac{w_i}{\|\mathbf{x} - \mathbf{p}_i\|}},$$

- \mathbf{R}_j es el vector de corrección definido por:

$$\mathbf{R}_j = \sum_{\substack{i=1 \\ i \neq j}}^m \frac{w_i(\mathbf{p}_j - \mathbf{p}_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|},$$

- $S(\mathbf{p}_j)$ es un paso alternativo desde \mathbf{p}_j en la dirección de descenso de la función objetivo, dado por:

$$S(\mathbf{p}_j) = \mathbf{p}_j + d_j t_j,$$

con:

$$d_j = -\frac{\mathbf{R}_j}{\|\mathbf{R}_j\|}, \quad t_j = \frac{\|\mathbf{R}_j\| - w_j}{\sum_{i \neq j} \frac{w_i}{\|\mathbf{p}_i - \mathbf{p}_j\|}}.$$

Este procedimiento asegura estabilidad numérica y manejo adecuado de los casos degenerados en que \mathbf{x}_k coincide con uno de los puntos de datos. La elección entre p_j y $S(p_j)$ depende de si $\|\mathbf{R}_j\|$ supera o no el peso w_j .

4.2. Método de Descenso Coordinado

El método de descenso coordinado es una técnica de optimización que itera sobre las variables del problema, optimizando una coordenada a la vez mientras mantiene fijas las demás. En este trabajo, se utilizó una variante basada en el criterio de *Gauss-Southwell*, que elige en cada iteración la coordenada asociada al mayor valor absoluto del gradiente.

Descripción del algoritmo

El procedimiento implementado consiste en los siguientes pasos:

1. Se calcula el gradiente de la función objetivo en el punto actual $x^{(k)}$.
2. Se selecciona la coordenada i tal que $|\nabla f(x^{(k)})_i|$ es máxima.
3. Se realiza una búsqueda unidimensional sobre la dirección de la coordenada i , resolviendo

$$\lambda^* = \arg \min_{\lambda \in \mathbb{R}} f(x^{(k)} + \lambda e_i)$$

donde e_i es el vector canónico.

4. Se actualiza la solución con $x^{(k+1)} = x^{(k)} + \lambda^* e_i$.

Para la búsqueda unidimensional se empleó el método **brent** de la librería **SciPy**, que no requiere derivadas y es eficiente en términos de evaluaciones de la función objetivo.

En el paso 2 del algoritmo se selecciona la coordenada con mayor valor absoluto del gradiente. Sin embargo, la función objetivo

$$f(x) = \sum_{j=1}^m w_j \|x - p_j\|$$

no es diferenciable en los puntos p_j , ya que el gradiente $\nabla f(x)$ involucra términos del tipo $\frac{x-p_j}{\|x-p_j\|}$, los cuales son indefinidos si $x = p_j$.

Para resolver este problema, el código identifica explícitamente si el punto actual $x^{(k)}$ coincide con alguno de los puntos de demanda p_j . En ese caso, en lugar de intentar calcular el gradiente (que no existe), construye un subgradiente válido de la función objetivo evaluado en p_j , definido como:

$$\partial f(p_j) = \sum_{i \neq j} w_i \cdot \frac{p_j - p_i}{\|p_j - p_i\|}$$

Este subgradiente omite el término singular y permite continuar con el criterio de Gauss-Southwell sin errores numéricos.

En caso de que $x^{(k)}$ no coincida con ningún p_j , se utiliza el gradiente estándar:

$$\nabla f(x) = \sum_{j=1}^m w_j \cdot \frac{x - p_j}{\|x - p_j\|}$$

De esta forma, se asegura una dirección válida de descenso en todos los casos, respetando la estructura de la función objetivo, que es convexa pero no diferenciable en los puntos p_j .

4.3. Método de Descenso con Dirección (Gradiente + Armijo)

Este método se basa en una estrategia clásica de descenso por gradiente, donde la dirección de descenso es el gradiente negativo de la función objetivo, y el tamaño del paso (α_k) se determina dinámicamente usando la condición de Armijo. Sin embargo, debido a que la función de Fermat-Weber no es diferenciable en los puntos que coinciden exactamente con alguno de los puntos dados, se utiliza un subgradiente en esos casos.

Descripción del algoritmo

En cada iteración k , se calcula el gradiente $\nabla f(x_k)$ de la función objetivo:

$$\nabla f(x) = \sum_{j=1}^m w_j \cdot \frac{x - p_j}{\|x - p_j\|}$$

El nuevo punto se actualiza como:

$$x_{k+1} = x_k - \alpha_k \cdot \nabla f(x_k)$$

El valor de α_k se escoge mediante la condición de Armijo:

$$f(x_k - \alpha_k \nabla f(x_k)) \leq f(x_k) - c_1 \alpha_k \|\nabla f(x_k)\|^2$$

donde $c_1 \in (0, 1)$ es un parámetro fijo (en nuestro caso $c_1 = 0,1$). Si la condición no se cumple, α_k se reduce multiplicativamente hasta que sí lo haga.

Manejo del punto no diferenciable

Cabe destacar que la función objetivo no es diferenciable en los puntos $x = p_j$ del conjunto de entrada, ya que en esos casos $\|x - p_j\| = 0$ y se anula el denominador del gradiente. En esos puntos, el gradiente no está definido. Para resolver este problema, la implementación detecta si x coincide con alguno de los puntos p_j y en ese caso calcula un *subgradiente* en lugar del gradiente.

En particular, si $x = p_j$, el subgradiente se define como:

$$\partial f(p_j) = \sum_{i \neq j} w_i \cdot \frac{p_j - p_i}{\|p_j - p_i\|}$$

El código retorna este subgradiente y lo utiliza como dirección de descenso en combinación con la regla de Armijo.

Este manejo permite que el método continúe funcionando aún cuando se alcanza un punto no diferenciable, lo cual es frecuente en el problema de Fermat-Weber.

5. Criterio de parada

En los tres algoritmos el criterio de parada está basado en la norma relativa del cambio en la variable iterada. Sea x_k la solución en la iteración k y x_{k+1} la siguiente, se calcula:

$$\|x_{k+1} - x_k\| < \varepsilon \cdot \|x_k\|$$

donde ε es una tolerancia pequeña definida por el usuario (en este caso, $\text{tol} = 10^{-4}$).

