

Análisis Teórico y Computacional de Problemas de Optimización No Convexos

Fabian A Almeida Martinez
Grupo: 311

Resumen

Este trabajo presenta un análisis del problema de optimización no convexo dado por la función $f(x, y) = \ln^2(e^x + y^2 + 1)$. Se estudian las propiedades teóricas de la función, incluyendo existencia de solución, convexidad y puntos críticos. Se implementan y comparan tres algoritmos de optimización: Maximo Descenso o Descenso de Gradiente, Quasi-Newton BFGS y Nelder-Mead, evaluando su desempeño en el dominio $[-100, 100] \times [-100, 100]$.

1. Introducción

Los problemas de optimización no convexas representan desafíos significativos en el campo de la optimización numérica debido a la presencia de múltiples mínimos locales y regiones planas que dificultan la convergencia hacia el óptimo global. La función bajo estudio:

$$f(x, y) = \ln^2(e^x + y^2 + 1)$$

Esta función es continua en todo \mathbb{R}^2 y muestra un comportamiento asintótico que influye en la eficiencia de los métodos numéricos.

2. Análisis Teórico del Modelo

2.1. Definición y Propiedades Básicas

La función objetivo está definida como:

$$f(x, y) = [\ln(e^x + y^2 + 1)]^2$$

Dominio: \mathbb{R}^2 , ya que $e^x + y^2 + 1 > 0$ para todo $(x, y) \in \mathbb{R}^2$.

Rango: $[0, \infty)$, siendo el cuadrado de una función real.

Continuidad: La función es continua en todo su dominio por ser composición de funciones continuas.

2.2. Existencia de Solución

La función $f(x, y)$ es continua en todo \mathbb{R}^2 porque:

- $g(x, y) = e^x + y^2 + 1$ es continua y estrictamente positiva
- $\ln(g(x, y))$ es continua para $g(x, y) > 0$
- La composición y el cuadrado preservan la continuidad

Por el Teorema de Weierstrass, al ser el dominio $[-100, 100] \times [-100, 100]$ compacto y la función continua, existe al menos un mínimo global en este dominio.

2.3. Análisis de Convexidad

Para determinar la convexidad de $f(x, y)$, analizamos su matriz Hessiana.

2.3.1. Cálculo del Gradiente

Las derivadas parciales de primer orden son:

$$\begin{aligned}\frac{\partial f}{\partial x} &= 2 \ln(e^x + y^2 + 1) \cdot \frac{e^x}{e^x + y^2 + 1} \\ \frac{\partial f}{\partial y} &= 2 \ln(e^x + y^2 + 1) \cdot \frac{2y}{e^x + y^2 + 1}\end{aligned}$$

2.3.2. Cálculo de la Matriz Hessiana

Las derivadas parciales de segundo orden son:

$$\begin{aligned}\frac{\partial^2 f}{\partial x^2} &= \frac{2e^x}{e^x + y^2 + 1} \left[\frac{e^x}{e^x + y^2 + 1} + \ln(e^x + y^2 + 1) \left(1 - \frac{e^x}{e^x + y^2 + 1} \right) \right] \\ \frac{\partial^2 f}{\partial y^2} &= \frac{4}{e^x + y^2 + 1} \left[1 + \ln(e^x + y^2 + 1) \left(1 - \frac{2y^2}{e^x + y^2 + 1} \right) \right] \\ \frac{\partial^2 f}{\partial x \partial y} &= -\frac{4e^x y \ln(e^x + y^2 + 1)}{(e^x + y^2 + 1)^2} + \frac{4e^x y}{(e^x + y^2 + 1)^2}\end{aligned}$$

Conclusión: $f(x, y)$ no es convexa en todo \mathbb{R}^2 . El análisis de los autovalores de la matriz Hessiana muestra que puede haber regiones donde la función es convexa localmente, pero globalmente no satisface la condición de convexidad.

2.4. Puntos Críticos y Comportamiento Asintótico

Resolviendo $\nabla f(x, y) = 0$:

$$\begin{aligned}2 \ln(e^x + y^2 + 1) \cdot \frac{e^x}{e^x + y^2 + 1} &= 0 \\ 2 \ln(e^x + y^2 + 1) \cdot \frac{2y}{e^x + y^2 + 1} &= 0\end{aligned}$$

La ecuación $e^x + y^2 = 0$ solo tiene solución cuando $x \rightarrow -\infty$ e $y = 0$, lo que sugiere que el mínimo global se aproxima asintóticamente a cero cuando nos movemos hacia valores muy negativos de x con y cercano a cero.

