

Proyecto de Modelos de Optimización

C312 Alex Samuel Bas Beovides

1 Análisis de la Función Price 2

La función Price 2 es una función de optimización bidimensional con las siguientes características:

- **Expresión matemática:** $f_{95}(x) = 1 + \sin^2(x_1) + \sin^2(x_2) - 0.1e^{-x_1^2 - x_2^2}$
- **Dominio:** $x_1, x_2 \in [-10, 10]$
- **Mínimo global:** $x^* = (0, 0), f(x^*) = 0.9$

1.1 Propiedades principales

1. **Continuidad:** La función es continua en todo su dominio, sin saltos ni discontinuidades.
2. **Diferenciabilidad:** Es diferenciable en todo su dominio, con un gradiente suave y bien definido.
3. **No separabilidad:** Las variables x_1 y x_2 están interconectadas, lo que impide evaluar sus efectos de manera independiente.
4. **No escalabilidad:** La función está definida únicamente para dos variables, sin una estructura que permita aumentar sus dimensiones.
5. **Multimodalidad:** Presenta múltiples mínimos locales además del mínimo global, lo que complica la optimización.

1.2 Implicaciones para la optimización

- La continuidad y diferenciabilidad permiten el uso de métodos basados en gradientes.
- La no separabilidad aumenta la complejidad del problema de optimización.
- La multimodalidad requiere algoritmos capaces de explorar ampliamente el espacio de búsqueda para evitar quedar atrapados en mínimos locales.
- La naturaleza bidimensional de la función facilita su visualización y análisis, pero limita su aplicabilidad a problemas de mayor dimensionalidad.

2 Análisis de Algoritmos de Optimización

2.1 Gradiente Descendente (BFGS)

El método cuasi-Newton BFGS utiliza aproximaciones de la matriz Hessiana para guiar la búsqueda. Es eficaz en funciones suaves y convexas, ofreciendo una rápida convergencia en problemas bien definidos, con una complejidad de almacenamiento $O(n^2)$. Destaca por su alta eficiencia en número de iteraciones y por no requerir el cálculo explícito de la matriz Hessiana. Sin embargo, puede quedar atrapado en mínimos locales, es sensible a la inicialización y su rendimiento disminuye en problemas de alta dimensión.

Pasos clave:

1. Inicialización con un punto de partida.
2. Actualización iterativa usando el gradiente y la aproximación de la matriz Hessiana.
3. Ajuste de la dirección mediante búsqueda de línea.
4. Actualización de la matriz Hessiana aproximada.

2.2 Método Nelder-Mead (Simplex)

Este algoritmo sin gradiente emplea un simplex geométrico para buscar el mínimo. Es útil para funciones no diferenciables o ruidosas, mostrando robustez en funciones no suaves o con discontinuidades. Su rendimiento disminuye en alta dimensión. Entre sus ventajas se encuentra que no requiere cálculo de gradientes y es relativamente simple de implementar. Sin embargo, presenta una convergencia lenta en problemas complejos, puede quedar atrapado en mínimos locales y es sensible a la geometría del problema y la inicialización.

Pasos clave:

1. Inicialización con un simplex de $n + 1$ puntos.
2. Evaluación de la función en los vértices.
3. Transformaciones del simplex (reflexión, expansión, contracción, reducción).
4. Convergencia hacia el área del mínimo.

2.3 Evolución Diferencial (DE)

La Evolución Diferencial es un algoritmo evolutivo que simula procesos de evolución natural para optimización global. Es eficaz en problemas multimodales y de optimización global, ofreciendo una buena exploración del espacio de búsqueda, aunque requiere mayor tiempo de cómputo que los métodos de gradiente. Destaca por su robustez en búsqueda global y multimodal, y por no requerir información del gradiente. Sus principales desventajas son la necesidad

de ajuste manual de parámetros y una convergencia potencialmente lenta en problemas de alta dimensión.

Pasos clave:

1. Inicialización de población aleatoria.
2. Mutación de vectores.
3. Recombinación de componentes.
4. Selección de mejores soluciones.
5. Repetición hasta convergencia.

2.4 Recocido Simulado (SA)

Inspirado en el proceso de enfriamiento de metales, este algoritmo permite movimientos ocasionales hacia soluciones peores. Es efectivo en evitar mínimos locales y útil en problemas complejos de optimización combinatoria y continua, ofreciendo una convergencia lenta pero robusta. Su flexibilidad para escapar de mínimos locales y su eficacia en funciones multimodales complejas son sus principales ventajas. No obstante, presenta una convergencia lenta y es sensible a la selección de parámetros de enfriamiento.

Pasos clave:

1. Inicialización con solución y temperatura altas.
2. Generación de soluciones vecinas.
3. Evaluación y aceptación probabilística.
4. Enfriamiento gradual.
5. Repetición hasta alcanzar temperatura mínima.

3 Resultados

A continuación, se realizó un análisis comparativo de los distintos algoritmos de optimización aplicados a la función seleccionada. Para cada método se llevaron a cabo 25 ejecuciones, variando los puntos de inicio de manera aleatoria dentro del rango establecido de $[-10, 10]$ para ambas variables x_1 y x_2 . El objetivo fue evaluar el rendimiento en términos de precisión de los resultados y tiempos de ejecución.

Cada ejecución generó una solución que fue evaluada y almacenada, con el fin de calcular las métricas más representativas para cada algoritmo. Estas incluyen la mejor solución obtenida, la peor, la solución promedio, la desviación estándar y el tiempo promedio de ejecución. Los resultados obtenidos permiten observar el comportamiento general de cada método bajo las mismas condiciones

Algoritmo	Mejor Solución	Peor Solución	Solución Promedio	Desviación Estándar	Tiempo Promedio (s)
BFGS	0.9999948274173730	1.000000000269380	0.9999994827340108	0.000001551772098552	0.00146725177764892
Nelder-Mead	0.9000001033675767	1.00000000026851370	0.9949994892257534	0.021794353878813448	0.00115002393722534
DE	0.9000000000005221	1.0000000000171372	0.9350000000024490	0.047696960074022264	0.03936660289764404
SA	0.9000000001871293	1.0000000000000113	0.9050000000037576	0.021794494716843898	0.12230147123336792

Table 1: Resultados de los Algoritmos de Optimización

experimentales. En la tabla se pueden observar las mejores soluciones, peores, la solución promedio, desviación estándar y los tiempos de ejecución asociados.

Además de los valores óptimos encontrados, es importante mencionar los puntos (x_1, x_2) correspondientes a las mejores soluciones de cada algoritmo:

- **BFGS:** (3.14157639e+00, -6.82902587e-09)
- **Nelder-Mead:** (3.92497337e-05, -7.39727159e-06)
- **DE:** (-1.19634508e-09, 2.97068134e-09)
- **SA:** (-4.10723934e-09, -7.59699419e-09)

El método BFGS fue el más rápido con un tiempo promedio de 0.00147 segundos, y un promedio cercano al valor óptimo de 0.999999. Aunque Nelder-Mead obtuvo un mejor valor promedio (0.994999), tuvo una desviación estándar mayor (0.02179), indicando más variabilidad en sus resultados, aunque su tiempo promedio también fue muy rápido (0.00115 segundos). DE (Evolución Diferencial), si bien logró encontrar el mínimo global exacto (0.9), mostró una desviación estándar mayor (0.04769) y fue más lento (0.03937 segundos). Finalmente, Recocido Simulado (SA) también encontró el mínimo global pero con un promedio de 0.905, mayor variabilidad que DE (Std: 0.02179) y el tiempo más alto (0.1223 segundos).