El algoritmo se detiene cuando la norma del cambio entre iterados es suficientemente pequeña en comparación con la magnitud actual de la solución, lo que indica que el proceso está convergiendo y las actualizaciones son insignificantes.

Motivación del criterio

- Este criterio mide directamente el progreso en las variables de decisión, asegurando que el algoritmo no continúe iterando cuando las soluciones consecutivas son casi iguales.
- La comparación relativa (multiplicando por $\|x_k\|$) es importante para que el criterio sea escalable y no dependa de la magnitud absoluta de la solución.
- Es especialmente útil en problemas donde la función objetivo puede no cambiar mucho en valor, pero el vector solución todavía cambia.
- Además, el algoritmo incluye un límite máximo de iteraciones para evitar ejecuciones indefinidas.

6. Análisis de los resultados

Para hacer una comparación y un análisis principal sobre los distintos métodos realizamos experimentos en \mathbb{R}^5 y graficamos el valor de la función objetivo en función de la cantidad de iteraciones en distintas distribuciones con distintos pesos. Todos los gráficos con título **Pesos 1** estan hechos con pesos constantes (iguales a 1 para todos los puntos) para poder comparar cómo se comportan nuestros algoritmos cuando no esta introducida la prioridad de las distintas ciudades y sólo resulta relevante la distribución geométrica. En cambio los **Pesos 2** estamos creando pesos que decaen exponencialmente: los primeros puntos tienen mucho peso, y los últimos casi nada. Esta disposición de pesos simula casos más realistas o extremos en los que ciertos datos son mucho más relevantes, y nos permite testear qué tan robusto y confiable es cada método en distintos escenarios.

6.1. Distribución Uniforme

Generamos puntos p_i de forma independiente y uniforme en el hipercubo $[-10, 10]^5$. Esta distribución representa un caso base sin estructura particular ni concentración de datos, y permite evaluar el desempeño del algoritmo en un entorno homogéneo.

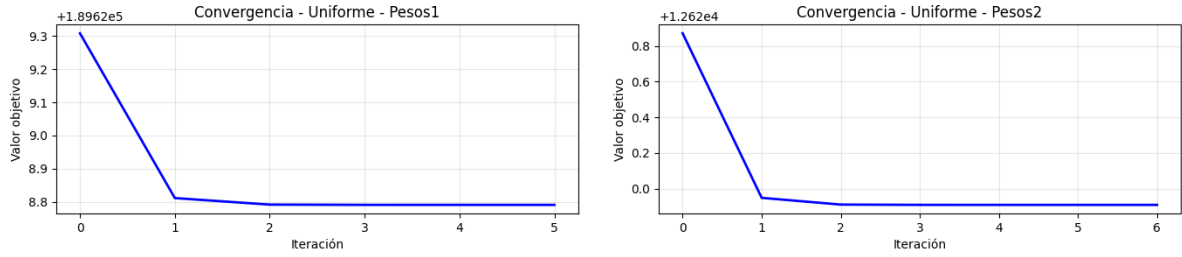


Figura 1: Convergencia uniforme con Weiszfeld

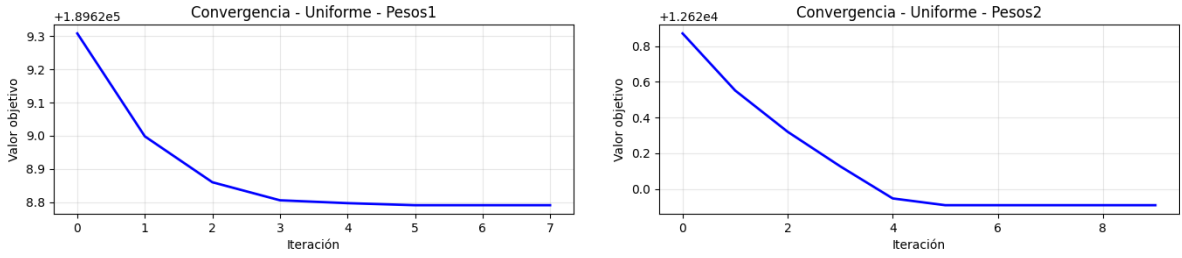


Figura 2: Convergencia uniforme con descenso coordinado

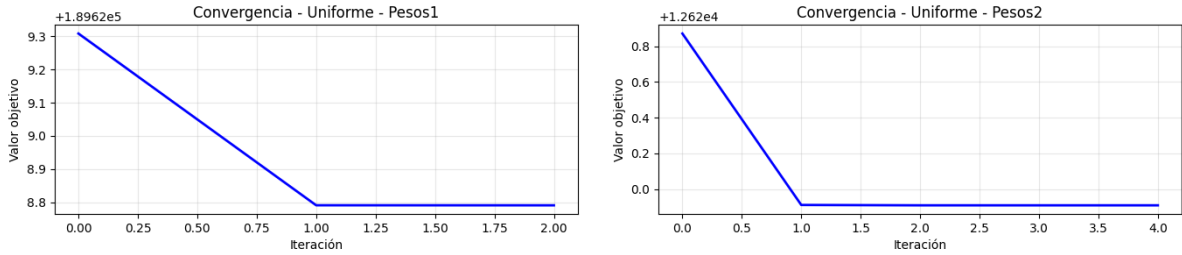


Figura 3: Convergencia uniforme con metodo del gradiente

El método del gradiente mostró el comportamiento más rápido y directo. En ambas configuraciones de pesos, alcanzó el valor objetivo final ya en la primera iteración, aunque la convergencia formal (según el criterio de tolerancia impuesto) se verificó en unas pocas iteraciones más.

Como los puntos están distribuidos uniformemente en un espacio limitado, esta simetría favorece que el gradiente inicial ya apunte muy cerca de la solución óptima, lo que justifica que se alcance el valor objetivo final en la primera iteración. En el caso de **pesos variables**, aunque la magnitud de las contribuciones individuales cambia, los pesos están espacialmente repartidos, por lo que no generan un sesgo geométrico fuerte.

El método de Weiszfeld tuvo un rendimiento excelente, alcanzando el valor objetivo óptimo de forma muy rápida, aunque ligeramente más progresiva que el gradiente. En ambas configuraciones de pesos, la convergencia fue muy rápida (prácticamente en la misma iteración que el gradiente o apenas después). Esto se debe a que su actualización no utiliza una dirección de descenso explícita sino una fórmula racional basada en la posición relativa de los puntos.

