



UNIVERSIDAD AUTÓNOMA DE
CHIAPAS



FACULTAD DE CIENCIAS EN FÍSICA Y
MATEMÁTICAS

Ubicación de almacenes preposicionados e
inventario humanitario

T E S I S

QUE PARA OBTENER EL TÍTULO DE:
LICENCIADA EN MATEMÁTICAS APLICADAS

PRESENTA:
YULISSA DEL ROCIO HERNANDEZ VAZQUEZ

DIRECTOR:
Dr. Yofre Hernán García Gómez

Tuxtla Gutiérrez, Chiapas a November 21, 2025.

Dedicatoria

A mis padres ,
.

Agradecimientos

Tabla de contenidos

2	Introducción	6
3	Fundamentación Teórica y Formulación Matemática del Modelo	8
3.1	Introducción al problema integrado de localización e inventario	8
3.1.1	Relevancia en logística humanitaria y gestión de emergencias . . .	8
3.1.2	Limitaciones de los enfoques clásicos bajo incertidumbre y no linealidad	9
3.1.3	Estructura híbrida: decisiones discretas (ubicación) y continuas (inventario)	12
3.2	Marco general de modelado: variables, parámetros y espacio de decisión .	15
3.2.1	Conjuntos de índices: instalaciones potenciales $\mathcal{I}$ y zonas de demanda $\mathcal{J}$	15
3.2.2	Variables de decisión: binarias (x_i) y continuas (y_{ij})	17
3.2.3	Ejemplo ilustrativo: interpretación y acoplamiento de variables	18
3.2.4	Parámetros del sistema: costos fijos, unitarios, demanda y capacidades	19
3.3	Definición formal del espacio factible $\mathcal{X}$	23
3.4	Definición (Espacio factible).	23
3.4.1	Propiedades estructurales del espacio $\mathcal{X}$	23
3.4.2	Función objetivo general en problemas de localización–inventario .	25
4	Métodos numéricos y optimización con restricciones tipo caja	29
4.1	Fundamentos de métodos iterativos de primer y segundo orden	29
4.1.1	Método de gradiente descendente y sus propiedades de convergencia	29
4.1.2	Métodos cuasi-Newton: aproximación del Hessiano y convergencia superlineal	31
4.1.3	Familia Broyden y el método BFGS como caso prototípico	34
4.2	Extensión a dominios con restricciones tipo caja	36
4.2.1	Proyección ortogonal sobre conjuntos de cotas: definición y propiedades	37
4.2.2	El algoritmo L-BFGS-B: formulación y justificación estructural . .	39
4.2.3	Condiciones de optimalidad bajo cotas: KKT para problemas con restricciones tipo caja	41
4.3	Convergencia del método L-BFGS-B	44
4.3.1	Hipótesis de regularidad: Lipschitz-continuidad del gradiente y compacidad del dominio	44

4.3.2	Teorema de convergencia global: límite de puntos estacionarios que satisfacen KKT	47
4.3.3	Corolario aplicado al modelo propuesto: existencia y unicidad de $Q^* \geq 1$ tal que $\nabla f(Q^*) = 0$	50
4.4	Implementación y análisis numérico	52
4.4.1	Estrategias de inicialización y escalado del inverso del Hessiano aproximado	52
4.4.2	Estimación empírica del número de iteraciones y dependencia del parámetro de memoria m	56
4.4.3	Sensibilidad a las condiciones iniciales y robustez del algoritmo	59
4.4.4	Comparación numérica con otros métodos: gradiente descendente, Newton inexacto y BFGS sin restricciones	62
4.5	Discusión y perspectivas computacionales	65
4.5.1	Ventajas del enfoque de memoria limitada en problemas de gran escala	65
4.5.2	Limitaciones en problemas mal condicionados y alternativas híbridas	68
4.5.3	Posibilidades de extensión a problemas con estructura parcialmente separable	72
5	Implementación y herramientas computacionales	76
5.1	Descripción del entorno computacional	76
5.2	Estructura general del modelo computacional	77
5.3	Implementación del algoritmo <i>L-BFGS-B</i>	77
6	Formulacion del Problema	79
6.1	Supuestos del modelo	79
6.2	Parámetros	79
6.3	Variables de decisión	80
6.4	Función objetivo	80
6.5	Restricciones del modelo	81
6.6	Función de pérdida logística	81
6.7	Modelo de inventario	84
6.8	Fill rate global	86
6.9	Escenario base estado de Veracruz	86
6.9.1	Resultados obtenidos	92
7	Escenario principal estado de Chiapas	94
7.1	Contexto geográfico y socioeconómico del área de estudio	94
7.1.1	Características del municipio de Cacahoatán	94
7.1.2	Vulnerabilidad ante inundaciones	95
7.2	Metodología para el cálculo de pesos posicionales	96
7.2.1	Enfoque multicriterio para la selección de ubicaciones estratégicas	96
7.2.2	Componentes del índice y justificación teórica	96

7.3	Resultados del cálculo de pesos posicionales	97
7.3.1	Ranking de localidades estratégicas	97
7.3.2	Selección de almacenes estratégicos	98
7.4	Diseño de la red logística optimizada	99
7.4.1	Localidades inundables y asignaciones	99
7.4.2	Distribución poblacional por grupos de edad	99
7.5	Resultados de la optimización del sistema	100
7.5.1	Eficiencia del sistema logístico implementado	100
7.5.2	Interpretación y comparación con Veracruz	101
7.5.3	Inventario humanitario optimizado	101
7.5.4	Análisis de la optimización	104
7.6	Validación y análisis de robustez	105
7.6.1	Escenarios de prueba implementados	106
7.6.2	Métricas de desempeño en condiciones estándar	106
7.7	Conclusiones del caso Chiapas	106
7.7.1	Implicaciones para la planeación logística	107
7.7.2	Perspectivas de escalabilidad	107

Referencias

108

1

2 Introducción

Las inundaciones constituyen uno de los fenómenos naturales más frecuentes y devastadores a nivel global, afectando la infraestructura, el acceso a servicios básicos y, sobre todo, la vida y dignidad de las personas. En este contexto, el diseño y aplicación de modelos matemáticos orientados a la logística humanitaria cobra una relevancia crítica, no sólo como una herramienta de gestión operativa, sino como un instrumento de justicia social y resiliencia territorial. Barojas-Payán et al. (2021)

En esa línea, Insani, Widodo, y Rahman (2024) desarrollaron un modelo de programación entera mixta orientado a coordinar simultáneamente la evacuación y la entrega de ayuda en contextos de inundaciones tempranas. Su propuesta incorpora entregas divididas, reutilización de vehículos y múltiples viajes, siendo resuelta mediante un algoritmo genético modificado que alcanzó una eficiencia 92.5 % superior frente a métodos exactos tradicionales.

Por su parte, Sheikholeslami y Zarrinpoor (2022) propusieron un modelo de programación lineal entera mixta multiperíodo bajo condiciones de incertidumbre. Este modelo integra restricciones difusas y estocásticas para optimizar la localización de almacenes, la gestión de inventarios y la provisión de atención médica posterior a los desastres.

En otro enfoque, Romero-Mancilla, Martínez-Flores, y Sánchez-Partida (2024) diseñaron un modelo multimodal que combina transporte terrestre y aéreo mediante drones, estructurado como un modelo multiobjetivo que considera transbordos y múltiples depósitos. Su objetivo principal es equilibrar el costo logístico con los tiempos de entrega, especialmente en escenarios donde la infraestructura vial ha sido severamente afectada.

Asimismo, Santana-Robles, López, y Rivera (2024) formularon un modelo híbrido que combina programación lineal entera con problemas de ruteo vehicular (VRP), enfocado en asignar refugios y optimizar la entrega de suministros bajo variaciones de demanda y recursos limitados.

De forma complementaria, Mashrut y Rahimi (2024) propuso un modelo robusto-fuzzy-probabilístico biobjetivo que busca minimizar tanto el costo operativo como el costo de privación, es decir, el impacto social derivado de la falta de ayuda oportuna.

Finalmente, Pujiana, Utama, y Rahmawati (2020) implementaron un modelo de ruteo multi-depósito (MDVRP) aplicado a la fase posterior a inundaciones, optimizando el uso de depósitos temporales, rutas y cobertura territorial con base en restricciones de capacidad y demanda.

Estos enfoques demuestran que la preparación logística previa al desastre no sólo mejora la eficiencia de la respuesta, sino que permite reducir desigualdades territoriales y proteger de manera diferenciada a las comunidades más vulnerables. La implementación de modelos que integren criterios técnicos (distancia, inventario, costo), operativos (capacidad, transporte) y sociales (accesibilidad, prioridad) es indispensable para enfrentar los desafíos logísticos que imponen los desastres hidrometeorológicos. En el caso del presente modelo, se implementa en el estado de Veracruz como área piloto, con miras a expandirse hacia el estado de Chiapas, dado que comparte vulnerabilidades similares frente a fenómenos hidrometeorológicos, alta dispersión poblacional y limitada infraestructura logística.

Por todo lo anterior, el presente estudio se basa en la construcción de un modelo de optimización entero-mixto no lineal, que articula variables de localización de almacenes preposicionados, gestión de inventarios y niveles de servicio (*fill rate*), con el fin de diseñar una red logística humanitaria eficiente, flexible y ética. Este modelo se inspira en las mejores prácticas de la literatura científica reciente, adaptándolas a condiciones de incertidumbre, alta demanda y restricciones operativas.

3 Fundamentación Teórica y Formulación Matemática del Modelo

3.1 Introducción al problema integrado de localización e inventario

3.1.1 Relevancia en logística humanitaria y gestión de emergencias

La logística humanitaria desempeña un papel fundamental en la respuesta a desastres, al coordinar la disponibilidad, distribución y acceso equitativo a bienes esenciales en entornos caracterizados por alta incertidumbre, infraestructura comprometida y recursos limitados (Balcik & Beamon, 2008; Caunhye et al., 2012). A diferencia de los sistemas logísticos comerciales —donde predominan criterios de eficiencia económica—, los contextos humanitarios priorizan objetivos de **rapidez, cobertura universal y mitigación del riesgo**, lo que exige modelos matemáticos capaces de integrar decisiones estratégicas y tácticas bajo condiciones de información imperfecta.

En este marco, el **problema integrado de localización e inventario** se ha consolidado como una herramienta clave para la planificación anticipada, ya que permite determinar, de manera conjunta:

1. **La ubicación óptima de almacenes de preposicionamiento, y**
2. **Los niveles de inventario que deben almacenarse antes de la ocurrencia de un evento disruptivo.**

Ambas decisiones condicionan el desempeño global del sistema en términos de costo total, robustez operativa y equidad en la atención (Vanajakum & Kachitvichyanukul, 2023). Su naturaleza híbrida —estratégica (ubicación) y táctica (inventario)— impone desafíos estructurales que no pueden abordarse mediante enfoques secuenciales o modelos clásicos aislados.

Aunque la demanda en estos escenarios es inherentemente incierta, en este trabajo se adopta un **enfoque determinista robusto**, ampliamente fundamentado en la teoría de inventarios bajo demanda aleatoria (Hadley & Whitin, 1963). Este enfoque no modela la incertidumbre mediante variables aleatorias ni formulaciones estocásticas, sino que la **incorpora de forma paramétrica** mediante un **stock de seguridad fijo**, derivado de estimaciones de la variabilidad de la demanda.

Específicamente, se asume que la demanda en cada zona $j \in \mathcal{J}$ posee un valor esperado $d_j > 0$ y una desviación estándar $\sigma_j \geq 0$, ambos tratados como **parámetros exógenos conocidos**. Dado un tiempo de reposición $L > 0$ y un nivel de servicio objetivo $\alpha \in (0, 1)$, se define el **factor de seguridad** como $Z = \Phi^{-1}(\alpha)$, donde Φ^{-1} denota la inversa de la función de distribución acumulada de la normal estándar. Aunque esta construcción tiene raíces probabilísticas, en el modelo propuesto **todos los elementos son deterministas**, y el término $Z\sigma\sqrt{L}$ se incorpora como un parámetro de ajuste que refleja la magnitud de la incertidumbre.

Este enfoque permite formular un **modelo de optimización no lineal determinista**, cuya estructura matemática se preserva intacta frente a la complejidad computacional inherente a los modelos estocásticos. La ausencia de esperanzas, escenarios o variables de segundo nivel garantiza trazabilidad analítica, estabilidad numérica y compatibilidad con métodos de optimización eficientes para problemas con cotas, como se discutirá en capítulos posteriores.