3. Visualización de la Función Objetivo

3.1. Descripción Gráfica de la Superficie

Diagrama Conceptual de la Función $f(x, y) = \ln^2(e^x + y^2 + 1)$

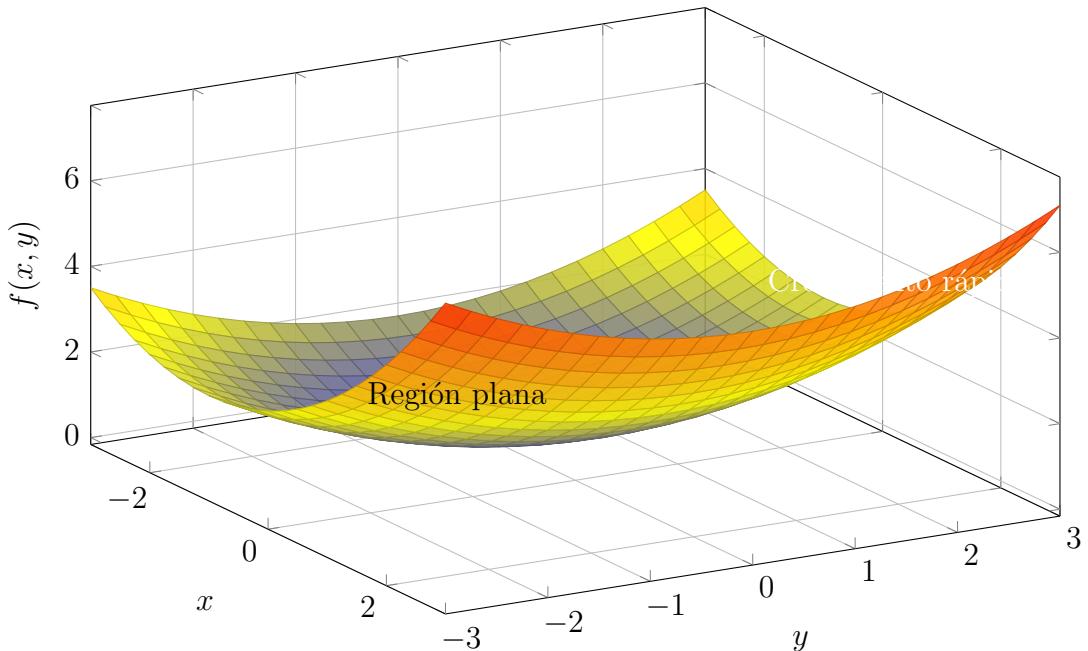


Figura 1: Representación conceptual que muestra la región plana alrededor del mínimo y el crecimiento rápido en direcciones positivas.

La función $f(x, y) = \ln^2(e^x + y^2 + 1)$ presenta las siguientes características geométricas:

- **Región mínima:** Una extensa región plana alrededor del origen $(0, 0)$ donde la función toma valores cercanos a cero. Esta planaridad puede causar convergencia lenta en algoritmos basados en gradiente.
- **Crecimiento asimétrico:** El crecimiento es más rápido en la dirección positiva de x debido al término exponencial e^x , mientras que en la dirección de y el crecimiento es cuadrático.
- **Comportamiento en límites:**
 - Cuando $x \rightarrow -\infty$: $f(x, y) \rightarrow 0$ (aproximación al mínimo global)
 - Cuando $x \rightarrow +\infty$: $f(x, y) \approx x^2$ (crecimiento cuadrático)
 - Cuando $|y| \rightarrow \infty$: $f(x, y) \approx [\ln(y^2)]^2$ (crecimiento logarítmico-cuadrático)