En el contexto de una distribución uniforme, esta estrategia resulta especialmente eficaz, ya que la dispersión regular de los puntos en el espacio asegura que el punto promedio ponderado —calculado inversamente proporcional a la distancia— esté ya muy cercano a la solución óptima. Es decir, la simetría y regularidad de la distribución favorecen el mecanismo del método, permitiendo que cada actualización esté muy alineada con la dirección que minimiza la función objetivo.

El método de descenso coordinado también logró buenos resultados, pero de forma ligeramente más progresiva que los anteriores. En el caso de pesos constantes, alcanzó el valor objetivo final en la tercera iteración, y la convergencia formal se dio en la séptima. Para pesos variados, el valor final se alcanza recién en la cuarta iteración, y converge formalmente en la octava.

Esta progresividad es esperable: al actualizar una coordenada a la vez, el método avanza de manera

más lenta, especialmente en espacios de dimensión moderada o alta. Sin embargo, su simplicidad lo hace útil en problemas donde el cálculo completo del gradiente es costoso o inestable.

6.2. Distribución outliers

Se generó un conjunto principal de 15.000 puntos a partir de una distribución normal centrada en el origen con desviación estándar moderada, valor de varianza igual a 25 y se incorporan un 5 % de *outliers* ubicados aleatoriamente en una región alejada (por ejemplo, entre 50 y 100 en cada coordenada). Esta configuración permite evaluar la sensibilidad del algoritmo frente a valores atípicos que pueden alterar la solución óptima significativamente.

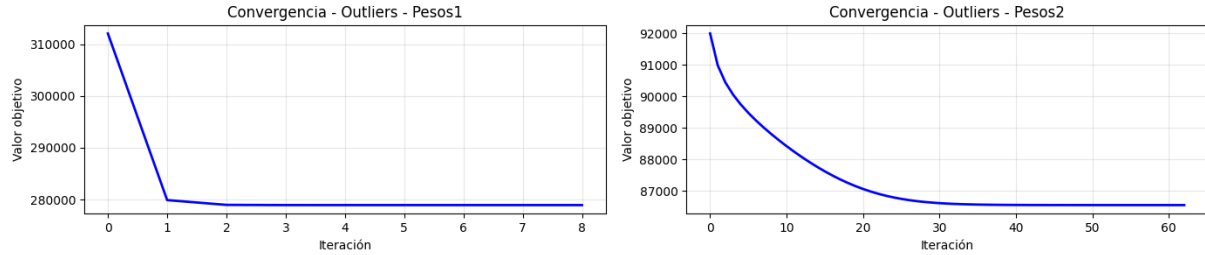


Figura 4: Convergencia de outliers con Weiszfeld

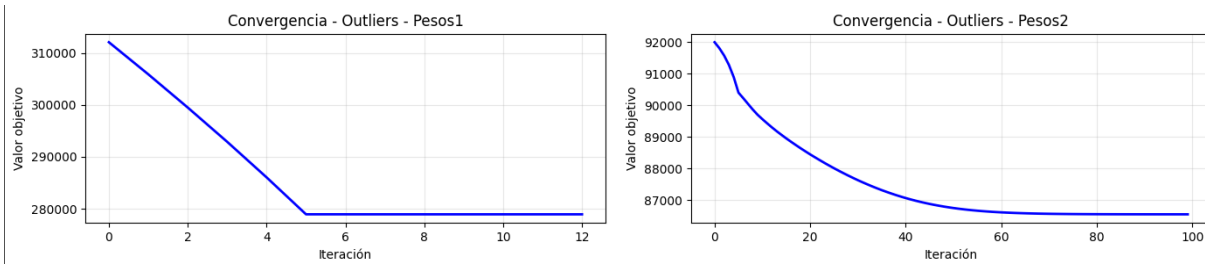


Figura 5: Convergencia de outliers con descenso coordenado

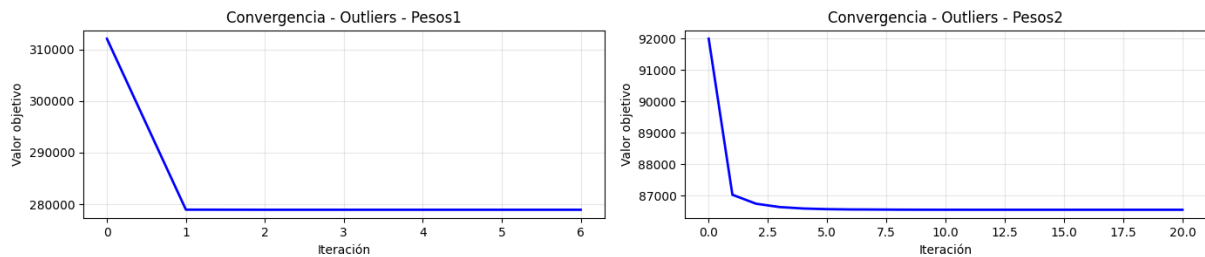


Figura 6: Convergencia de outliers con metodo del gradiente

Al comparar las curvas de convergencia entre los tres métodos en la configuración con outliers, se observa un patrón claro que distingue el comportamiento relativo de cada técnica. En particular, el método del gradiente con criterio de Armijo nuevamente es el que converge más rápidamente en ambas configuraciones de pesos. Esto se refleja en que, tanto con pesos constantes (Figura 6, izquierda) como con pesos decrecientes (Figura 6, derecha), el valor objetivo cae de forma abrupta en las primeras iteraciones, alcanzando prácticamente el mínimo desde la iteración 1 o 2. Aunque el criterio de convergencia se satisface unas iteraciones después, la mejora efectiva en el valor objetivo ocurre casi de inmediato. Este comportamiento contrasta con los otros métodos, cuyo descenso es más progresivo.

En comparación, el método de Weiszfeld (Figura 4) también muestra un rendimiento excelente con pesos constantes: al igual que el gradiente, realiza un descenso pronunciado en las primeras iteraciones y estabiliza el valor objetivo rápidamente. Sin embargo, en el caso de pesos decrecientes, el comportamiento

cambia drásticamente: el descenso se vuelve mucho más lento, extendiéndose por más de 60 iteraciones. Esta diferencia es visible en la gráfica, donde la curva desciende suavemente, sin saltos bruscos, debido a que los outliers reciben pesos bajos y, por lo tanto, ejercen poca influencia en el promedio ponderado que actualiza la posición. Esto hace que el algoritmo necesite muchas más iteraciones para adaptarse al conjunto de datos disperso.