3.1.2 Limitaciones de los enfoques clásicos bajo incertidumbre y no linealidad

Los modelos clásicos de localización e inventario —como el problema de localización sin capacidades (UFLP), el modelo de lote económico (EOQ) o sus variantes deterministas— han sido ampliamente estudiados bajo el supuesto de **información perfecta y estructura lineal**, un enfoque cuya pertinencia disminuye notablemente en escenarios humanitarios caracterizados por **incertidumbre estructural y no linealidad inherente** en costos y restricciones (Hadley & Whitin, 1963; Snyder, 2006).

3.1.2.1 Supuestos restrictivos del modelo EOQ clásico

El modelo EOQ se fundamenta en supuestos como:

1. Demanda constante y conocida $D > 0$.
2. Tiempo de entrega determinista $L \geq 0$.
3. Costos lineales: costo de pedido S , costo unitario de mantenimiento H , sin faltantes.
4. Reposición instantánea y sin restricciones de capacidad (Hadley & Whitin, 1963).

El costo total del modelo es:

$$Z^{EOQ}(Q) = \frac{D}{Q}S + \frac{Q}{2}H + DC,$$

y su solución óptima clásica:

$$Q^{EOQ} = \sqrt{\frac{2DS}{H}}.$$

Este resultado depende de la **deterministicidad** y la **convexidad estricta**. Sin embargo, en logística humanitaria:

- La demanda es aleatoria $D(\omega)$.
- L es incierto debido a afectaciones en infraestructura.
- Los faltantes deben modelarse mediante una penalización finita $\beta > 0$.
- Los costos pueden presentar economías de escala y no convexidades (Porteus, 2002).

3.1.2.2 Incapacidad del EOQ para capturar riesgo

Si la demanda durante el tiempo de entrega es $D_L \sim \mathcal{N}(\mu_L, \sigma_L^2)$, el inventario de seguridad requiere:

$$\mathbb{P}(D_L \leq Q/2 + Z_\alpha \sigma_L) \geq \alpha.$$

Con ello, la función de costo extendida se vuelve:

$$Z(Q) = \frac{D}{Q}S + \left(\frac{Q}{2} + Z_\alpha \sigma_L\right)H + DC + \frac{D}{Q}\beta C. \quad (2.1)$$

Esta función **no es convexa globalmente** cuando σ_L depende de Q , como ocurre cuando existe congestión en transporte o variabilidad dependiente del tamaño del pedido (Bertsimas & Thiele, 2006; Shapiro et al., 2021).

Proposición (No linealidad estricta).

La función $Z(Q)$ definida en (2.1) es diferenciable en $(0, \infty)$, pero no es convexa globalmente si $\beta > 0$ y $\sigma_L > 0$.

Si σ_L es constante:

$$\frac{d^2 Z}{dQ^2} = \frac{2D(S + \beta C)}{Q^3} > 0.$$

Pero si $\sigma_L = \sigma\sqrt{L(Q)}$ y $L(Q)$ es no lineal, entonces $Z(Q)$ pierde convexidad.

Esto demuestra que muchos métodos clásicos basados en convexidad y separabilidad resultan inaplicables en condiciones realistas.

3.1.2.3 Limitaciones de modelos de localización deterministas

El modelo clásico UFLP resuelve:

$$\min_{x \in \{0,1\}^m, y \geq 0} \sum_i f_i x_i + \sum_i \sum_j c_{ij} y_{ij},$$

bajo restricciones de asignación deterministas. Sin embargo:

- La demanda d_j es aleatoria.
- Los costos pueden variar según accesibilidad o daños.
- Las capacidades reales son inciertas.

Modelar d_j mediante $\mathbb{E}[d_j]$ genera soluciones frágiles frente a escenarios adversos (Snyder, 2006).

3.1.2.4 Ejemplo ilustrativo: EOQ clásico vs. EOQ extendido con stock de seguridad y penalización

Considérese un contexto en el que la demanda esperada es $D = 1000$ unidades, con una variabilidad estimada cuya desviación estándar es $\sigma = 10$. Aunque la demanda real puede fluctuar (por ejemplo, entre 800 y 1200 unidades en eventos históricos), en este enfoque se modela la incertidumbre mediante parámetros fijos derivados de su comportamiento estadístico. Los demás parámetros son:

- Costo de pedido: $S = 100$
- Costo unitario de mantenimiento: $H = 5$
- Costo unitario de adquisición: $C = 10$
- Penalización por faltante: $\beta = 50$
- Tiempo de reposición: $L = 0.1$
- Nivel de servicio objetivo: $95\% \Rightarrow Z_\alpha = 1.645$

El stock de seguridad se calcula como:

$$SS = Z_\alpha \sigma \sqrt{L} \approx 1.645 \cdot 10 \cdot \sqrt{0.1} \approx 5.20,$$

aunque en la función de costo extendida, este valor se incorpora como un término constante.

3.1.2.5 EOQ clásico (ignora incertidumbre y faltantes)

$$Q_{EOQ} = \sqrt{\frac{2DS}{H}} = \sqrt{\frac{2(1000)(100)}{5}} = 200,$$

y su costo asociado es:

$$Z_{EOQ} = \frac{200}{1000} \cdot 100 + \frac{2}{200} \cdot 5 + 1000 \cdot 10 = 500 + 500 + 10000 = 11000.$$

3.1.2.6 EOQ extendido (con stock de seguridad y penalización por faltantes)

La función de costo adopta la forma:

$$Z(Q) = \frac{Q}{D}(S + \beta C) + \left(\frac{2}{Q} + Z_\alpha \sigma \sqrt{L} \right) H + DC.$$

Sustituyendo los valores:

$$Z(Q) = \frac{Q}{1000}(100 + 500) + \left(\frac{2}{Q} + 5.20 \right) \cdot 5 + 10000 = \frac{Q}{600000} + 2.5Q + 10026.$$

Minimizando esta función determinista:

$$Q^* = \sqrt{\frac{2.5}{600000}} \approx 489.9, \quad Z(Q^*) \approx 12475.44.$$

Si, por error, se usara la política del EOQ clásico ($Q = 200$):

$$Z(200) = \frac{200}{600000} + 2.5 \cdot 200 + 10026 = 3000 + 500 + 10026 = 13526,$$

lo que representa un **8.4% más de costo** respecto a la política óptima del modelo extendido, además de una **exposición significativamente mayor al riesgo de faltante**.

Este contraste demuestra que los modelos deterministas que ignoran la variabilidad de la demanda y los costos de faltante pueden producir políticas subóptimas o inadmisibles en contextos humanitarios, donde la insatisfacción de la demanda tiene consecuencias operativas y éticas relevantes.

3.1.3 Estructura híbrida: decisiones discretas (ubicación) y continuas (inventario)

Los problemas integrados de localización e inventario forman parte de una clase fundamental denominada *modelos híbridos* o *problemas de decisión mixta*, en los que coexisten variables discretas y continuas. Este tipo de modelos presenta una complejidad inherente debida a la ruptura topológica inducida por las variables binarias, lo cual conlleva dominios no convexos y no conexos, impidiendo la aplicación directa de técnicas clásicas basadas en convexidad (Wolsey, 1998; Grossmann, 2002; Daskin, 2013).

3.1.3.1 Formulación matemática del espacio de decisiones

Sea: - $\mathcal{I} = \{1, \dots, m\}$ el conjunto de ubicaciones posibles para almacenes, - $\mathcal{J} = \{1, \dots, n\}$ las zonas de demanda, - $x = (x_1, \dots, x_m)^\top \in \{0, 1\}^m$ las decisiones binarias de apertura, - $y = (y_{ij}) \in \mathbb{R}_+^{m \times n}$ los flujos desde almacenes a zonas, - $s = (s_1, \dots, s_m)^\top \in \mathbb{R}_+^m$ el inventario preposicionado.

El espacio factible se define como:

$$\mathcal{X} := \left\{ (x, y, s) \in \{0, 1\}^m \times \mathbb{R}_+^{m \times n} \times \mathbb{R}_+^m \left| \begin{array}{l} \sum_{i \in \mathcal{I}} y_{ij} \geq d_j, \quad \forall j \in \mathcal{J}, \\ \sum_{j \in \mathcal{J}} y_{ij} \leq s_i, \quad \forall i \in \mathcal{I}, \\ s_i \leq Mx_i, \quad \forall i \in \mathcal{I} \end{array} \right. \right\}. \quad (2.2)$$

Aquí, $d_j > 0$ representa la demanda de la zona j y M es un parámetro de gran magnitud. La tercera restricción describe un vínculo lógico: si $x_i = 0$, entonces $s_i = 0$ y, en consecuencia, $y_{ij} = 0$. Este acoplamiento induce una **no convexidad estructural** (Bertsekas, 1999).

Proposición (No convexidad).

El conjunto $\mathcal{X}$ definido en (2.2) **no es convexo**.

Demostración. Considérense las configuraciones factibles $x^{(1)} = (1, 0, \dots, 0)$ y $x^{(2)} = (0, 1, 0, \dots, 0)$.

El punto medio $\frac{1}{2}x^{(1)} + \frac{1}{2}x^{(2)}$ tiene componentes iguales a $1/2$, lo cual viola la condición binaria. Luego, $\mathcal{X}$ no es convexo.

La consecuencia directa es que condiciones de optimalidad de tipo KKT **no garantizan** optimalidad global (Lewis & Overton, 2013).

3.1.3.2 Función objetivo y acoplamiento estructural

El costo total del sistema se modela como:

$$Z(x, y, s) = \underbrace{\sum_{i \in \mathcal{I}} f_i(x_i)}_{\text{Costos fijos}} + \underbrace{\sum_{i \in \mathcal{I}} h_i s_i}_{\text{Costos de inventario}} + \underbrace{\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} y_{ij}}_{\text{Costos de transporte}}. \quad (2.3)$$

Aunque la parte en variables continuas es lineal, el término $f_i(x_i)$ genera una **no linealidad discreta**; más aún, en la práctica humanitaria pueden existir economías de escala, accesibilidad variable o costos de apertura no lineales, reforzando la complejidad global (Grossmann, 2002).

El sistema presenta una jerarquía:

1. x determina la infraestructura disponible,
2. s depende de x ,
3. y depende de s .

Esto produce una dependencia en cascada que elimina la separabilidad global del problema.

Proposición (Convexidad condicional).

Para cualquier $\bar{x} \in \{0, 1\}^m$, el subproblema

$$\min_{(y,s) \in \mathcal{X}(\bar{x})} Z(\bar{x}, y, s)$$

es un programa lineal y, por tanto, convexo.

Demostración. Fijado $\bar{x}$, todas las restricciones son lineales en (y, s) y la función objetivo es afín.

Esta propiedad respalda el uso de descomposición tipo Benders o esquemas maestro-subproblema (Wolsey, 1998).

3.1.3.3 Ejemplo ilustrativo: interacción discreto-continuo

- Considérese: - $m = 2$ almacenes,
 - $n = 2$ demandas con $d_1 = d_2 = 50$,
 - Costos fijos $f_1 = 300$, $f_2 = 400$,
 - Costos de inventario $h_1 = h_2 = 3$,
 - Costos de transporte:
 $c_{11} = 8$, $c_{12} = 15$,
 $c_{21} = 12$, $c_{22} = 7$,
 - $M = 100$.

(i) Abrir solo 1:

$$Z = 300 + 3 \cdot 100 + (8 \cdot 50 + 15 \cdot 50) = 1750.$$

(ii) Abrir solo 2:

$$Z = 400 + 3 \cdot 100 + (12 \cdot 50 + 7 \cdot 50) = 1650.$$

(iii) Abrir ambos:

Asignación eficiente, pero mayor costo fijo:

$$Z = 300 + 400 + 3 \cdot 100 + (8 \cdot 50 + 7 \cdot 50) = 1750.$$

Conclusión: es óptimo **abrir solo el almacén 2**.

La solución cambiaría si los costos fijos disminuyeran, lo que evidencia la *sensibilidad estructural* del diseño híbrido (Daskin, 2013).

La integración simultánea de decisiones discretas y continuas induce dominios no convexos, dependencias lógicas y estructuras no separables que requieren marcos teóricos propios de la programación entera mixta, análisis no suave y descomposición matemática. Esta base establece los fundamentos del modelo propuesto en las siguientes secciones, donde se formaliza la integración bajo condiciones logísticas realistas y restricciones humanitarias.

3.2 Marco general de modelado: variables, parámetros y espacio de decisión

3.2.1 Conjuntos de índices: instalaciones potenciales $\mathcal{J}$ y zonas de demanda

La construcción rigurosa de un modelo de optimización comienza con la definición precisa de los conjuntos de índices, los cuales encapsulan la estructura topológica y semántica del sistema bajo estudio. En el contexto del problema integrado de localización e inventario para logística humanitaria, dos conjuntos indexados fundamentales determinan la arquitectura del modelo: el conjunto de instalaciones potenciales y el conjunto de zonas de demanda.