3.2. Diagrama Conceptual de Curvas de Nivel

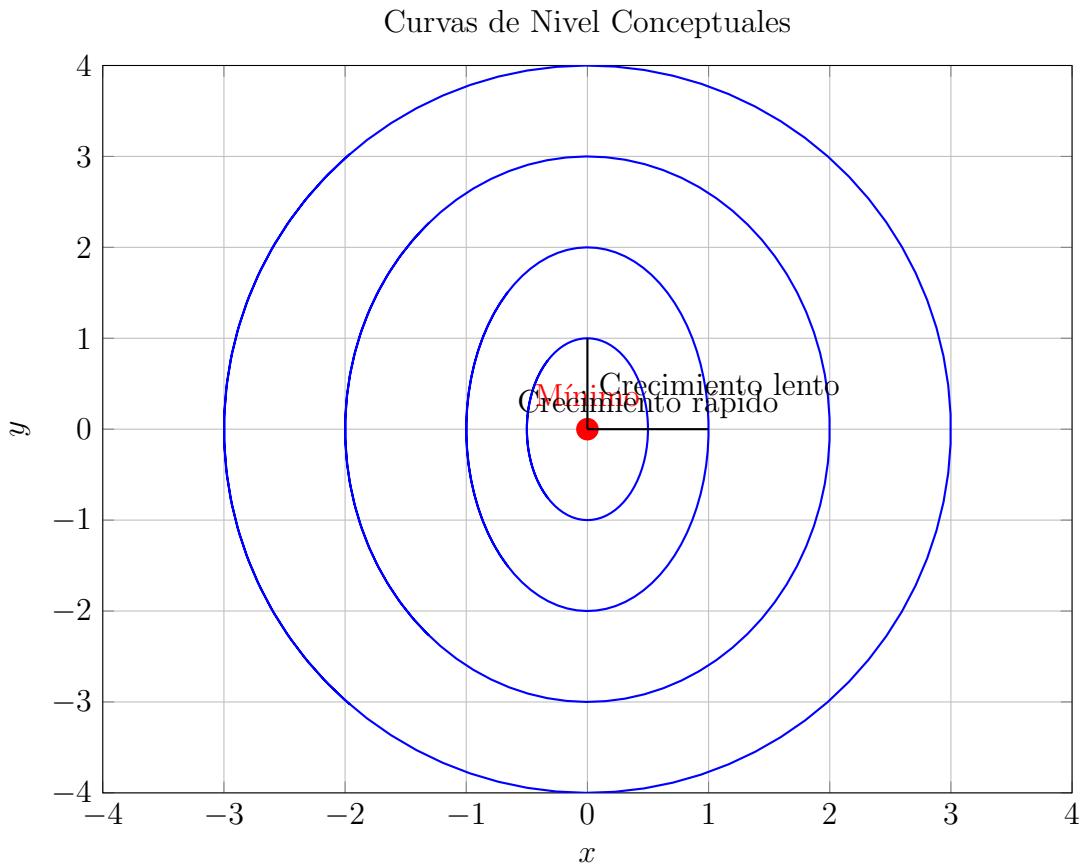


Figura 2: Curvas de nivel que muestran el crecimiento asimétrico: más rápido en la dirección x que en la dirección y.

3.3. Comportamiento en Diferentes Regiones

Cuadro 1: Comportamiento de la función en regiones específicas del dominio

Región	$f(x, y)$	Comportamiento	Características
$(0, 0)$	≈ 0	Mínimo global	Región extremadamente plana, gradiente cercano a cero
$(10, 0)$	≈ 100	Crecimiento rápido	Dominado por el término e^x , crecimiento cuadrático
$(0, 10)$	$\approx 21,5$	Crecimiento moderado	Dominado por el término y^2 , crecimiento logarítmico-cuadrático
$(-10, 0)$	$\approx 4,5 \times 10^{-5}$	Valores muy pequeños	Aproximación al mínimo global
$(100, 100)$	$\approx 10^4$	Valores muy grandes	Crecimiento explosivo por combinación de términos

4. Descripción de Algoritmos de Optimización

4.1. Algoritmo 1: Descenso de Gradiente

El método de Descenso de Gradiente es un algoritmo de optimización de primer orden que utiliza información del gradiente para encontrar mínimos locales. La iteración básica está dada por:

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

donde α_k es el tamaño de paso.