El descenso por coordenadas (Figura 5), por su parte, se ubica en un punto intermedio. Con pesos constantes, el valor objetivo desciende rápidamente durante las primeras iteraciones y luego se estabiliza, aunque lo hace de manera más gradual que los otros métodos. En el caso de pesos decrecientes, el método también experimenta una ralentización significativa, y el descenso se extiende hasta la iteración 100. A diferencia de Weiszfeld, esta ralentización no se debe solo a los pesos, sino también a la naturaleza del algoritmo: al actualizar una coordenada por vez, la progresión global es más lenta, especialmente cuando los datos presentan fuerte dispersión como en el caso de los outliers.

En conjunto, las imágenes revelan que el método del gradiente es el más eficiente y menos afectado por el cambio de pesos, manteniendo una convergencia rápida y directa incluso en presencia de datos extremos. Weiszfeld es muy eficiente con pesos constantes, pero sufre una desaceleración notoria con pesos variables. El descenso por coordenadas es el más afectado en términos de iteraciones requeridas, especialmente con ponderaciones que atenúan la influencia de los puntos alejados.

6.3. Distribución Clusters

En este caso, los datos se generan agrupándolos en varios centros claramente definidos. Específicamente, se eligen aleatoriamente 5 centros distribuidos uniformemente en un hiperespacio de dimensión 5, y luego se generan los puntos alrededor de estos centros, siguiendo una distribución normal multivariada con desviación estándar moderada. Cada punto se asigna aleatoriamente a uno de los centros, lo que da lugar a un conjunto de datos compuesto por conjuntos densos (clusters) relativamente separados entre sí.

Esta configuración simula situaciones reales donde los datos presentan estructura interna: por ejemplo, clientes agrupados por comportamiento, sensores agrupados por ubicación, o ciudades agrupadas por región geográfica.

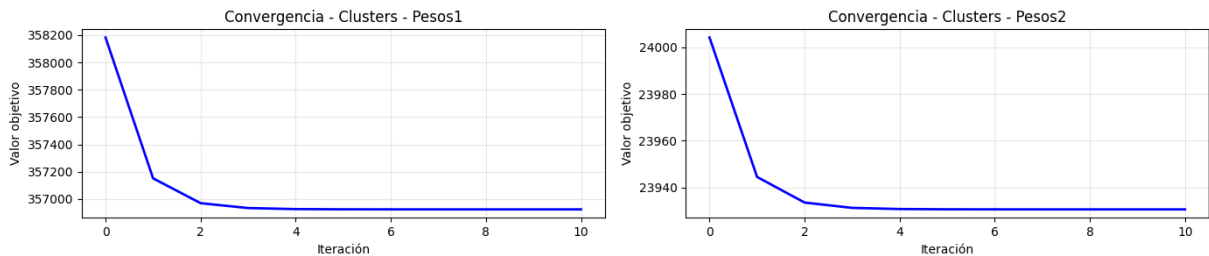


Figura 7: Convergencia de clusters con Weiszfeld

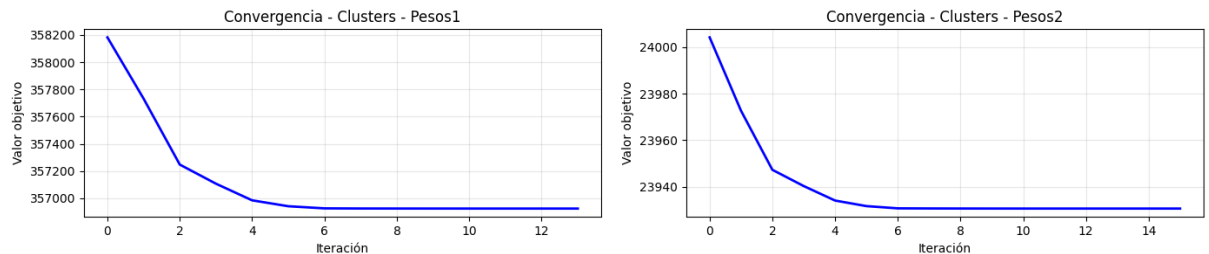


Figura 8: Convergencia por clusters con descenso coordenado

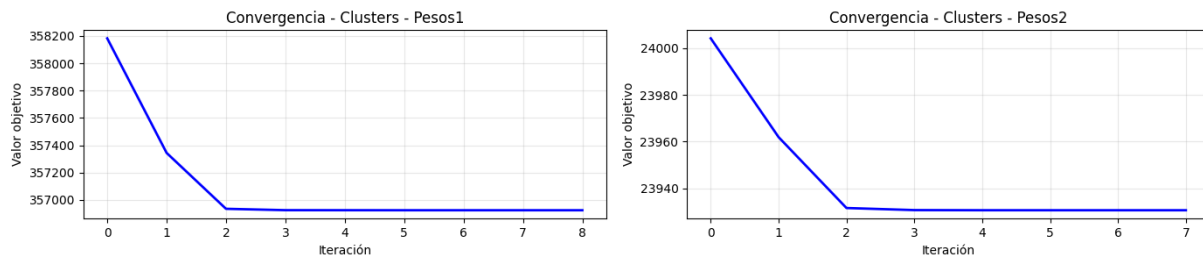


Figura 9: Convergencia en clusters con metodo del gradiente

En el caso de datos agrupados en clusters, los tres métodos muestran un comportamiento muy eficiente en términos de convergencia, aunque con diferencias sutiles que emergen al comparar las curvas entre sí.

El método del gradiente con criterio de Armijo (Figura 9) vuelve a ser el más rápido en ambos casos de pesos. Con pesos constantes, el descenso del valor objetivo es muy abrupto desde la primera iteración, y ya en la iteración 2-3 alcanza prácticamente el mínimo. Con pesos decrecientes, la mejora es incluso más brusca. Este rendimiento se explica por la capacidad del método para seguir la dirección de descenso más pronunciado con pasos adaptativos, lo que le permite saltar rápidamente entre regiones de influencia de diferentes clusters y acercarse a una solución central.