3.2.1.1 Definición (Conjunto de instalaciones potenciales).

Sea

$$\mathcal{J} := \{1, 2, \dots, m\}, \quad m \in \mathbb{N}, \quad m \geq 1,$$

el conjunto finito y numerable de ubicaciones candidatas para la apertura de almacenes de preposicionamiento. Cada índice $i \in \mathcal{J}$ representa una localización geográfica factible donde podría instalarse una instalación de almacenamiento y distribución.

3.2.1.2 Definición (Conjunto de zonas de demanda).

Sea

$$\mathcal{J} := \{1, 2, \dots, n\}, \quad n \in \mathbb{N}, \quad n \geq 1,$$

el conjunto finito y numerable de zonas afectadas o potenciales de demanda. Cada índice $j \in \mathcal{J}$ denota una región geográfica donde se anticipa la necesidad de suministros humanitarios tras un evento disruptivo.

Observación

Los conjuntos $\mathcal{I}$ y $\mathcal{J}$ se asumen disjuntos en función, aunque no necesariamente en localización física. Desde el punto de vista del modelo, los roles son distintos:

- Los elementos de $\mathcal{I}$ son fuentes de suministro.
- Los elementos de $\mathcal{J}$ son sumideros de demanda.

La cardinalidad de estos conjuntos —es decir, $m = |\mathcal{I}|$ y $n = |\mathcal{J}|$ — determina directamente la complejidad combinatoria del problema. En particular, el número de configuraciones posibles de apertura de almacenes es 2^m , lo que implica que incluso para valores modestos (por ejemplo, $m = 30$), el espacio discreto es del orden de 10^9 . Esta explosión combinatoria subraya la necesidad de métodos de optimización estructurada.

3.2.1.3 Ejemplo ilustrativo

Supóngase un escenario de planificación para respuesta a inundaciones dividido en cinco distritos. Se identifican tres ubicaciones potenciales para almacenes:

- $i = 1$: Aeropuerto internacional,
- $i = 2$: Base militar,
- $i = 3$: Centro de distribución de una ONG.

Asimismo, se consideran cuatro zonas vulnerables:

- $j = 1$: Zona ribereña,
- $j = 2$: Área rural,
- $j = 3$: Barrio urbano denso,
- $j = 4$: Comunidad indígena aislada.

Los conjuntos índice resultan:

$$\mathcal{I} = \{1, 2, 3\}, \quad \mathcal{J} = \{1, 2, 3, 4\}.$$

Sobre estos conjuntos se definirán:

- variables binarias $x_i \in \{0, 1\}$ para cada $i \in \mathcal{I}$,
- variables continuas de flujo $y_{ij} \geq 0$ para $(i, j) \in \mathcal{I} \times \mathcal{J}$,
- parámetros exógenos como c_{ij} (costos), f_i (costos fijos) y d_j (demanda).

Nota metodológica.

En modelado riguroso, los conjuntos índice deben definirse antes que las variables, dado que estas heredan su dominio directamente de aquellos.

Los conjuntos $\mathcal{I}$ y $\mathcal{J}$ constituyen estructuras matemáticas esenciales que determinan la escala, la conectividad y la complejidad del modelo. Su correcta definición resulta crucial para la formalización posterior de variables, restricciones y dependencias del sistema.

3.2.2 Variables de decisión: binarias (x_i) y continuas (y_{ij})

Una vez establecida la estructura combinatoria del problema mediante los conjuntos índice $\mathcal{I}$ y $\mathcal{J}$, el siguiente paso en la formulación rigurosa de un modelo de optimización es la definición precisa del **espacio de decisiones**, mediante la especificación de las **variables de decisión**. En el problema integrado de localización e inventario, este espacio es híbrido, compuesto por dos tipos cualitativamente distintos de variables:

- Variables binarias, que modelan elecciones discretas de infraestructura,
- Variables continuas, que representan flujos físicos de recursos.

Esta dualidad refleja la naturaleza multinivel del problema: decisiones estratégicas (¿dónde construir?) y decisiones tácticas u operativas (¿cuánto enviar?).

3.2.2.1 Definición (Variables binarias de localización).

Para cada $i \in \mathcal{I}$, se define la variable binaria

$$x_i \in \{0, 1\},$$

donde

$$x_i = \begin{cases} 1, & \text{si se decide abrir una instalación en la ubicación } i, \\ 0, & \text{en caso contrario.} \end{cases}$$

El vector $x = (x_i)_{i \in \mathcal{I}} \in \{0, 1\}^m$ codifica una **configuración de infraestructura** del sistema logístico. Estas variables pertenecen a un espacio discreto no convexo y finito cuya cardinalidad es 2^m .

3.2.2.2 Definición (Variables continuas de flujo).

Para cada par $(i, j) \in \mathcal{I} \times \mathcal{J}$, se define la variable continua

$$y_{ij} \in \mathbb{R}_+,$$

que representa la cantidad de unidades asignadas desde la instalación i a la zona de demanda j en el período de planificación.

El tensor $y = (y_{ij})_{(i,j) \in \mathcal{I} \times \mathcal{J}} \in \mathbb{R}_+^{m \times n}$ describe la estructura de asignación del sistema, y pertenece a un espacio vectorial convexo, cerrado y de dimensión finita.

Observación (Naturaleza híbrida del espacio de decisiones).

El espacio total de decisiones es el producto cartesiano

$$\mathcal{D} = \{0, 1\}^m \times \mathbb{R}_+^{m \times n},$$

que es no convexo, no conexo y discontinuo en la dirección discreta. Esta estructura impide la aplicación directa de herramientas del análisis convexo clásico y exige métodos de optimización mixta.

Las variables x_i y y_{ij} no son independientes. Su acoplamiento se establece mediante **restricciones de factibilidad**, que garantizan que no se asignen flujos desde instalaciones no abiertas. Para todo $i \in \mathcal{I}$ y $j \in \mathcal{J}$:

$$y_{ij} \leq Mx_i, \tag{2.4}$$

donde $M > 0$ es una constante suficientemente grande (por ejemplo, $M = \sum_{j \in \mathcal{J}} d_j$). La restricción (2.4) es una formulación *big-M* estándar que modela la implicación lógica:

$$x_i = 0 \Rightarrow y_{ij} = 0 \quad \forall j \in \mathcal{J}.$$

3.2.3 Ejemplo ilustrativo: interpretación y acoplamiento de variables

Sea:

$$\mathcal{I} = \{1, 2, 3\}, \quad \mathcal{J} = \{1, 2, 3, 4\}.$$

Supóngase que se decide abrir instalaciones en $i = 1$ y $i = 3$, pero no en $i = 2$:

$$x = (1, 0, 1).$$

Las restricciones big-M implican:

- $y_{2j} = 0$ para todo j ,
- $y_{1j}, y_{3j} \geq 0$ como flujos posibles.

Una asignación factible es:

$$\begin{aligned} y_{11} &= 30, & y_{12} &= 20, & y_{13} &= 0, & y_{14} &= 0, \\ y_{21} &= y_{22} = y_{23} = y_{24} = 0, \\ y_{31} &= 10, & y_{32} &= 0, & y_{33} &= 40, & y_{34} &= 25. \end{aligned}$$

Si la demanda es $d = (40, 20, 40, 25)$, esta asignación es factible y completa.

Este ejemplo muestra que las variables binarias **activan o desactivan** subespacios del espacio continuo de flujos, generando una partición de $\mathcal{D}$ en 2^m subespacios convexos.

Las variables x_i y y_{ij} constituyen los grados de libertad fundamentales del modelo. Su naturaleza híbrida —discreta y continua— refleja la dualidad entre planeación estratégica y operación táctica. El acoplamiento lógico mediante restricciones big-M introduce no convexidad, justificando el uso de técnicas de optimización entera y métodos especializados. Las siguientes secciones integrarán estas variables en la función objetivo y en las restricciones de balance de flujo.

3.2.4 Parámetros del sistema: costos fijos, unitarios, demanda y capacidades

En la formulación matemática de un problema de optimización, los **parámetros** constituyen los datos exógenos que estructuran el espacio factible y la función objetivo. Su correcta especificación es requisito previo a la definición de variables y restricciones; por tanto, deben presentarse con dominio y unidades explícitas (práctica recomendada por Lewis y colaboradores; Lewis, 2003). En el problema integrado de localización e inventario consideramos, de modo explícito, las cuatro familias de parámetros siguientes.

3.2.4.1 Costos fijos de apertura

Para cada $i \in \mathcal{I}$ definimos

$$f_i \in \mathbb{R}_+$$

como el **costo fijo de apertura** de la instalación en la localización i . En la formulación estándar el término contribuye a la función objetivo mediante $f_i x_i$ si x_i es la variable binaria de apertura. Se exige

$$f_i \geq 0, \quad \forall i \in \mathcal{I}.$$

Observación práctica: cuando la apertura implica economías o penalizaciones no triviales (por ejemplo, coste creciente con la capacidad instalada o costes discretos adicionales por remoción), f_i puede modelarse como función $f_i(x_i, s_i)$; sin embargo, para la formulación

base se adopta linealidad en x_i y se externalizan las no linealidades en extensiones (Shapiro et al., 2021).

3.2.4.2 Costos unitarios de transporte

Para cada par $(i, j) \in \mathcal{I} \times \mathcal{J}$ definimos

$$c_{ij} \in \mathbb{R}_+,$$

el **costo unitario de transportar** una unidad desde la instalación i hasta la zona de demanda j . Requerimos

$$c_{ij} \geq 0, \quad \forall i, j,$$

y denotamos la matriz $C := [c_{ij}]_{i,j} \in \mathbb{R}_+^{m \times n}$. En contextos humanitarios c_{ij} incorpora distancia, tiempo de acceso y penalizaciones por rutas inseguras; su estimación suele derivar de información geoespacial y modelos de capacidad vial (Daskin, 2013).

3.2.4.3 Demanda esperada

Para cada $j \in \mathcal{J}$, sea

$$d_j \in \mathbb{R}_{++}$$

la **demanda esperada** en la zona j durante el horizonte considerado. Adicionalmente definimos el vector $d = (d_j)_{j \in \mathcal{J}}$. En la formulación determinista de referencia usamos d_j como estimador puntual (p. ej. esperanza), con la advertencia explícita de que su uso directo puede producir soluciones frágiles en colas de distribución (Snyder, 2006). Notación:

$$d_j > 0, \quad \forall j, \quad d \in \mathbb{R}_{++}^n.$$

3.2.4.4 Capacidades de almacenamiento

Para cada $i \in \mathcal{I}$ definimos la **capacidad máxima** $s_i^{\max}$ mediante

$$s_i^{\max} \in \mathbb{R}_+ \cup \{+\infty\}.$$

La restricción de capacidad se escribe, asociada a la lógica de activación x_i :

$$\sum_{j \in \mathcal{J}} y_{ij} \leq s_i^{\max} x_i, \quad \forall i \in \mathcal{I}. \quad (2.5)$$

En ausencia de un tope operativo se toma $s_i^{\max} = +\infty$ (modelo no capacitado); en práctica real $s_i^{\max}$ suele venir de disponibilidad física, limitaciones regulatorias o logísticas.

3.2.4.5 Propiedades y observaciones formales

1. **Dominios y no ambigüedad:** Todos los parámetros deben venir acompañados de su dominio y unidades. Esto evita ambigüedades en pruebas de existencia, estabilidad numérica y escalado de variables (Lewis, 2003).
2. **Sensibilidad estructural:** Pequeñas variaciones en f_i , c_{ij} , d_j o $s_i^{\max}$ pueden cambiar la estructura del óptimo (p. ej. la configuración x^*). Por ello, el análisis de sensibilidad y escenarios es parte inseparable de la modelación robusta (Snyder, 2006; Shapiro et al., 2021).
3. **Elección de M en big-M:** Si se usa la formulación $y_{ij} \leq Mx_i$ para acoplar variables (versus la formulación con $s_i^{\max}$), se recomienda elegir M igual a una cota realista (ej. $\sum_j d_j$) para evitar degradación numérica; empero, la formulación (2.5) es preferible por su interpretación física y por estrechar el dominio factible.