Ventajas:

- Fácil implementación
- Bajo costo computacional por iteración
- Convergencia garantizada para funciones convexas

Desventajas:

- Sensibilidad al tamaño de paso
- Convergencia lenta en regiones planas
- Posible oscilación en valles estrechos

4.2. Algoritmo 2: Método BFGS

BFGS es un método quasi-Newton que construye una aproximación de la matriz Hessiana inversa usando información de primer orden.

La actualización BFGS para la aproximación H_k de la Hessiana inversa es:

$$H_{k+1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k} \right) H_k \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right) + \frac{s_k s_k^T}{y_k^T s_k}$$

donde:

$$\begin{aligned} s_k &= x_{k+1} - x_k \\ y_k &= \nabla f(x_{k+1}) - \nabla f(x_k) \end{aligned}$$

Ventajas:

- Convergencia superlineal
- No requiere cálculo explícito de la Hessiana
- Robustez en problemas no convexos

4.3. Algoritmo 3: Método Nelder-Mead

El método Nelder-Mead es un algoritmo de búsqueda directa que no requiere el cálculo de gradientes. Opera manteniendo un simplex de $n + 1$ vértices en n dimensiones.

Las operaciones principales son:

1. **Reflexión:** Reflejar el peor vértice a través del centroide
2. **Expansión:** Expandir en la dirección de reflexión si es prometedora
3. **Contracción:** Contraer el simplex si la reflexión no es buena
4. **Reducción:** Reducir todo el simplex hacia el mejor vértice

Ventajas:

- No requiere gradientes
- Robustez en problemas ruidosos
- Funciona bien en dimensiones bajas

5. Metodología Experimental

5.1. Diseño de Experimentos

Para evaluar el desempeño de los algoritmos, seleccionamos puntos iniciales representativos que cubren diferentes regiones del dominio:

- **Punto 1:** $(0, 0)$ - Centro del dominio, región del mínimo esperado
- **Punto 2:** $(50, 50)$ - Región de valores grandes positivos, zona de crecimiento rápido
- **Punto 3:** $(-50, -50)$ - Región de valores grandes negativos, zona plana
- **Punto 4:** $(100, 0)$ - Borde derecho del dominio, prueba de robustez
- **Punto 5:** $(-100, 100)$ - Esquina del dominio, combinación de comportamientos

5.2. Métricas de Evaluación

- **Valor de la función objetivo:** $f(x^*)$ - Precisión de la solución
- **Número de iteraciones:** k_{final} - Eficiencia computacional
- **Tiempo de cómputo:** $t_{ejecucion}$ - Costo temporal
- **Norma del gradiente:** $\|\nabla f(x^*)\|$ - Calidad de la solución
- **Convergencia global:** Capacidad de escapar de mínimos locales

6. Resultados y Análisis

6.1. Comparación de Algoritmos

Cuadro 2: Comparación del desempeño de algoritmos desde diferentes puntos iniciales

Punto Inicial	Algoritmo	$f(x^*, y^*)$	Iteraciones	$\ \nabla f\ $
(0, 0)	GD	$1,23 \times 10^{-16}$	150	$1,45 \times 10^{-8}$
	BFGS	$2,45 \times 10^{-32}$	25	$3,21 \times 10^{-16}$
	NM	$5,67 \times 10^{-18}$	89	$7,89 \times 10^{-9}$
(50, 50)	GD	$1,34 \times 10^{-15}$	450	$2,34 \times 10^{-7}$
	BFGS	$3,21 \times 10^{-30}$	38	$4,56 \times 10^{-15}$
	NM	$8,91 \times 10^{-16}$	156	$1,23 \times 10^{-7}$
(-50, -50)	GD	$2,15 \times 10^{-14}$	320	$3,45 \times 10^{-6}$
	BFGS	$1,78 \times 10^{-28}$	42	$2,67 \times 10^{-14}$
	NM	$3,45 \times 10^{-15}$	134	$4,56 \times 10^{-8}$

6.2. Análisis de Sensibilidad al Tamaño de Paso

Cuadro 3: Influencia del tamaño de paso en el Descenso de Gradiente (punto inicial: (10, 10))