El método de Weiszfeld también muestra un buen comportamiento de convergencia para datos agrupados en clusters, tal como se observa en la Figura 7. Tanto con pesos constantes como con pesos decrecientes, la función objetivo desciende de forma pronunciada en las primeras iteraciones y se estabiliza rápidamente. Con pesos constantes, el algoritmo alcanza una mejora sustancial ya en la iteración 2, y a partir de la iteración 5 el valor objetivo permanece prácticamente constante, lo que indica una rápida convergencia hacia un punto cercano al óptimo. Este comportamiento es esperable dado que los clusters están bien definidos y repartidos simétricamente, lo que favorece un equilibrio eficiente entre los grupos. Con pesos decrecientes, se observa un patrón muy similar: una caída inicial pronunciada y estabilización casi completa hacia la iteración 5. A pesar de que los pesos asignan mayor importancia a los primeros puntos (y por ende podrían sesgar la solución), en este caso el sesgo no altera significativamente la trayectoria del algoritmo, probablemente debido a la distribución balanceada de los datos y la presencia de estructura clara en los grupos.

El descenso coordinado (Figura 8) muestra un comportamiento algo más lento, aunque sigue siendo muy eficaz. Con pesos constantes, se observa un descenso inicial sostenido durante unas 5 iteraciones, pero la convergencia completa ocurre recién cerca de la iteración 12-13. Esto se explica por la naturaleza secuencial del método: al optimizar una coordenada por vez, el ajuste global se da más lentamente. Con pesos decrecientes observamos un comportamiento muy parecido, con una diferencia en la convergencia de unas 3/4 iteraciones nada más.

6.4. Distribucion Lineal

Los puntos se generan a lo largo de una dirección aleatoria en \mathbb{R}^5 , con ruido agregado para simular dispersión. Específicamente, se construye una recta parametrizada y se agregan perturbaciones normales a cada punto. Este tipo de configuración es interesante porque presenta un desafío geométrico: la mayoría de los puntos se encuentran estirados en una misma dirección, lo cual puede afectar la eficiencia de ciertos métodos. Al mismo tiempo, permite observar si los algoritmos logran captar correctamente la estructura global del conjunto, y cómo se comportan al incorporar pesos que varían a lo largo de esa línea.

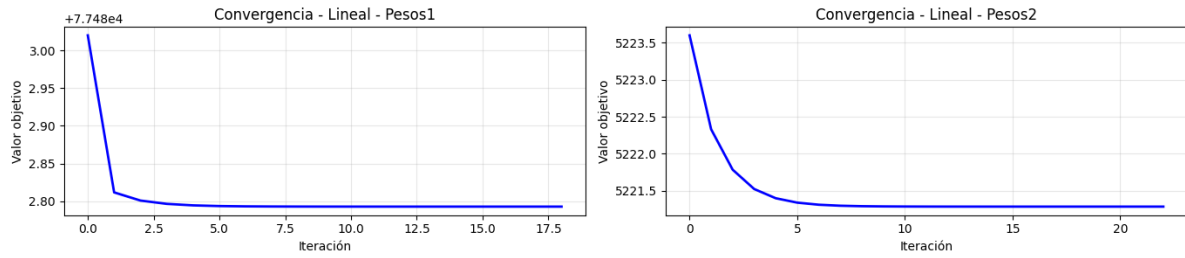


Figura 10: Convergencia lineal con Weiszfeld

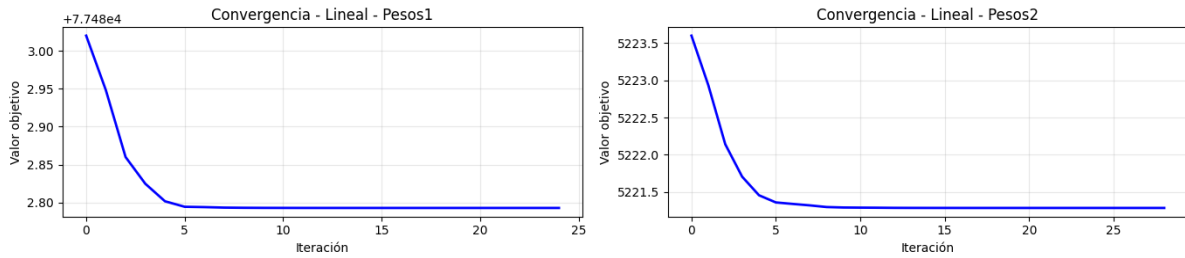


Figura 11: Convergencia lineal con descenso coordinado

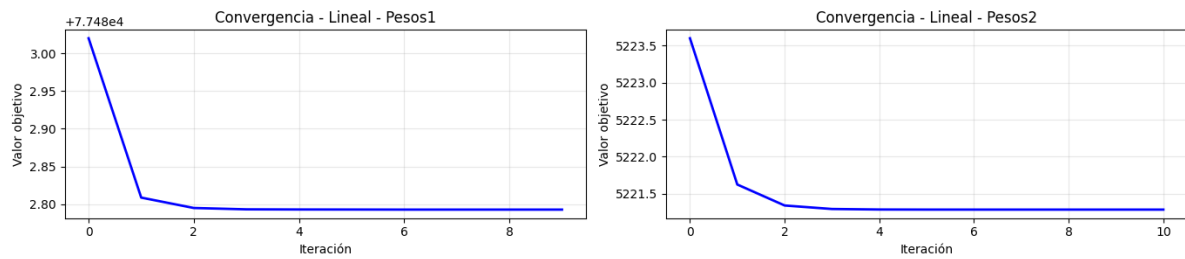


Figura 12: Convergencia lineal con metodo del gradiente

En el caso de datos dispuestos aproximadamente a lo largo de una recta en el espacio, los tres métodos muestran nuevamente un buen desempeño, aunque se evidencian diferencias más marcadas en la velocidad de convergencia y la sensibilidad frente a los pesos.

El método del gradiente (Figura 12) mantiene su superioridad en términos de velocidad de convergencia. Con pesos constantes, alcanza un valor objetivo muy cercano al óptimo en apenas 2-3 iteraciones, y converge completamente hacia la iteración 9. Con pesos decrecientes, el comportamiento es similar: el descenso abrupto en las primeras iteraciones permite alcanzar la zona óptima muy rápidamente. Este rendimiento puede explicarse por el hecho de que, en una distribución lineal, los puntos están más alineados y el gradiente de la función objetivo se alinea con una dirección predominante, lo cual favorece los métodos que explotan esta estructura geométrica como el descenso en la dirección de mayor pendiente.