3.2.4.6 Ejemplo numérico (verificación de factibilidad y cálculo de costo)

Tomemos la instancia didáctica ya utilizada en las secciones previas:

- $\mathcal{I} = \{1, 2, 3\}$, $\mathcal{J} = \{1, 2, 3, 4\}$.
- Costos fijos (miles USD): $f = (300, 400, 250)$.
- Matriz $C = [c_{ij}]$ (USD/unidad):

$$C = \begin{bmatrix} 8 & 20 & 15 & 25 \\ 12 & 18 & 10 & 30 \\ 20 & 10 & 12 & 18 \end{bmatrix}.$$

- Demanda: $d = (40, 20, 40, 25)$ (unidades).
- Capacidades: $s^{\max} = (100, 80, 90)$.
- Consideramos la decisión $x = (1, 0, 1)$ y la asignación

$$\begin{aligned} y_{11} &= 30, y_{12} = 20, y_{13} = 0, y_{14} = 0, \\ y_{21} &= y_{22} = y_{23} = y_{24} = 0, \\ y_{31} &= 10, y_{32} = 0, y_{33} = 40, y_{34} = 25. \end{aligned}$$

Paso 1 — Verificación de capacidades (ecuación (2.5)):

- Para $i = 1$:

$$\sum_j y_{1j} = 30 + 20 + 0 + 0 = 50 \leq s_1^{\max} x_1 = 100 \cdot 1 = 100 \quad \checkmark.$$

- Para $i = 2$:

$$\sum_j y_{2j} = 0 \leq 80 \cdot 0 = 0 \quad \checkmark.$$

- Para $i = 3$:

$$\sum_j y_{3j} = 10 + 0 + 40 + 25 = 75 \leq 90 \cdot 1 = 90 \quad \checkmark.$$

Paso 2 — Verificación de cobertura de demanda:

- $j = 1$: $y_{11} + y_{31} = 30 + 10 = 40 = d_1$,
- $j = 2$: $y_{12} = 20 = d_2$,
- $j = 3$: $y_{33} = 40 = d_3$,
- $j = 4$: $y_{34} = 25 = d_4$.

Cobertura completa $\Rightarrow$ factible.

Paso 3 — Cálculo del costo total

La función de costo (determinista, sin mantenimiento de inventario explícito) se toma como

$$Z(x, y) = \sum_{i \in \mathcal{I}} f_i x_i + \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} y_{ij}.$$

Sustituyendo:

- Término fijo: $f_1 x_1 + f_3 x_3 = 300 + 250 = 550$ (miles USD).
- Transporte (USD): calcular cada contribución

$$\begin{aligned} 8 \cdot 30 &= 240, & 20 \cdot 20 &= 400, & 20 \cdot 10 &= 200, \\ 12 \cdot 40 &= 480, & 18 \cdot 25 &= 450. \end{aligned}$$

$$\text{Suma transporte} = 240 + 400 + 200 + 480 + 450 = 1770 \text{ USD.}$$

Total (homogeneizando unidades a miles USD):

$$Z = 550 \text{ (miles)} + 1.770 \text{ (miles)} = 2.320 \text{ miles USD.}$$

Interpretación: la solución es factible y el costo cuantificado; variaciones en d , c_{ij} o f_i deben re-evaluarse mediante análisis de sensibilidad y, en contextos reales, mediante formulaciones estocásticas o robustas (Snyder, 2006; Shapiro et al., 2021).

La especificación explícita de f_i , c_{ij} , d_j y $s_i^{\max}$ permite no solo la construcción del problema determinista base sino también su extensión a modelos estocásticos (reemplazando d_j por variables aleatorias $\tilde{d}_j$) o robustos (definiendo conjuntos de incertidumbre para d_j y c_{ij}). La práctica recomendada (Lewis, 2003; Shapiro et al., 2021) es: (i) documentar fuentes y unidades de cada parámetro; (ii) fijar cotas realistas para $s_i^{\max}$ o M ; y (iii) efectuar pruebas de sensibilidad sistemáticas antes de adoptar políticamente cualquier solución.

3.3 Definición formal del espacio factible $\mathcal{X}$

En optimización matemática, la **estructura del conjunto factible** determina en gran medida la naturaleza del problema, su complejidad computacional y la aplicabilidad de métodos de solución. Para el problema integrado de localización e inventario, el espacio factible es un **subconjunto híbrido** del producto cartesiano entre un espacio discreto y uno continuo. A continuación se define formalmente este conjunto y se analizan sus propiedades estructurales clave.

3.4 Definición (Espacio factible).

Dado el conjunto de instalaciones potenciales $\mathcal{I}$, el conjunto de zonas de demanda $\mathcal{J}$, y los parámetros exógenos $d_j > 0$, $s_i^{\max} \in \mathbb{R}_+ \cup \{+\infty\}$, y una constante $M > 0$ suficientemente grande (por ejemplo, $M = \sum_{j \in \mathcal{J}} d_j$), el **espacio factible** del modelo se define como:

$$\mathcal{X} := \left\{ (x, y) \in \{0, 1\}^m \times \mathbb{R}_+^{m \times n} \left| \begin{array}{ll} \text{(i)} & \sum_{i \in \mathcal{I}} y_{ij} \geq d_j, \quad \forall j \in \mathcal{J}, \\ \text{(ii)} & \sum_{j \in \mathcal{J}} y_{ij} \leq s_i^{\max} x_i, \quad \forall i \in \mathcal{I}, \\ \text{(iii)} & y_{ij} \leq M x_i, \quad \forall (i, j) \in \mathcal{I} \times \mathcal{J} \end{array} \right. \right\}.$$

Las tres familias de restricciones tienen interpretaciones operativas precisas:

- **(i) Cobertura de demanda:** cada zona j debe recibir al menos su demanda esperada d_j .
- **(ii) Capacidad condicionada:** el flujo total saliente de la instalación i no puede exceder su capacidad máxima si está abierta ($x_i = 1$).
- **(iii) Activación lógica fuerte:** garantiza $y_{ij} = 0$ cuando $x_i = 0$, fortaleciendo las relajaciones lineales.

Observación 2.2.4.1.

Si $s_i^{\max} = +\infty$, la restricción (ii) se sustituye simplemente por $y_{ij} \leq M x_i$. La estructura analítica permanece intacta.

3.4.1 Propiedades estructurales del espacio $\mathcal{X}$

3.4.1.1 Proposición (No convexidad y no conexidad).

El conjunto $\mathcal{X}$ es **no convexo** y, en general, **no conexo** en la topología estándar de $\mathbb{R}^m \times \mathbb{R}^{m \times n}$.

Demostración.

Considérese $m \geq 2$. Sean:

- $(x^{(1)}, y^{(1)})$ con $x^{(1)} = e_1$ y $y_{1j}^{(1)} = d_j$,
- $(x^{(2)}, y^{(2)})$ con $x^{(2)} = e_2$ y $y_{2j}^{(2)} = d_j$.

Ambos puntos son factibles. Su media:

$$(\bar{x}, \bar{y}) = \frac{1}{2}(x^{(1)}, y^{(1)}) + \frac{1}{2}(x^{(2)}, y^{(2)})$$

tiene $\bar{x}_1 = \bar{x}_2 = 1/2 \notin \{0, 1\}$, por lo que no pertenece a $\mathcal{X}$.

Además, no existe un camino continuo dentro de $\mathcal{X}$ que conecte ambos puntos, ya que cualquier camino debe atravesar valores fraccionarios de x .

3.4.1.2 Proposición (Descomposición en secciones convexas).

$$\mathcal{X} = \bigcup_{\bar{x} \in \{0,1\}^m} (\{\bar{x}\} \times \mathcal{Y}(\bar{x})),$$

donde

$$\mathcal{Y}(\bar{x}) := \left\{ y \in \mathbb{R}_+^{m \times n} \left| \begin{array}{l} \sum_i y_{ij} \geq d_j, \quad \forall j, \\ \sum_j y_{ij} \leq s_i^{\max} \bar{x}_i, \quad \forall i \end{array} \right. \right\}$$

es un **poliedro convexo** (posiblemente vacío).

Demostración.

Para $\bar{x}$ fijo, las restricciones que definen $\mathcal{Y}(\bar{x})$ son lineales en y , por lo que el conjunto es convexo y cerrado.

La unión es disjunta porque cada sección responde a un valor distinto de x .

Esta descomposición motiva algoritmos híbridos: resolver un subproblema convexo para cada $\bar{x}$ y usar técnicas combinatorias para explorar $\{0, 1\}^m$.

3.4.1.3 Ejemplo ilustrativo: construcción explícita de $\mathcal{X}$

Sea:

- $\mathcal{J} = \{1, 2\}$
- $\mathcal{J} = \{1\}$

- $d_1 = 10$

- $s_1^{\max} = s_2^{\max} = 15$

- $M = 10$

Entonces:

1. $x = (0, 0)$:
 $y_{11} = y_{21} = 0$, pero exige $y_{11} + y_{21} \geq 10$ **infeasible**.
2. $x = (1, 0)$:
 $y_{21} = 0, y_{11} \in [10, 15]$.
3. $x = (0, 1)$:
 $y_{11} = 0, y_{21} \in [10, 15]$.
4. $x = (1, 1)$:
 $y_{11} + y_{21} \geq 10$, con $y_{ij} \leq 15$.
Es un poliedro factible en dimensión 2.

Este ejemplo muestra que $\mathcal{X}$ está formado por componentes disjuntas: dos segmentos y un poliedro 2D.

El conjunto $\mathcal{X}$ posee una estructura combinatoria compleja: es no convexo y no conexo, pero admite una descomposición en secciones convexas. Su correcta caracterización es esencial para asegurar consistencia del modelo, existencia de solución y diseño de algoritmos eficientes. En las siguientes secciones, este espacio actuará como dominio de la función de costo total.

3.4.2 Función objetivo general en problemas de localización–inventario

La **función objetivo** de un modelo de optimización encapsula el criterio de desempeño que se busca minimizar o maximizar. En problemas de localización–inventario para logística humanitaria, este criterio suele ser una medida **económica agregada** que refleja los costos totales del sistema, ponderando compromisos entre infraestructura fija, operación logística y servicio a la demanda. A continuación, se define formalmente la función objetivo general, se descompone en sus componentes estructurales y se analizan sus propiedades matemáticas.

3.4.2.1 Definición (Función objetivo total)

Dado el espacio factible $\mathcal{X}$ definido previamente y los parámetros $f_i \geq 0$, $c_{ij} \geq 0$, la **función objetivo total** $Z : \mathcal{X} \rightarrow \mathbb{R}$ se define como:

$$Z(x, y) := \underbrace{\sum_{i \in \mathcal{I}} f_i x_i}_{\text{(A) Costo fijo de apertura}} + \underbrace{\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} c_{ij} y_{ij}}_{\text{(B) Costo de transporte}} + \underbrace{\sum_{i \in \mathcal{I}} h_i \left(\sum_{j \in \mathcal{J}} y_{ij} \right)}_{\text{(C) Costo de inventario}}.$$

Aquí, $h_i \geq 0$ es el **costo unitario de mantenimiento de inventario** en la instalación i . El término modela el costo asociado al volumen total preposicionado (equivalente al total despachado en el modelo determinista base).

La función Z es **afín en y** para x fijo, y **lineal en x** cuando y es tratado como variable libre. Sin embargo, en el dominio restringido $\mathcal{X}$, la interacción discontinua entre ambas familias de variables induce una **estructura global no lineal**.

Observación (Interpretación económica de los términos)

- (A): decisiones estratégicas de largo plazo.
- (B): costos operativos de corto plazo.
- (C): costo de oportunidad del inventario preposicionado.

En logística humanitaria, f_i , c_{ij} y h_i pueden incorporar factores no monetarios (tiempo, riesgo, deterioro, ética), siempre que estén normalizados en una métrica común.

3.4.2.2 Propiedades analíticas de Z

3.4.2.3 Proposición (Continuidad y acotación inferior)

La función $Z : \mathcal{X} \rightarrow \mathbb{R}$ es **continua** en la topología relativa de $\mathcal{X}$ y está **acotada inferiormente** por cero.

Demostración.

Como $\mathcal{X} \subset \{0, 1\}^m \times \mathbb{R}_+^{m \times n}$, su topología relativa es la unión disjunta de secciones del tipo $\{\bar{x}\} \times \mathcal{Y}(\bar{x})$. En cada sección, la función $y \mapsto Z(\bar{x}, y)$ es afín, y por tanto continua. Dado que todos los coeficientes f_i , c_{ij} , h_i son no negativos, se tiene $Z(x, y) \geq 0$ para todo punto factible.

3.4.2.4 Proposición (Coercividad condicional y existencia de solución)

Si $\mathcal{X} \neq \emptyset$, entonces el problema

$$\min_{(x,y) \in \mathcal{X}} Z(x,y)$$

admite solución óptima.

Demostración.