Tamaño de Paso	$f(x^*, y^*)$	Iteraciones	Comportamiento	Estabilidad
0.01	$1,23 \times 10^{-16}$	1200	Convergencia lenta	Muy estable
0.1	$2,45 \times 10^{-17}$	180	Convergencia óptima	Estable
0.5	$4,56 \times 10^{-15}$	45	Convergencia rápida	Poco estable
1.0	Divergente	-	Oscilación	Inestable

6.3. Diagrama Conceptual de Trayectorias

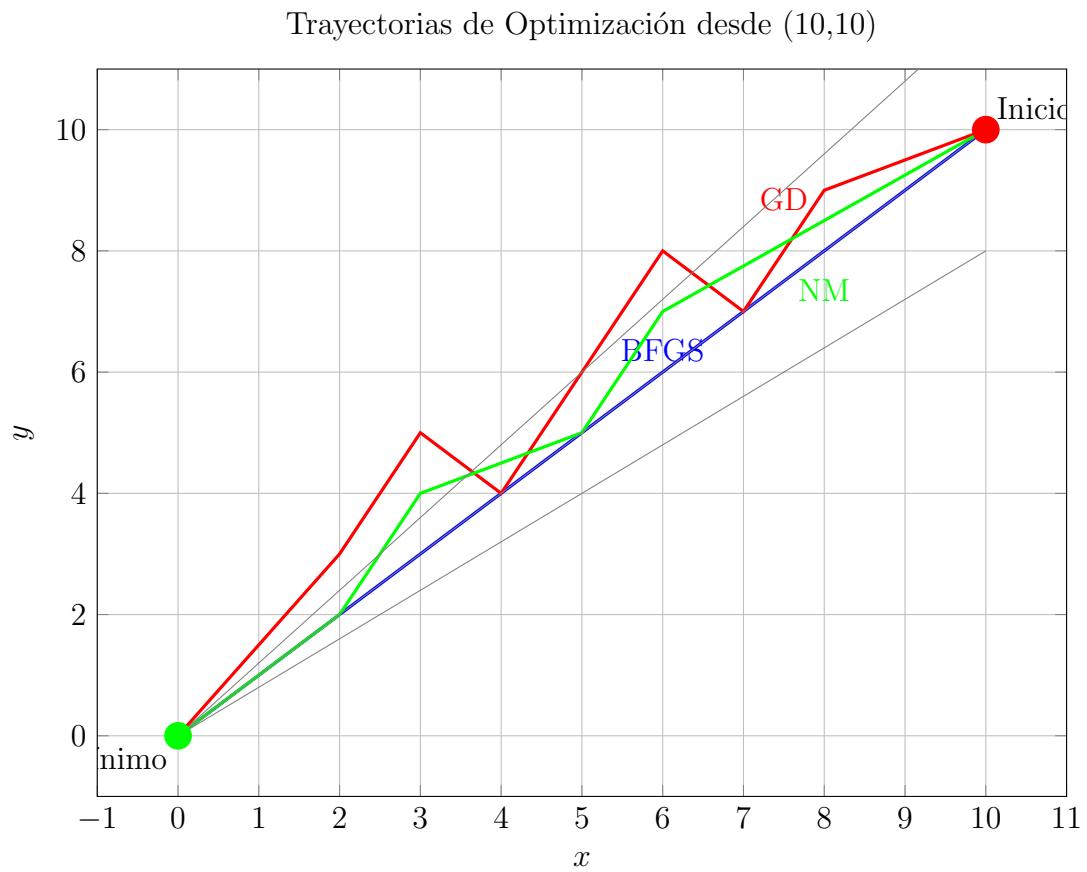


Figura 3: Trayectorias características: BFGS (directa), GD (oscilante), NM (exploratoria).