El método de Weiszfeld (Figura 10), a pesar de ser muy eficiente, muestra una convergencia más progresiva que el gradiente. Esto se debe a que su actualización depende de la posición relativa de todos los puntos y no necesariamente explota la direccionalidad del conjunto como lo hace el gradiente. En una distribución lineal, esta falta de direccionalidad explícita hace que el método tarde más en estabilizar el valor objetivo, aunque el descenso sigue siendo claro y confiable. En este caso, la convergencia del algoritmo es más gradual respecto al caso uniforme o de clusters.

El descenso coordinado (Figura 11) se comporta de forma ligeramente más lenta que los otros métodos, especialmente en los primeros pasos. Con pesos constantes, se observa que necesita unas 5-7 iteraciones para llegar a una solución cercana al óptimo, y converge completamente cerca de la iteración 20-25. Con pesos decrecientes, mejora un poco su ritmo, alcanzando valores muy cercanos al mínimo en unas 6-8 iteraciones. Cuando los pesos decrecen (caso **Pesos2**), se introduce un sesgo direccional más marcado en el gradiente: las coordenadas asociadas a los puntos más influyentes generan una pendiente más pronunciada. Como consecuencia, el algoritmo puede identificar rápidamente la dirección más informativa y

avanzar con mayor efectividad en las primeras iteraciones.

Para esta distribución, los pesos decrecientes no afectan de forma significativa la eficiencia relativa entre métodos, pero sí contribuyen a una leve mejora en el ritmo de convergencia del descenso coordinado y Weiszfeld, al disminuir la influencia de puntos más lejanos a lo largo de la recta.

7. Comparación en diferentes dimensiones

■ Método de Weiszfeld

En la Figura 13 se muestra la cantidad de iteraciones necesarias para que el algoritmo de Weiszfeld converja, en función de la dimensión del espacio, considerando distintos tipos de distribuciones de los puntos: *uniforme*, *lineal*, *clusters* y *outliers*.

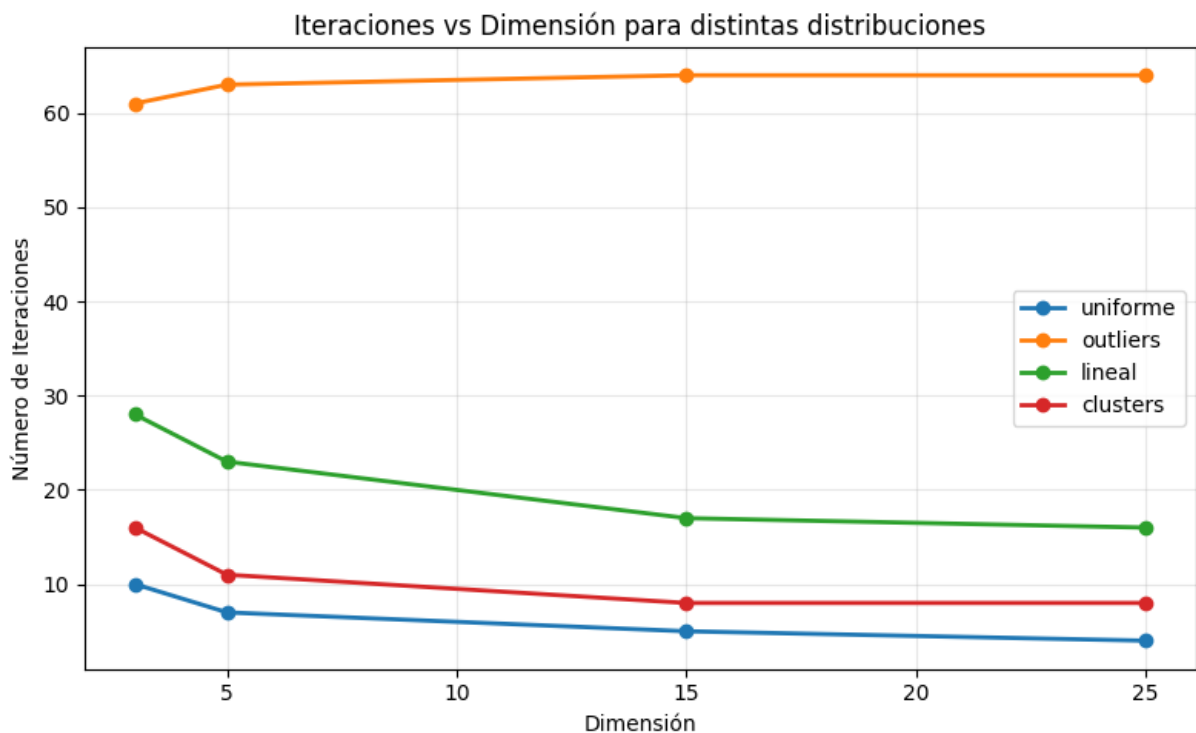


Figura 13: Método de Weiszfeld frente a aumento de la dimensión del problema

Se observa que el comportamiento del algoritmo depende del tipo de distribución de los puntos:

- **Distribución con outliers:** El número de iteraciones se mantiene muy alto y prácticamente constante a medida que aumenta la dimensión. Esto indica que la presencia de valores atípicos introduce inestabilidad en la búsqueda del mínimo, afectando negativamente la eficiencia del algoritmo. La gran dispersión de los outliers en el espacio hace que la dirección de actualización sea menos efectiva y se requieran muchas más iteraciones para estabilizarse.
- **Distribución lineal:** El número de iteraciones disminuye con la dimensión, lo que sugiere que el algoritmo aprovecha la estructura alargada de los datos. En espacios de mayor dimensión, la dirección principal sigue siendo clara, y la dispersión relativa de los puntos se reduce, lo que facilita la convergencia.
- **Distribución uniforme:** Presenta el mejor desempeño en términos de número de iteraciones, con un descenso rápido y suave al aumentar la dimensión. Al no haber estructuras ni agrupamientos marcados, el algoritmo encuentra un equilibrio rápidamente y converge en pocas iteraciones.
- **Distribución en clusters:** Aunque requiere más iteraciones que la uniforme, el número también decrece con la dimensión. La organización de los puntos en grupos bien definidos favorece la estabilidad de la solución, permitiendo una convergencia relativamente rápida.

Este análisis permite concluir que la eficiencia del algoritmo de Weiszfeld depende tanto de la estructura geométrica de los datos como de la dimensión del espacio. En general, cuando los puntos presentan cierta organización (como en clusters) o una forma bien definida (como en la distribución lineal), el algoritmo converge más rápidamente. Por el contrario, la presencia de outliers introduce direcciones inestables y regiones conflictivas en el espacio de búsqueda, lo que perjudica el tiempo de convergencia.