Como el conjunto discreto $\{0,1\}^m$ es finito, la minimización puede escribirse como:

$$\min_{\bar{x} \in \{0,1\}^m} \left\{ \sum_i f_i \bar{x}_i + \min_{y \in \mathcal{Y}(\bar{x})} \left(\sum_{i,j} (c_{ij} + h_i) y_{ij} \right) \right\}.$$

Para cada $\bar{x}$ tal que $\mathcal{Y}(\bar{x}) \neq \emptyset$, el subproblema interno es un **programa lineal** definido sobre un poliedro no vacío y acotado (ya que existen cotas inferiores por demanda y los coeficientes son no negativos). Por el teorema de existencia de mínimo en poliedros, el subproblema posee solución. Como el número de posibles $\bar{x}$ es finito, el mínimo global existe.

3.4.2.5 Ejemplo ilustrativo: evaluación explícita de $Z(x,y)$

Considérese el escenario:

- $\mathcal{I} = \{1, 2\}$,
- $\mathcal{J} = \{1\}$,
- $d_1 = 10$,
- $f_1 = 300, f_2 = 400$,
- $c_{11} = 10, c_{21} = 12$,
- $h_1 = 3, h_2 = 2$.

Caso 1: $x = (1, 0), y_{11} = 12, y_{21} = 0$

$$\begin{aligned} Z(x,y) &= f_1 x_1 + f_2 x_2 + (c_{11} + h_1) y_{11} + (c_{21} + h_2) y_{21} \\ &= 300 + 0 + (10 + 3) \cdot 12 + (12 + 2) \cdot 0 \\ &= 300 + 156 = 456. \end{aligned}$$

Caso 2: $x = (0, 1)$, $y_{21} = 10$

$$Z = 0 + 400 + (12 + 2) \cdot 10 = 540.$$

Caso 3: $x = (1, 1)$, $y_{11} = 6$, $y_{21} = 4$

$$Z = 300 + 400 + 13 \cdot 6 + 14 \cdot 4 = 834.$$

El valor mínimo se obtiene abriendo únicamente la instalación 1, con inventario despachado 12 (mayor que la demanda), lo cual es admisible y económicamente óptimo bajo estos parámetros.

La función objetivo $Z(x, y)$ formaliza el análisis de compromisos entre inversión en infraestructura, costos operativos e inventario preposicionado. Su estructura afín por partes y la no convexidad del dominio $\mathcal{X}$ definen un MILP clásico, aunque constituyen la base teórica necesaria para extensiones no lineales, robustas o estocásticas. La existencia garantizada de al menos una solución óptima justifica el enfoque algorítmico planteado en el Capítulo 3.

4 Métodos numéricos y optimización con restricciones tipo caja

En este capítulo se presentan los fundamentos teóricos y computacionales de los métodos de optimización aplicados a problemas con restricciones tipo caja, es decir, aquellos en los que las variables de decisión se encuentran acotadas superior e inferiormente dentro de un dominio compacto. Este tipo de problemas es común en la modelación de sistemas físicos, económicos y logísticos, donde las condiciones naturales o estructurales limitan el espacio de búsqueda de soluciones Bertsekas (2016).

El desarrollo del capítulo se organiza en cinco secciones. En primer lugar, se introducen los fundamentos de los métodos iterativos de primer y segundo orden, enfatizando los mecanismos de actualización del gradiente y la aproximación del Hessiano. Posteriormente, se extiende el análisis hacia dominios con restricciones tipo caja, detallando la formulación y justificación estructural del algoritmo **L-BFGS-B**, uno de los más eficientes para resolver problemas de gran escala con cotas bajo el esquema de Byrd et al. (1995); Nocedal y Wright (2006). En la tercera sección se analizan los resultados teóricos de convergencia del método bajo hipótesis de regularidad, presentando un teorema global que garantiza la existencia de puntos estacionarios que satisfacen las condiciones de Karush-Kuhn-Tucker (KKT). En seguida, la cuarta parte describe la implementación computacional del algoritmo y los experimentos numéricos realizados para evaluar su desempeño frente a otros métodos clásicos de optimización sin restricciones. Finalmente, se incluye una discusión crítica acerca de las ventajas, limitaciones y posibles extensiones del enfoque propuesto.

El propósito de este capítulo es ofrecer una comprensión integral del método **L-BFGS-B**, integrando su base matemática, su estructura computacional y su relevancia práctica para la resolución eficiente de problemas con restricciones tipo caja.

4.1 Fundamentos de métodos iterativos de primer y segundo orden

4.1.1 Método de gradiente descendente y sus propiedades de convergencia

El método de gradiente descendente constituye el algoritmo más elemental dentro de los métodos iterativos de primer orden para la minimización no lineal de funciones

diferenciables. Su fundamento se basa en la propiedad geométrica de que el gradiente de una función escalar $f : \mathbb{R}^n \rightarrow \mathbb{R}$, diferenciable en un punto $x \in \mathbb{R}^n$, apunta en la dirección de máximo crecimiento local. En consecuencia, la dirección opuesta, $-\nabla f(x)$, indica un camino de descenso local siempre que $\nabla f(x) \neq 0$ (Nocedal y Wright (2006)).

Formalmente, dado un punto inicial $x_0 \in \mathbb{R}^n$, el método genera una sucesión $\{x_k\}_{k \geq 0}$ definida por

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

donde $\alpha_k > 0$ es la longitud de paso seleccionada en cada iteración mediante una búsqueda lineal adecuada.

La elección de α_k resulta determinante para la convergencia. Una estrategia robusta impone las **condiciones de Wolfe** (Nocedal y Wright (2006)):

$$\begin{aligned} f(x_k - \alpha_k \nabla f(x_k)) &\leq f(x_k) - c_1 \alpha_k \|\nabla f(x_k)\|^2, \\ \nabla f(x_k - \alpha_k \nabla f(x_k))^\top \nabla f(x_k) &\geq c_2 \|\nabla f(x_k)\|^2, \end{aligned}$$

donde $0 < c_1 < c_2 < 1$. La primera asegura un descenso suficiente y la segunda evita pasos demasiado pequeños.

4.1.1.1 Propiedades de convergencia

Bajo las hipótesis:

1. f es continuamente diferenciable en un conjunto abierto que contiene al conjunto de nivel $\mathcal{L} = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$,
2. $\mathcal{L}$ es compacto,

el método de gradiente descendente con pasos que satisfacen las condiciones de Wolfe garantiza que

$$\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0.$$

Si, además, f es **fuertemente convexa**, es decir, existe $m > 0$ tal que

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq m \|x - y\|^2,$$

entonces la sucesión $\{x_k\}$ converge linealmente al minimizador x^* , cumpliendo

$$\|x_{k+1} - x^*\| \leq \rho \|x_k - x^*\|, \quad \rho \in (0, 1).$$

4.1.1.2 Ejemplo ilustrativo

Consideremos $f(x) = \frac{1}{2}x^\top Ax - b^\top x$, con

$$A = \begin{pmatrix} 2 & 0 \\ 0 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 0 \\ 0 \end{pmatrix}.$$

El gradiente es $\nabla f(x) = Ax$. Partiendo de $x_0 = (1, 1)^\top$, se obtiene mediante búsqueda lineal exacta:

$$\alpha_0 = \frac{13}{126} \approx 0.1032, \quad x_1 = (0.7937, -0.0317)^\top.$$

La trayectoria oscila hacia $x^* = (0, 0)^\top$, evidenciando la convergencia lineal y la influencia del número de condición $\kappa(A) = 5$.

Este comportamiento justifica la necesidad de métodos que incorporen información de segundo orden, como los métodos de tipo cuasi-Newton, presentados en la siguiente sección.

4.1.2 Métodos cuasi-Newton: aproximación del Hessiano y convergencia superlineal

Los métodos cuasi-Newton constituyen una clase fundamental de algoritmos iterativos para la minimización de funciones diferenciables sin restricciones, diseñados para emular el comportamiento del método de Newton sin requerir la evaluación explícita del Hessiano $\nabla^2 f(x_k)$. En lugar de ello, estos métodos construyen **aproximaciones sucesivas** $B_k \in \mathbb{R}^{n \times n}$ (o $H_k \approx (\nabla^2 f(x_k))^{-1}$) a partir de la información de gradientes acumulada durante la iteración, logrando así tasas de convergencia **superlineales** bajo hipótesis razonables Nocedal y Wright (2006).

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continuamente diferenciable. El esquema genérico de un método cuasi-Newton se expresa como

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

donde B_k es una matriz simétrica definida positiva que aproxima el Hessiano $\nabla^2 f(x_k)$, y $\alpha_k > 0$ se elige mediante una búsqueda lineal (por ejemplo, las condiciones de Wolfe).

La actualización de B_k se basa en la **ecuación secante**, derivada de la expansión de Taylor de primer orden del gradiente:

$$\nabla f(x_{k+1}) - \nabla f(x_k) \approx \nabla^2 f(x_k)(x_{k+1} - x_k).$$

Definiendo los incrementos

$$s_k := x_{k+1} - x_k, \quad y_k := \nabla f(x_{k+1}) - \nabla f(x_k),$$

la ecuación secante exige que la nueva aproximación B_{k+1} satisfaga

$$B_{k+1}s_k = y_k.$$

Dado que esta ecuación impone solo n restricciones lineales sobre los $\frac{n(n+1)}{2}$ grados de libertad de una matriz simétrica, se requiere un criterio adicional para determinar B_{k+1} de forma única. Los métodos cuasi-Newton clásicos surgen al imponer que B_{k+1} sea la **matriz más cercana a B_k** en alguna norma matricial adecuada, sujeta a la ecuación secante y a la simetría.

4.1.2.1 El método BFGS

El algoritmo **BFGS** (Broyden–Fletcher–Goldfarb–Shanno) es el método cuasi-Newton más utilizado en la práctica. Se obtiene al minimizar la distancia entre $H_{k+1} = B_{k+1}^{-1}$ y $H_k = B_k^{-1}$ en una norma de Frobenius ponderada que garantiza invariancia de escala. La fórmula de actualización de la matriz inversa resultante es

$$H_{k+1} = (I - \rho_k s_k y_k^\top) H_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top,$$

donde $\rho_k = (y_k^\top s_k)^{-1}$ y se asume $y_k^\top s_k > 0$ (condición de curvatura), garantizada si la búsqueda lineal cumple las condiciones de Wolfe.

El método BFGS posee propiedades teóricas destacadas:

- Si f es dos veces continuamente diferenciable y su Hessiano $\nabla^2 f$ es Lipschitz-continuo en una vecindad de un minimizador x^* , y la búsqueda lineal satisface las condiciones de Wolfe, entonces la sucesión $\{x_k\}$ cumple

$$\lim_{k \rightarrow \infty} \|x_k - x^*\| = 0,$$

y la convergencia es **superlineal**:

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Este resultado se basa en la caracterización de Dennis y Moré (1977):

$$\lim_{k \rightarrow \infty} \frac{\|(B_k - \nabla^2 f(x^*))s_k\|}{\|s_k\|} = 0.$$

- Si además f es cuadrática y fuertemente convexa, BFGS converge en a lo sumo n pasos, generando direcciones conjugadas respecto al Hessiano exacto.

4.1.2.2 Ejemplo ilustrativo: minimización cuadrática con BFGS

Sea la función cuadrática fuertemente convexa en $\mathbb{R}^2$:

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x, \quad A = \begin{pmatrix} 2 & 0 \\ 0 & 10 \end{pmatrix}, \quad b = \begin{pmatrix} 2 \\ 10 \end{pmatrix},$$

cuyo minimizador es $x^* = A^{-1}b = (1, 1)^\top$.

Partimos de $x_0 = (0, 0)^\top$ y elegimos $H_0 = I$ como aproximación inicial de la inversa del Hessiano. Utilizamos búsqueda lineal exacta.

Iteración $k = 0$:

- $\nabla f(x_0) = Ax_0 - b = (-2, -10)^\top$.
- Dirección: $p_0 = -H_0 \nabla f(x_0) = (2, 10)^\top$.
- Paso óptimo:

$$\alpha_0 = \frac{\nabla f(x_0)^\top p_0}{p_0^\top A p_0} = \frac{104}{1008} = \frac{13}{126} \approx 0.1032.$$

- Nuevo punto:

$$x_1 = x_0 + \alpha_0 p_0 = (0.2063, 1.0317)^\top.$$

- Incrementos:

$$s_0 = x_1 - x_0, \quad y_0 = \nabla f(x_1) - \nabla f(x_0) = A s_0.$$

- Condición de curvatura: $y_0^\top s_0 \approx 10.746 > 0$.
- Actualización BFGS:

$$H_1 \approx \begin{pmatrix} 0.503 & -0.004 \\ -0.004 & 0.100 \end{pmatrix},$$

que aproxima bien a $A^{-1} = \text{diag}(0.5, 0.1)$.