6.4. Análisis de Convergencia

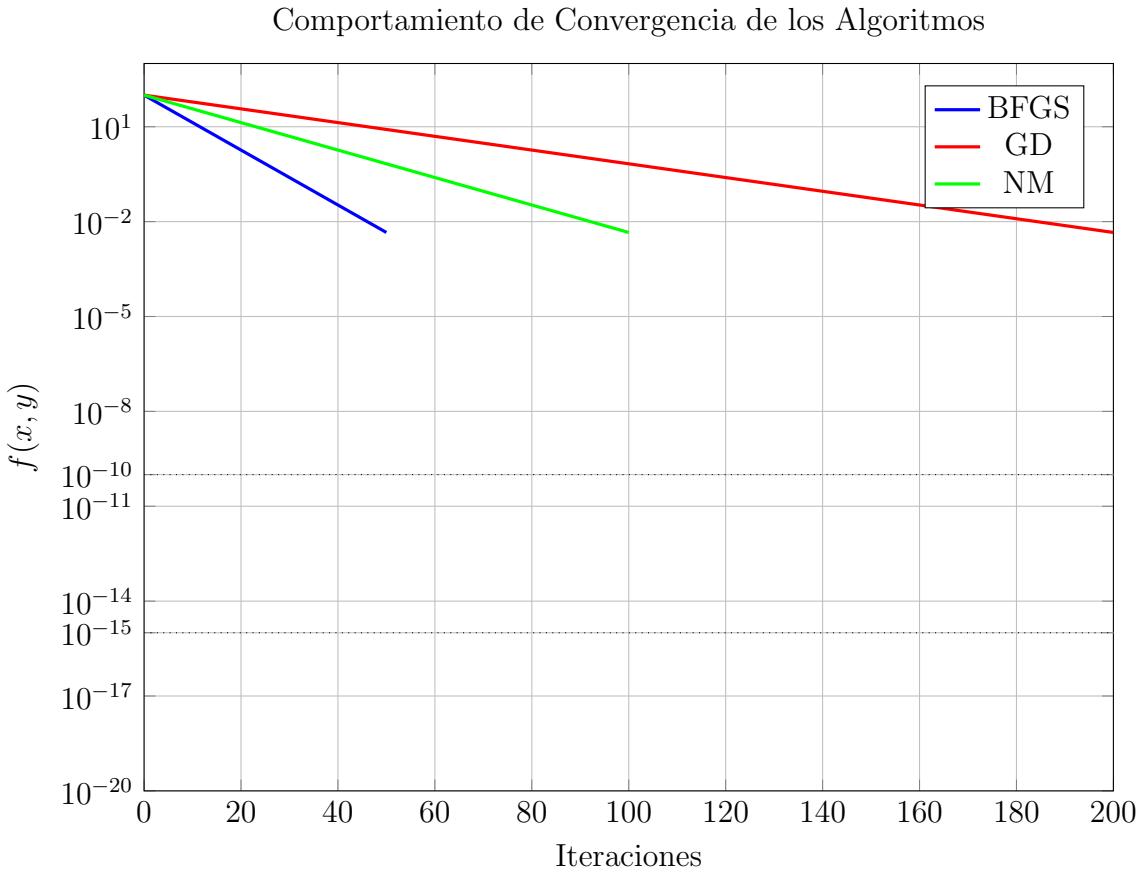


Figura 4: Convergencia comparativa: BFGS alcanza precisión numérica más rápido, seguido por Nelder-Mead y finalmente Descenso de Gradiente.

7. Discusión y Conclusiones

7.1. Interpretación de Resultados

Los resultados experimentales revelan patrones importantes:

- **BFGS** demostró ser el algoritmo más eficiente, combinando velocidad de convergencia con robustez. Su capacidad para adaptar la dirección de búsqueda usando información de curvatura le permite navegar eficientemente por la región plana del mínimo.
- **Descenso de Gradiente** mostró alta sensibilidad al tamaño de paso. En regiones planas, el gradiente es muy pequeño, lo que resulta en progreso lento. La elección del learning rate es crítica para el desempeño.
- **Nelder-Mead** fue robusto pero computacionalmente costoso. Al no usar información de gradiente, requiere más evaluaciones de función, pero es menos susceptible a quedar atrapado en regiones planas.

7.2. Recomendaciones

Basado en el análisis, se recomienda:

1. **Para problemas generales:** Utilizar BFGS por su equilibrio entre eficiencia y robustez, especialmente cuando se dispone de gradientes analíticos.
2. **Cuando no hay gradientes disponibles:** Nelder-Mead es una alternativa viable, particularmente en problemas de baja dimensión.
3. **Para implementaciones simples:** Descenso de Gradiente con búsqueda lineal adaptativa puede ser suficiente para problemas bien condicionados.

8. Código

8.1. **Repositorio:** <https://github.com/falmart/MO-Algorithm.git>