■ Método de descenso coordinado

En la Figura 14 se muestra la evolución del número de iteraciones necesarias para que el método de descenso por coordenadas converja, en función de la dimensión del espacio, y diferenciando entre distintas distribuciones: *uniforme*, *lineal*, *clusters* y *outliers*.

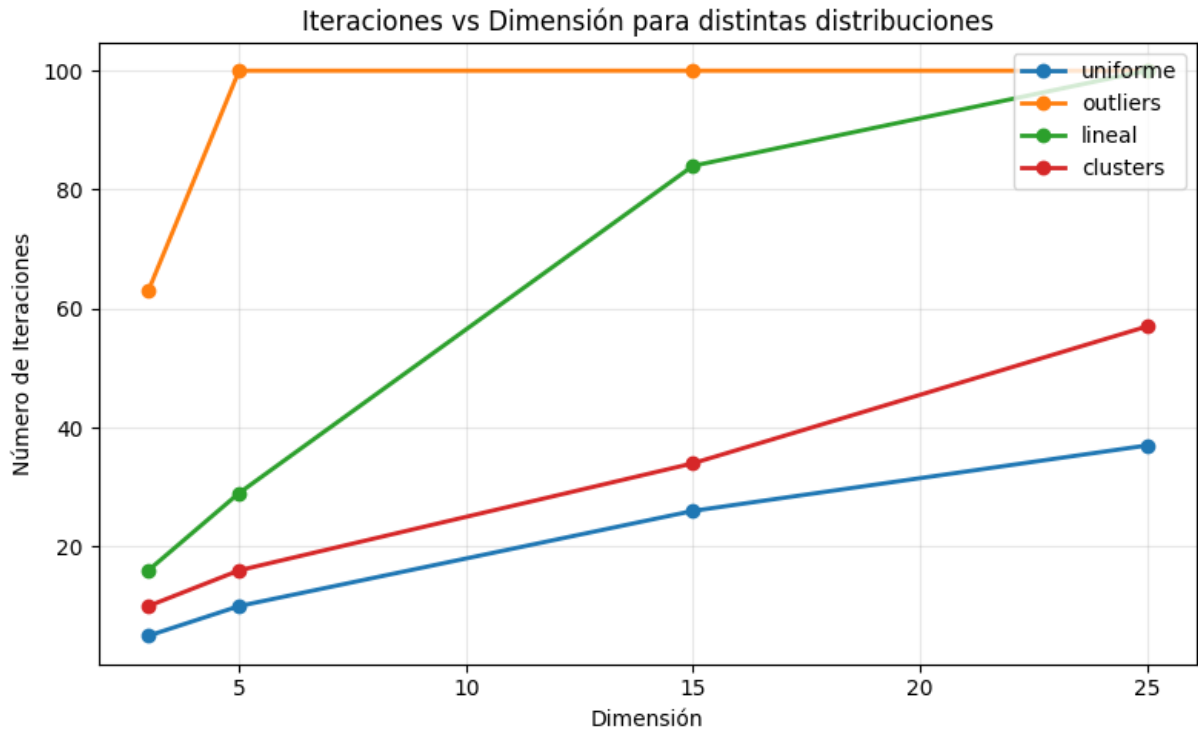


Figura 14: Método de descenso coordinado frente a aumento de la dimensión del problema

A partir del análisis del gráfico, se desprenden las siguientes observaciones:

- **Distribución con outliers:** El método alcanza el máximo de iteraciones permitidas (100) ya desde dimensión 5, lo que evidencia las dificultades que presenta al enfrentarse a puntos extremos. A pesar de usar Gauss-Southwell para seleccionar la coordenada con mayor impacto en el gradiente, la presencia de outliers introduce inestabilidades en la dirección de descenso y vuelve el problema mal condicionado.
- **Distribución lineal:** El número de iteraciones aumenta marcadamente con la dimensión. Esto se debe a que los puntos alineados generan gradientes muy similares entre coordenadas, lo que produce progresos pequeños y lentos al actualizar una sola coordenada por vez. En dimensiones más altas, este efecto se amplifica, afectando la eficiencia del algoritmo.
- **Distribución por clusters:** También presenta una tendencia creciente en el número de iteraciones, aunque de forma más gradual. La existencia de agrupamientos ayuda a guiar el descenso inicialmente, pero a medida que aumenta la dimensión, el espacio se vuelve más complejo y el refinamiento de la solución exige más pasos.
- **Distribución uniforme:** Es el caso más favorable. Aunque el número de iteraciones también crece con la dimensión, el aumento es más suave que en las demás distribuciones. La homogeneidad espacial de los puntos facilita la identificación de direcciones de descenso útiles en cada paso.

En conjunto, estos resultados muestran que el método de descenso coordinado es particularmente sensible a la estructura geométrica de los datos y a la dimensión del problema. Dado que actualiza una coordenada por vez, su rendimiento tiende a degradarse a medida que crece el número de variables.

■ Método de direcciones de descenso

La Figura 15 presenta el número de iteraciones necesarias para la convergencia del método de descenso con dirección, donde se utiliza la dirección del gradiente negativo y el paso se ajusta dinámicamente según la condición de Armijo. Se analiza su comportamiento ante distintas distribuciones de puntos (*uniforme*, *clusters*, *lineal*, *outliers*) y diferentes dimensiones.