Iteración $k = 1$:

- $\nabla f(x_1) = Ax_1 - b \approx (-1.587, 0.317)^\top$.
- Dirección: $p_1 = -H_1 \nabla f(x_1) \approx (0.792, -0.032)^\top$.
- Paso óptimo: $\alpha_1 \approx 1.033$.
- Nuevo punto: $x_2 \approx (1.022, 0.999)^\top$, muy cercano a $x^* = (1, 1)^\top$.

En solo dos iteraciones el algoritmo reduce el error residual en más de un orden de magnitud, demostrando la **convergencia superlineal** del método BFGS para funciones cuadráticas. Este comportamiento se mantiene en problemas no cuadráticos siempre que el condicionamiento sea razonable y la búsqueda lineal sea suficientemente precisa.

4.1.3 Familia Broyden y el método BFGS como caso prototípico

La clase de métodos cuasi-Newton puede organizarse en torno a la **familia de actualizaciones de Broyden**, un conjunto paramétrico de fórmulas que preservan la ecuación secante y generan matrices simétricas actualizadas a partir de la información de gradiente acumulada. Esta familia engloba los métodos más relevantes en la práctica, entre ellos el **BFGS** (Broyden–Fletcher–Goldfarb–Shanno) y el **DFP** (Davidon–Fletcher–Powell), y permite analizar sus propiedades de convergencia, estabilidad y eficiencia computacional de manera unificada Nocedal y Wright (2006).

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función dos veces continuamente diferenciable. En cada iteración k , se dispone de los incrementos

$$s_k := x_{k+1} - x_k, \quad y_k := \nabla f(x_{k+1}) - \nabla f(x_k),$$

y se busca una matriz simétrica $B_{k+1} \in \mathbb{R}^{n \times n}$ que satisfaga la **ecuación secante**

$$B_{k+1}s_k = y_k. \quad (3.11)$$

La **familia de Broyden** se define como el conjunto de actualizaciones

$$B_{k+1}(\phi_k) = B_k - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k} + \frac{y_k y_k^\top}{y_k^\top s_k} + \phi_k v_k v_k^\top, \quad (3.12)$$

donde

$$v_k = \left(\frac{y_k}{y_k^\top s_k} - \frac{B_k s_k}{s_k^\top B_k s_k} \right), \quad (3.13)$$

y $\phi_k \in \mathbb{R}$ es un parámetro que determina el miembro particular de la familia.

La fórmula (3.12) puede interpretarse como una combinación convexa de dos actualizaciones extremas:

- **BFGS**: $\phi_k = 0$,
- **DFP**: $\phi_k = 1$.

De hecho,

$$B_{k+1}(\phi_k) = (1 - \phi_k)B_{k+1}^{\text{BFGS}} + \phi_k B_{k+1}^{\text{DFP}}. \quad (3.14)$$

4.1.3.1 Propiedades de la familia restringida de Broyden

Una subclase especialmente relevante es la **familia restringida de Broyden**, asociada a $\phi_k \in [0, 1]$. Bajo la **condición de curvatura**

$$y_k^\top s_k > 0, \quad (3.15)$$

garantizada si la búsqueda lineal cumple las condiciones de Wolfe, todos los miembros de esta subclase preservan la **definida positividad** de B_k , siempre que B_0 sea simétrica definida positiva. Esta propiedad asegura que la dirección de búsqueda

$$p_k = -B_k^{-1} \nabla f(x_k) \quad (3.16)$$

sea de descenso. Además, si f es una función cuadrática fuertemente convexa, cualquier método de la familia restringida de Broyden converge en a lo sumo n pasos, generando direcciones conjugadas respecto al Hessiano exacto $\nabla^2 f$ Dennis y Moré (1977).

4.1.3.2 El método BFGS como caso prototípico

El método **BFGS** corresponde a $\phi_k = 0$ en (3.12). Su fórmula de actualización de la **inversa del Hessiano aproximado** $H_k = B_k^{-1}$ es

$$H_{k+1} = (I - \rho_k s_k y_k^\top) H_k (I - \rho_k y_k s_k^\top) + \rho_k s_k s_k^\top, \quad \rho_k = (y_k^\top s_k)^{-1}. \quad (3.17)$$

Esta formulación evita la inversión explícita de matrices y facilita el cálculo de la dirección de búsqueda:

$$p_k = -H_k \nabla f(x_k).$$

El método BFGS se distingue por:

- Su **invariancia afín**: el comportamiento es independiente de escalas y transformaciones lineales.
- Su **propiedad de autocorrección** (*self-scaling*): si H_k es una mala aproximación, la iteración corrige rápidamente la orientación de la matriz.
- Su **convergencia superlineal** garantizada bajo hipótesis estándar de suavidad y búsqueda lineal Fletcher (1987).

4.1.3.3 Ejemplo ilustrativo: actualización BFGS en $\mathbb{R}^2$

Sea

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x, \quad A = \begin{pmatrix} 4 & 1 \\ 1 & 2 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 1 \end{pmatrix}.$$

El minimizador es $x^* = A^{-1}b = (1/7, 3/7)^\top \approx (0.1429, 0.4286)^\top$. Partimos de $x_0 = (0, 0)^\top$ y $H_0 = I$.

Iteración $k = 0$:

- $\nabla f(x_0) = (-1, -1)^\top$.
- $p_0 = (1, 1)^\top$.
- $\alpha_0 = \frac{2}{8} = 0.25$.
- $x_1 = (0.25, 0.25)^\top$.
- $\nabla f(x_1) = (0.25, -0.25)^\top$.
- $s_0 = (0.25, 0.25)^\top$, $y_0 = (1.25, 0.75)^\top$.
- $y_0^\top s_0 = 0.5 > 0$, $\rho_0 = 2$.
- $H_1 \approx \begin{pmatrix} 0.406 & -0.344 \\ -0.344 & 0.906 \end{pmatrix}$.

Iteración $k = 1$:

- $p_1 = -H_1(0.25, -0.25)^\top \approx (-0.1875, 0.3125)^\top$.
- $\alpha_1 \approx 0.857$, $x_2 \approx (0.089, 0.518)^\top$.

Tras dos iteraciones, $\|x_2 - x^*\| \approx 0.09$, y H_1 aproxima bien A^{-1} , demostrando la capacidad del método para capturar la geometría de segundo orden con pocas actualizaciones.

El método BFGS representa el **caso prototípico** dentro de la familia de Broyden, al combinar eficiencia computacional, robustez teórica y convergencia superlineal incluso en problemas moderadamente mal condicionados.

4.2 Extensión a dominios con restricciones tipo caja

En la práctica, muchos problemas de optimización imponen límites naturales o estructurales sobre las variables de decisión, dando origen a los denominados **problemas con restricciones tipo caja**. En este contexto, las variables están acotadas por límites inferiores y superiores que definen un dominio factible de la forma $\Omega = \{x \in \mathbb{R}^n : \ell_i \leq x_i \leq u_i\}$.

Estos problemas aparecen con frecuencia en aplicaciones de ingeniería, finanzas y logística, donde ciertas magnitudes no pueden sobrepasar intervalos físicamente admisibles. Extender los métodos de optimización a dominios de este tipo requiere incorporar

operadores de **proyección ortogonal** que aseguren la factibilidad de cada iteración sin comprometer la convergencia del algoritmo. La presente sección desarrolla las propiedades analíticas de dicha proyección, que constituye el núcleo geométrico de los métodos con restricciones tipo caja.

4.2.1 Proyección ortogonal sobre conjuntos de cotas: definición y propiedades

En la optimización con restricciones tipo caja, el **conjunto factible** se define como

$$\Omega = \{x \in \mathbb{R}^n : \ell_i \leq x_i \leq u_i, \ i = 1, \dots, n\},$$

donde $\ell = (\ell_1, \dots, \ell_n)^\top \in \mathbb{R}^n \cup \{-\infty\}^n$ y $u = (u_1, \dots, u_n)^\top \in \mathbb{R}^n \cup \{+\infty\}^n$ satisfacen $\ell_i < u_i$ para todo i . Este conjunto es **convexo, cerrado y no vacío**, lo que garantiza que la proyección ortogonal de cualquier punto $z \in \mathbb{R}^n$ sobre Ω está bien definida y es única.

4.2.1.1 Proyección Ortogonal

La proyección ortogonal de z sobre Ω , denotada $P_\Omega(z)$, se define como la solución única del problema de minimización

$$P_\Omega(z) = \arg \min_{x \in \Omega} \frac{1}{2} \|x - z\|^2. \quad (3.18)$$

Dado que Ω es un producto cartesiano de intervalos cerrados, la proyección se **descompone coordenada a coordenada**:

$$[P_\Omega(z)]_i = \min\{u_i, \max\{\ell_i, z_i\}\}, \quad i = 1, \dots, n. \quad (3.19)$$

Esta expresión corresponde a la llamada *función de recorte* (*clipping function*), que restringe cada componente de z al intervalo factible $[\ell_i, u_i]$.

4.2.1.2 Propiedades fundamentales

1. No expansividad

$$\|P_\Omega(z) - P_\Omega(w)\| \leq \|z - w\|, \quad \forall z, w \in \mathbb{R}^n. \quad (3.20)$$

Por tanto, P_Ω es una aplicación **Lipschitz continua con constante 1**, lo que implica estabilidad numérica.

2. Caracterización variacional

Un punto $x^* \in \Omega$ satisface $x^* = P_\Omega(z)$, si y solo si,

$$(z - x^*)^\top (x - x^*) \leq 0, \quad \forall x \in \Omega. \quad (3.21)$$

Geoméricamente, esto significa que el vector residual $z - x^*$ forma un ángulo obtuso con cualquier dirección factible desde x^* .

3. Condiciones de complementariedad (KKT)

Para cada i se cumple

$$\begin{cases} x_i = \ell_i & \Rightarrow z_i \leq \ell_i, \\ x_i = u_i & \Rightarrow z_i \geq u_i, \\ \ell_i < x_i < u_i & \Rightarrow z_i = x_i, \end{cases} \quad (3.22)$$

que equivalen a las condiciones de optimalidad de primer orden del problema cuadrático (3.18).

4.2.1.3 Ejemplo: proyección en $\mathbb{R}^3$

Sea

$$\Omega = \{x \in \mathbb{R}^3 : 1 \leq x_1 \leq 3, x_2 \leq 2, x_3 \geq 0\}, \quad z = (0.5, 2.5, -1.2)^\top.$$

Aplicando (3.19) coordenada a coordenada:

- $x_1^* = \min\{3, \max\{1, 0.5\}\} = 1,$
- $x_2^* = \min\{2, \max\{-\infty, 2.5\}\} = 2,$
- $x_3^* = \min\{+\infty, \max\{0, -1.2\}\} = 0.$

Por tanto,

$$P_\Omega(z) = (1, 2, 0)^\top.$$

Comprobamos la caracterización variacional:

$$(z - x^*)^\top (x - x^*) = (-0.5, 0.5, -1.2)^\top \cdot (x_1 - 1, x_2 - 2, x_3 - 0).$$

Dado que $x_1 \geq 1$, $x_2 \leq 2$, $x_3 \geq 0$, cada término parcial es no positivo, y la suma total ≤ 0 .

Asimismo, las condiciones (3.22) se satisfacen coordenada a coordenada: $x_1^* = 1$ con $z_1 = 0.5 \leq 1$, $x_2^* = 2$ con $z_2 = 2.5 \geq 2$, y $x_3^* = 0$ con $z_3 = -1.2 \leq 0$.

Este ejemplo ilustra cómo la proyección actúa localmente sobre cada componente, garantizando factibilidad sin resolver sistemas lineales. Su simplicidad y propiedades de estabilidad hacen que el operador P_Ω sea esencial en algoritmos de optimización a gran escala, como el método **L-BFGS-B**, en el cual se aplica en cada iteración para mantener las variables dentro de los límites definidos.

4.2.2 El algoritmo L-BFGS-B: formulación y justificación estructural

El algoritmo **L-BFGS-B** (*Limited-memory BFGS with Bounds*) extiende el método cuasi-Newton BFGS al caso de **restricciones tipo caja**, es decir, dominios de la forma

$$\Omega = \{x \in \mathbb{R}^n : \ell \leq x \leq u\},$$

donde las desigualdades son componente a componente y $\ell, u \in \mathbb{R}^n \cup \{\pm\infty\}^n$ con $\ell_i < u_i$. El método integra tres ideas fundamentales:

1. **Aproximación cuasi-Newton de memoria limitada**, evitando el almacenamiento completo de matrices densas;
2. **Proyección ortogonal** sobre Ω para mantener factibilidad;
3. **Retención parcial de información de curvatura**, conservando sólo las m parejas más recientes (s_i, y_i) .

4.2.2.1 Formulación del algoritmo

Dado un punto inicial $x_0 \in \Omega$, el esquema iterativo se expresa como

$$x_{k+1} = P_\Omega(x_k - \alpha_k H_k \nabla f(x_k)), \quad (3.23)$$

donde $\alpha_k > 0$ satisface las condiciones de Wolfe y H_k es la aproximación de memoria limitada al inverso del Hessiano, construida mediante las m actualizaciones más recientes:

$$s_i = x_{i+1} - x_i, \quad y_i = \nabla f(x_{i+1}) - \nabla f(x_i), \quad i = k - m, \dots, k - 1,$$

con $y_i^\top s_i > 0$ (condición de curvatura).

La proyección P_Ω se calcula coordenada a coordenada como

$$[P_\Omega(z)]_i = \min\{u_i, \max\{\ell_i, z_i\}\}, \quad i = 1, \dots, n. \quad (3.24)$$

El producto $H_k \nabla f(x_k)$ se obtiene por la **recursión de dos bucles**, iniciada con $H_k^{(0)} = \gamma_k I$, donde

$$\gamma_k = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}. \quad (3.25)$$

Este escalamiento mejora la estabilidad numérica y la aceptación de pasos unitarios.

4.2.2.2 Justificación estructural

El esquema (3.23) garantiza factibilidad a cada iteración mediante la proyección P_Ω . Definiendo el conjunto de índices activos

$$\mathcal{A}_k = \{i : x_{k,i} = \ell_i \text{ o } x_{k,i} = u_i\},$$

el método se comporta localmente como BFGS en los índices inactivos ($i \notin \mathcal{A}_k$) y mantiene las componentes activas en las cotas.

De este modo, la actualización proyectada puede interpretarse como

$$x_{k+1} = \text{Proj}_\Omega(x_k - \alpha_k \nabla^2 f(x_k)^{-1} \nabla f(x_k)),$$

con $\nabla^2 f(x_k)^{-1}$ sustituido por la aproximación limitada H_k .

Esta interpretación explica por qué L-BFGS-B conserva tanto la factibilidad como las propiedades de descenso y la convergencia superlineal del BFGS original.

En términos computacionales, mientras que BFGS requiere $\mathcal{O}(n^2)$ memoria, L-BFGS-B reduce el costo a $\mathcal{O}(mn)$, con $m \ll n$ (típicamente $3 \leq m \leq 20$).

4.2.2.3 Ejemplo: minimización cuadrática con restricciones tipo caja

Considérese

$$\min_{x \in \mathbb{R}^2} f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad \text{sueto a} \quad 1 \leq x_1 \leq 3, \quad x_2 \geq 0.$$

El minimizador irrestricto $x^* = (2, 1)^\top$ pertenece a Ω , por lo que coincide con el minimizador restringido.

Con $m = 1$ y $x_0 = (1, 0)^\top$, el procedimiento es:

Iteración 0

$\nabla f(x_0) = (-2, -2)^\top$, $H_0 = I$, $d_0 = (2, 2)^\top$.
 Con $\alpha_0 = 1$, se obtiene $\tilde{x}_1 = (3, 2)^\top$ y $x_1 = P_\Omega(\tilde{x}_1) = (3, 2)^\top$.
 $\nabla f(x_1) = (2, 2)^\top$, $s_0 = (2, 2)^\top$, $y_0 = (4, 4)^\top$.

Iteración 1

$$\gamma_1 = \frac{s_0^\top y_0}{y_0^\top y_0} = 0.5,$$

$$H_1^{(0)} = 0.5I.$$

Aplicando la recursión de dos bucles:

$$q = \nabla f(x_1), \alpha_0 = 0.5, q = q - 0.5y_0 = (0, 0)^\top,$$

$$r = H_1^{(0)} q = (0, 0)^\top,$$

$$r = r + s_0(\alpha_0 - \frac{y_0^\top r}{y_0^\top s_0}) = (1, 1)^\top.$$

$$\text{Dirección } d_1 = -r = (-1, -1)^\top.$$

$$\text{Con } \alpha_1 = 1, \tilde{x}_2 = (2, 1)^\top, x_2 = P_\Omega(\tilde{x}_2) = (2, 1)^\top.$$

El método converge en dos iteraciones, demostrando que la estructura de memoria limitada y la proyección garantizan simultáneamente **eficiencia y factibilidad**.

La formulación L-BFGS-B preserva las propiedades de convergencia global del BFGS bajo las hipótesis de suavidad y curvatura positiva. Para más detalles Byrd et al. (1995); Zhu et al. (1997), y constituye uno de los métodos estándar para problemas de gran escala con restricciones tipo caja.

4.2.3 Condiciones de optimalidad bajo cotas: KKT para problemas con restricciones tipo caja

Tras haber introducido la formulación proyectada del algoritmo L-BFGS-B, resulta necesario caracterizar formalmente las condiciones de optimalidad que dicho método busca satisfacer. En particular, cuando el dominio está delimitado por cotas inferiores y superiores, las condiciones de Karush–Kuhn–Tucker (KKT) adquieren una estructura particularmente simple y separable por coordenadas. Esta sección desarrolla esa caracterización, que constituye el fundamento teórico de la convergencia de los métodos con restricciones tipo caja.

Consideremos el problema de optimización

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sujeto a} \quad \ell \leq x \leq u, \quad (3.26)$$

donde $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable en un conjunto abierto que contiene al conjunto factible

$$\Omega = \{x \in \mathbb{R}^n : \ell_i \leq x_i \leq u_i, i = 1, \dots, n\},$$

con $\ell_i < u_i$ para todo i . Este problema constituye un caso particular de programación no lineal con restricciones de desigualdad simples.

4.2.3.1 Condiciones de Karush–Kuhn–Tucker (KKT)

Reescribiendo las restricciones como desigualdades estándar,

$$c_i^\ell(x) := \ell_i - x_i \leq 0, \quad c_i^u(x) := x_i - u_i \leq 0, \quad i = 1, \dots, n, \quad (3.27)$$

y suponiendo que se cumple la condición de calificación de Mangasarian–Fromovitz (MFQC), las **condiciones necesarias de Karush–Kuhn–Tucker** establecen que si $x^* \in \Omega$ es un minimizador local, existen multiplicadores de Lagrange $\lambda^\ell, \lambda^u \in \mathbb{R}^n$ tales que

$$\nabla f(x^*) - \lambda^\ell + \lambda^u = 0. \quad (4.2.1)$$

$$\lambda_i^\ell \geq 0, \quad \lambda_i^u \geq 0. \quad (4.2.2)$$

$$\lambda_i^\ell (x_i^* - \ell_i) = 0. \quad (4.2.3)$$

$$\lambda_i^u (u_i - x_i^*) = 0. \quad (4.2.4)$$

para todo $i = 1, \dots, n$.

Las ecuaciones (3.28c)–(3.28d) expresan la **complementariedad** entre los multiplicadores y las restricciones activas.

4.2.3.2 Forma equivalente sin multiplicadores

Eliminando λ^ℓ y λ^u , las condiciones KKT pueden escribirse componente a componente como

$$\forall i = 1, \dots, n : \quad \begin{cases} \nabla_i f(x^*) = 0, & \text{si } \ell_i < x_i^* < u_i, \\ \nabla_i f(x^*) \geq 0, & \text{si } x_i^* = \ell_i, \\ \nabla_i f(x^*) \leq 0, & \text{si } x_i^* = u_i, \end{cases} \quad (3.29)$$

donde las desigualdades se interpretan componente a componente. Geométricamente, esto significa que el gradiente proyectado sobre el **cono tangente** de Ω en x^* es nulo.

Una formulación alternativa, útil en algoritmos numéricos, es la **condición de estacionariedad proyectada**:

$$\nabla_{\Omega} f(x^*) := P_{\Omega}(x^* - \nabla f(x^*)) - x^* = 0. \quad (3.30)$$

La igualdad (3.30) es **equivalente** a las condiciones KKT cuando f es convexa, y constituye una **condición necesaria** de optimalidad en el caso no convexo. En la práctica, se emplea como criterio de convergencia en algoritmos proyectados como L-BFGS-B.

4.2.3.3 Ejemplo: verificación explícita de KKT

Sea

$$\min_{x \in \mathbb{R}^2} f(x) = (x_1 - 1)^2 + (x_2 - 2)^2, \quad \text{sujeto a } 1 \leq x_1 \leq 3, \quad x_2 \geq 0. \quad (3.31)$$

El gradiente es $\nabla f(x) = (2(x_1 - 1), 2(x_2 - 2))^{\top}$.

Paso 1. Identificación del candidato.

El minimizador irrestricto $(1, 2)^{\top}$ pertenece a Ω , por lo tanto $x^* = (1, 2)^{\top}$ es candidato a solución óptima.

Paso 2. Verificación de (3.29).

- $x_1^* = \ell_1 = 1 \Rightarrow \nabla_1 f(x^*) = 0 \geq 0$
- $\ell_2 < x_2^* < +\infty \Rightarrow \nabla_2 f(x^*) = 0$

Paso 3. Multiplicadores KKT.

De (3.28a),

$$\nabla f(x^*) - \lambda^{\ell} + \lambda^u = 0 \Rightarrow \lambda^{\ell} = \lambda^u = 0.$$

Paso 4. Estacionariedad proyectada.

$$x^* - \nabla f(x^*) = (1, 2), \quad P_{\Omega}(1, 2) = (1, 2), \quad \Rightarrow \nabla_{\Omega} f(x^*) = 0.$$

Así, x^* satisface todas las condiciones de Karush–Kuhn–Tucker y es el minimizador global.

Este análisis demuestra que las condiciones KKT para restricciones tipo caja poseen una estructura **desacoplada y local**, lo que permite su implementación eficiente en métodos de gran escala. En particular, la condición proyectada (3.30) constituye la base

teórica de los algoritmos de proyección iterativa, incluyendo el método L-BFGS-B, cuya convergencia global se verifica precisamente mediante la nulidad asintótica de $\nabla_{\Omega} f(x_k)$.

4.3 Convergencia del método L-BFGS-B

4.3.1 Hipótesis de regularidad: Lipschitz-continuidad del gradiente y compacidad del dominio

El análisis de convergencia global del algoritmo **L-BFGS-B** requiere hipótesis estructurales sobre la función objetivo $f : \mathbb{R}^n \rightarrow \mathbb{R}$ y sobre el conjunto factible $\Omega \subset \mathbb{R}^n$. Estas condiciones garantizan la existencia de minimizadores, la acotación de las iteraciones y la controlabilidad del gradiente, elementos esenciales para asegurar tanto la **bien definición del algoritmo** como su **convergencia hacia puntos estacionarios**.

4.3.1.1 Hipótesis fundamentales

Sea

$$\Omega = \{x \in \mathbb{R}^n : \ell \leq x \leq u\},$$

con $\ell, u \in \mathbb{R}^n \cup \{\pm\infty\}^n$ y $\ell_i < u_i$ para todo i . Supondremos que f satisface las siguientes condiciones:

(H1) Diferenciabilidad y Lipschitz-continuidad del gradiente.

La función f es continuamente diferenciable en un conjunto abierto que contiene a Ω , y su gradiente ∇f es **Lipschitz continuo** en Ω ; es decir, existe una constante $L > 0$ tal que

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Omega. \quad (3.37)$$

(H2) Compacidad del conjunto de nivel inicial.

Dado un punto inicial $x_0 \in \Omega$, el conjunto de nivel asociado

$$\mathcal{L} := \{x \in \Omega : f(x) \leq f(x_0)\} \quad (3.38)$$

es **no vacío, cerrado y acotado**, y por tanto **compacto**.

La hipótesis (H1) implica que el gradiente no cambia bruscamente, lo que permite controlar los pasos del algoritmo mediante la cota superior L . Equivalentemente, la desigualdad de suavidad se cumple:

$$f(y) \leq f(x) + \nabla f(x)^\top (y - x) + \frac{L}{2} \|y - x\|^2, \quad \forall x, y \in \Omega. \quad (3.39)$$

Esta relación garantiza que las búsquedas lineales que satisfacen las **condiciones de Wolfe** producen pasos uniformemente acotados y no degenerados.

La hipótesis (H2), por su parte, asegura que las iteraciones $\{x_k\}$ permanecen en un subconjunto **compacto y factible** de Ω . Esto permite aplicar resultados de convergencia por subsucesión y garantiza que las secuencias de gradientes y valores de función no divergen.

4.3.1.2 Consecuencias teóricas

Bajo las hipótesis (H1) y (H2), la sucesión $\{x_k\}$ generada por L-BFGS-B con una búsqueda lineal que cumple las condiciones de Wolfe posee las siguientes propiedades:

1. **Existencia de puntos límite.**

Toda subsucesión de $\{x_k\}$ posee una subsucesión convergente cuyo límite pertenece a $\mathcal{L} \subset \Omega$.