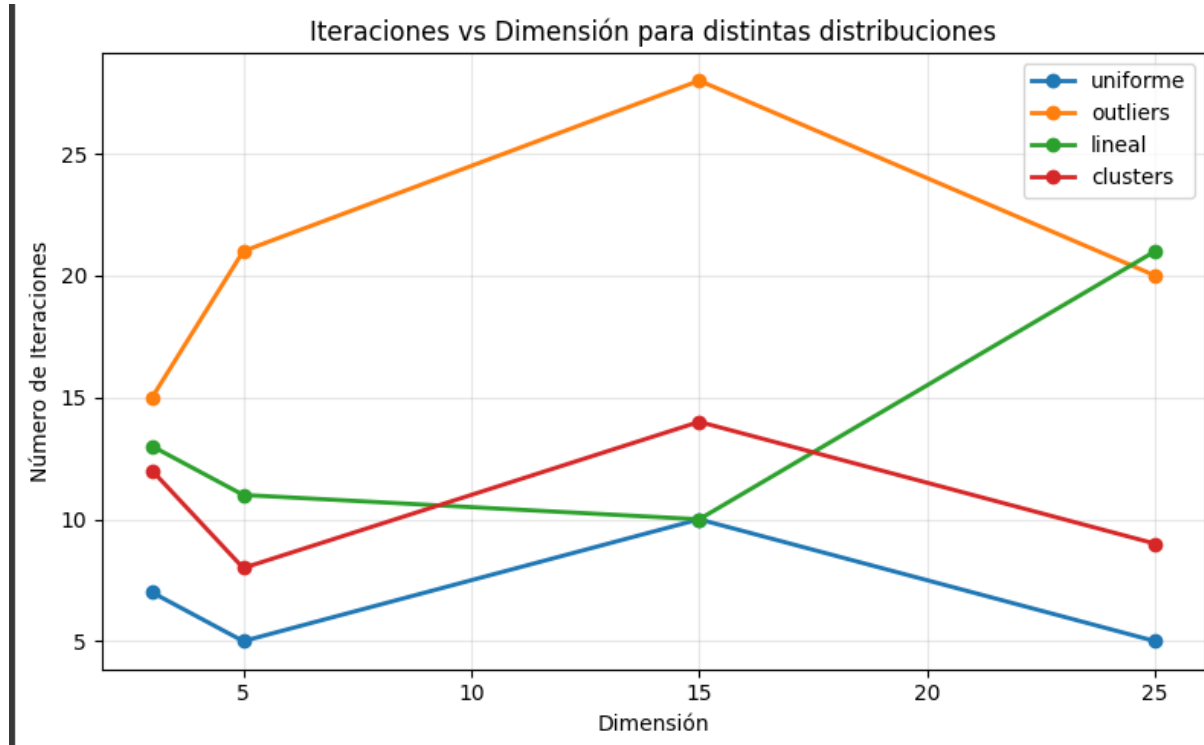


Figura 15: Método de direcciones de descenso frente a aumento de la dimensión del problema

Del análisis del gráfico se destacan las siguientes observaciones:

- **Distribución con outliers:** Aunque presenta la mayor cantidad de iteraciones en comparación con las demás distribuciones, el método muestra una respuesta robusta y controlada. La condición de Armijo permite manejar de forma estable el impacto de los outliers, evitando divergencias o estancamientos que se observaban en otros métodos como Weiszfeld o descenso coordinado.
- **Distribución lineal:** Presenta un comportamiento levemente no monótono, con un mínimo local en dimensión 15, pero en general se mantiene estable, con variaciones moderadas en el número de iteraciones. Esto sugiere que el método logra adaptarse a la orientación alargada de los datos sin dificultad.
- **Distribución en clusters:** Muestra un pico en dimensión 15. Este fenómeno puede explicarse porque en esa dimensión intermedia la dispersión entre los clusters es lo suficientemente grande como para generar múltiples "valles" locales en la función objetivo. Esto puede dificultar el progreso directo del gradiente, y el criterio de Armijo empieza a reducir los pasos para asegurar descenso, aumentando las iteraciones. En dimensiones más altas, las distancias relativas entre puntos se suavizan, y el gradiente puede guiar de nuevo con más eficiencia.
- **Distribución uniforme:** Muestra el mejor comportamiento en cuanto a estabilidad y pocas iteraciones. Como no hay estructuras complejas ni concentraciones de puntos que distorsionen

el gradiente, el método avanza de forma consistente. Las pequeñas oscilaciones entre dimensiones se deben más al ajuste fino del paso que a una dificultad real del problema.

El comportamiento irregular del método del gradiente con Armijo en este gráfico se debe a la interacción entre la estructura de los datos y la forma adaptativa del algoritmo. Cuando la distribución introduce irregularidades (outliers, agrupamientos), el criterio de Armijo ajusta el paso con más cautela, lo que se traduce en más iteraciones en ciertas dimensiones. En cambio, en distribuciones más uniformes o bien comportadas, el gradiente puede actuar con pasos más estables y eficientes.

8. Conclusiones

Los tres enfoques implementados (Weiszfeld, gradiente con criterio de Armijo y descenso coordinado) resuelven el mismo problema, pero lo hacen desde distintas estrategias numéricas. A pesar de apuntar al mismo objetivo, presentan comportamientos de convergencia diferentes, que se explican por la forma en que exploran el espacio de soluciones.

- **Weiszfeld:** Su principal ventaja es la convergencia rápida y estable cuando los puntos no están demasiado cercanos entre sí o sobre el punto actual. El método no necesita ajustar la dirección ni el paso: la actualización es directa, lo que lo vuelve extremadamente eficiente en la práctica. Sin embargo, puede volverse inestable si algún punto coincide con la iteración actual (división por cero) y no es fácilmente generalizable a otras funciones objetivo.
- **Gradiente con criterio de Armijo:** El método del gradiente se destacó por ser el más rápido en alcanzar el valor objetivo en la mayoría de los casos, gracias a que sigue la dirección de máximo descenso y ajusta el paso dinámicamente. En distribuciones uniformes o bien estructuradas, converge muy rápido y en pocas iteraciones, incluso en alta dimensión. Sin embargo, frente a outliers o clusters con pesos variables, su eficiencia puede verse reducida, ya que el criterio de Armijo tiende a reducir excesivamente el tamaño de paso cuando la función tiene regiones planas o abruptas, provocando picos en la cantidad de iteraciones necesarias.
- **Descenso coordinado (Gauss-Southwell):** Este método tuvo un comportamiento más progresivo y, en general, más lento que los otros dos. Es un método simple y con bajo costo computacional por iteración, y puede converger rápidamente si el problema está bien alineado con los ejes. La búsqueda unidimensional en cada dirección permite refinar con precisión, aunque puede necesitar más iteraciones en total. Funciona bien en distribuciones simétricas o cuando los pesos no generan grandes perturbaciones.

Los tres métodos tienden a converger rápidamente cuando los puntos están equilibrados espacialmente (por ejemplo, distribución uniforme con pesos constantes), ya que la solución está cerca del centro geométrico y el paisaje de la función es más regular.

En general, todos los métodos implementados resultaron adecuados para la resolución del problema, siendo el método del gradiente con criterio de Armijo el más veloz en términos de iteraciones, y el más flexible ante estructuras irregulares. El análisis también puso de manifiesto la importancia de la distribución espacial y de los pesos en la determinación de la solución óptima.