2. **Condición de curvatura.**

Las parejas de actualización

$$s_k = x_{k+1} - x_k, \quad y_k = \nabla f(x_{k+1}) - \nabla f(x_k),$$

satisfacen $y_k^\top s_k > 0$ siempre que se cumplan las condiciones de Wolfe, lo que garantiza que las aproximaciones cuasi-Newton sean definidas positivas (Byrd et al., 1995).

3. **Acotación de los operadores de memoria limitada.**

La recursión de dos bucles empleada en L-BFGS-B mantiene direcciones de búsqueda acotadas y de descenso, gracias a la compacidad de $\mathcal{L}$ y a la positividad de los escalares $\rho_k = (y_k^\top s_k)^{-1}$.

4.3.1.3 Teorema de convergencia global (versión esquemática)

Teorema.

Supóngase que f satisface (H1) y (H2), y que la búsqueda lineal cumple las condiciones de Wolfe. Si las actualizaciones de memoria limitada satisfacen $y_k^\top s_k > 0$ o son rechazadas en caso contrario (salvaguarda numérica), entonces la sucesión $\{x_k\}$ generada por L-BFGS-B verifica

$$\liminf_{k \rightarrow \infty} \|\nabla_\Omega f(x_k)\| = 0, \tag{3.40}$$

donde el **gradiente proyectado** se define como

$$\nabla_\Omega f(x) := P_\Omega(x - \nabla f(x)) - x. \tag{3.41}$$

Bosquejo de prueba.

De (H2) se sigue que $\{x_k\} \subset \mathcal{L}$ es acotada, por lo que posee subsucesiones convergentes. La desigualdad (3.39) y las condiciones de Wolfe garantizan un descenso suficiente de f . La condición de curvatura $y_k^\top s_k > 0$ asegura que las matrices H_k permanecen definidas positivas y acotadas. Por tanto, todo punto límite $\bar{x}$ de la sucesión satisface la condición de estacionariedad proyectada $\nabla_\Omega f(\bar{x}) = 0$. Nocedal y Wright (2006); Byrd et al. (1995).

4.3.1.4 Ejemplo de verificación de las hipótesis

Consideremos el problema

$$\min_{x \in \mathbb{R}^2} f(x) = x_1^4 + x_2^2 \quad \text{sujeto a } 0 \leq x_1 \leq 2, \quad -1 \leq x_2 \leq 1. \quad (3.42)$$

Paso 1. Verificación de (H1).

El gradiente es

$$\nabla f(x) = (4x_1^3, 2x_2)^\top, \quad \nabla^2 f(x) = \begin{pmatrix} 12x_1^2 & 0 \\ 0 & 2 \end{pmatrix}.$$

En $\Omega = [0, 2] \times [-1, 1]$,

$$\|\nabla^2 f(x)\|_2 = \max\{12x_1^2, 2\} \leq 48.$$

Por tanto, $L = 48$ es una constante de Lipschitz para ∇f en Ω , y (H1) se cumple.

Paso 2. Verificación de (H2).

Sea $x_0 = (2, 1)^\top \in \Omega$. Entonces

$$f(x_0) = 2^4 + 1^2 = 17, \quad \mathcal{L} = \{x \in \Omega : x_1^4 + x_2^2 \leq 17\}.$$

Dado que Ω es compacto y $\mathcal{L} \subset \Omega$, el conjunto de nivel es cerrado, acotado y no vacío (pues $x^* = (0, 0)^\top \in \mathcal{L}$). Así, (H2) también se satisface.

Paso 3. Implicaciones para L-BFGS-B.

Bajo estas condiciones, la sucesión $\{x_k\}$ generada por L-BFGS-B permanece en $\mathcal{L}$, el gradiente es Lipschitz con constante $L = 48$, y las parejas (s_k, y_k) satisfacen $y_k^\top s_k > 0$. Por tanto, el algoritmo produce direcciones de descenso bien definidas, y los límites de las iteraciones satisfacen las condiciones de Karush–Kuhn–Tucker bajo restricciones tipo caja.

En síntesis, las hipótesis (H1) y (H2) proporcionan el **marco teórico mínimo** para garantizar la convergencia global del método L-BFGS-B. Estas condiciones, verificables en la práctica, sustentan la robustez del algoritmo en problemas suaves y acotados típicos de la optimización aplicada.

4.3.2 Teorema de convergencia global: límite de puntos estacionarios que satisfacen KKT

El algoritmo **L-BFGS-B** está diseñado para resolver problemas de optimización no lineal con restricciones tipo caja de la forma

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sujeto a} \quad \ell \leq x \leq u. \quad (3.60)$$

Aquí, $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es continuamente diferenciable, y el conjunto factible

$$\Omega = \{x \in \mathbb{R}^n : \ell \leq x \leq u\}$$

es cerrado, convexo y no vacío.

El **resultado fundamental de convergencia global**, Byrd et al. (1995), establece que, bajo hipótesis estándar de regularidad y condiciones adecuadas de búsqueda lineal, la sucesión generada por el algoritmo posee puntos límite que satisfacen las **condiciones necesarias de optimalidad de Karush–Kuhn–Tucker (KKT)**. Dicho de otra forma, el algoritmo converge —en el sentido de subsecuencias— hacia el conjunto de puntos estacionarios proyectados.

4.3.2.1 Teorema de convergencia global

Teorema 3.2 (Convergencia global de L-BFGS-B).

Sea $f : \mathbb{R}^n \rightarrow \mathbb{R}$ una función continuamente diferenciable que satisface:

1. ∇f es **Lipschitz continuo** en Ω ; existe $L > 0$ tal que

$$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|, \quad \forall x, y \in \Omega; \quad (3.61)$$

2. El conjunto de nivel inicial

$$\mathcal{L} := \{x \in \Omega : f(x) \leq f(x_0)\} \quad (3.62)$$

es **compacto**.

Supóngase que la sucesión $\{x_k\} \subset \Omega$ se genera mediante el algoritmo L-BFGS-B con direcciones de búsqueda

$$d_k = P_\Omega(x_k - H_k \nabla f(x_k)) - x_k, \quad (3.63)$$

donde H_k es la aproximación inversa del Hessiano obtenida mediante la recursión de dos bucles con memoria limitada m , y que las longitudes de paso $\alpha_k > 0$ satisfacen las **condiciones de Wolfe**:

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + c_1 \alpha_k \nabla f(x_k)^\top d_k, \\ \nabla f(x_k + \alpha_k d_k)^\top d_k &\geq c_2 \nabla f(x_k)^\top d_k, \end{aligned} \quad (3.64)$$

con $0 < c_1 < c_2 < 1$.

Entonces, la sucesión $\{x_k\}$ verifica:

$$\liminf_{k \rightarrow \infty} \|\nabla_\Omega f(x_k)\| = 0, \quad (3.65)$$

donde

$$\nabla_\Omega f(x) := P_\Omega(x - \nabla f(x)) - x \quad (3.66)$$

es el **gradiente proyectado**.

En consecuencia, toda subsucesión convergente de $\{x_k\}$ converge a un punto $x^* \in \Omega$ que satisface las **condiciones KKT** del problema (3.60).

4.3.2.2 Bosquejo de la demostración

1. Acotación de la sucesión.

Por compacidad de $\mathcal{L}$, las iteraciones $\{x_k\} \subset \mathcal{L}$ son acotadas, y existe una subsucesión convergente $x_{k_j} \rightarrow x^* \in \mathcal{L}$.

2. Positividad de la curvatura.

Bajo las condiciones de Wolfe, las diferencias $s_k = x_{k+1} - x_k$ y $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ satisfacen

$$y_k^\top s_k \geq (1 - c_2) \|\nabla f(x_k)\|^2 \alpha_k > 0, \quad (3.67)$$

asegurando que las actualizaciones L-BFGS mantengan $H_k \succ 0$.

3. Dirección de descenso.

La proyección garantiza que

$$\nabla f(x_k)^\top d_k \leq -\|d_k\|^2, \quad (3.68)$$

es decir, d_k es una dirección de descenso o nula solo en puntos estacionarios.

4. Aplicación del lema de Zoutendijk.

El análisis estándar Zoutendijk (1970); Nocedal y Wright (2006) implica que

$$\sum_{k=0}^{\infty} \frac{(\nabla f(x_k)^\top d_k)^2}{\|d_k\|^2} < \infty. \quad (3.69)$$

Dado que el numerador está acotado inferiormente por una constante positiva cuando $\|\nabla_\Omega f(x_k)\|$ no tiende a cero, se concluye que $\|d_k\| \rightarrow 0$.

Como $d_k = \nabla_\Omega f(x_k)$, se obtiene el resultado (3.65).

5. Condición KKT en el límite.

Por continuidad de P_Ω y de ∇f , si $x_{k_j} \rightarrow x^*$, entonces

$$P_\Omega(x^* - \nabla f(x^*)) = x^*,$$

que equivale a las condiciones KKT (véase Sección 3.2.3).

4.3.2.3 Ejemplo de convergencia hacia un punto KKT

Consideremos:

$$\min_{x \in \mathbb{R}^2} f(x) = (x_1 - 2)^2 + (x_2 - 0.5)^2 \quad \text{sujeto a } x_1 \geq 1, x_2 \geq 1. \quad (3.70)$$

El minimizador irrestricto es $x^{\text{free}} = (2, 0.5)^\top \notin \Omega$.

El punto más cercano factible es $x^* = (2, 1)^\top$.

Verificación KKT.

- $x_1^* > 1 \Rightarrow \nabla_1 f(x^*) = 0$,
 - $x_2^* = 1 = \ell_2 \Rightarrow \nabla_2 f(x^*) = 2(1 - 0.5) = 1 \geq 0$.
- Por tanto, x^* satisface KKT.

Evolución de L-BFGS-B (memoria limitada $m = 2$):

- Desde $x_0 = (1, 1)^\top$, se obtiene $x_1 = (3, 1)$, $x_2 = (2.5, 1)$, $x_3 = (2.25, 1)$, $x_4 = (2.125, 1)$, hasta estabilizarse en $x_k \rightarrow (2, 1)$.
- Para cada iteración:

$$\nabla_\Omega f(x_k) = P_\Omega(x_k - \nabla f(x_k)) - x_k \rightarrow 0,$$

confirmando la convergencia proyectada.

En consecuencia, bajo las hipótesis de Lipschitz-continuidad del gradiente y compacidad del dominio, **toda sucesión generada por L-BFGS-B converge hacia el conjunto de puntos estacionarios que satisfacen las condiciones KKT**. Este resultado formaliza la robustez teórica del método y sustenta su amplio uso en problemas de gran escala y restricciones tipo caja.

4.3.3 Corolario aplicado al modelo propuesto: existencia y unicidad de $Q^* \geq \mathbf{1}$ tal que $\nabla f(Q^*) = 0$

En el contexto del modelo propuesto, el problema de optimización adopta la forma

$$\min_{Q \in \mathbb{R}^n} f(Q) \quad \text{sujeto a} \quad Q \geq \mathbf{1}, \quad (3.75)$$

donde $\mathbf{1} \in \mathbb{R}^n$ es el vector de unos, y $f : \mathbb{R}^n \rightarrow \mathbb{R}$ representa un funcional diferenciable derivado de la formulación teórica del modelo (por ejemplo, una energía libre regularizada, un potencial penalizado o un costo de control con barrera logarítmica).

Se asume que f cumple las siguientes **hipótesis estructurales**:

(M1) Diferenciabilidad y convexidad estricta.

f es dos veces continuamente diferenciable en un conjunto abierto que contiene $\{Q \geq \mathbf{1}\}$, y es **estrictamente convexa**, es decir:

$$(\nabla f(Q) - \nabla f(R))^\top (Q - R) > 0, \quad \forall Q \neq R, Q, R \geq \mathbf{1}. \quad (3.76)$$

(M2) Coercividad en el dominio factible.

Se cumple que

$$\lim_{\|Q\| \rightarrow \infty, Q \geq \mathbf{1}} f(Q) = +\infty. \quad (3.77)$$

(M3) Lipschitz-continuidad del gradiente en conjuntos de nivel.

Para todo $Q_0 \geq \mathbf{1}$ existe $L > 0$ tal que

$$\|\nabla f(Q) - \nabla f(R)\| \leq L\|Q - R\|, \quad \forall Q, R \in \{Q \geq \mathbf{1} : f(Q) \leq f(Q_0)\}. \quad (3.78)$$

Bajo estas condiciones, el problema (3.75) admite un **único minimizador global** $Q^* \geq \mathbf{1}$.

Además, por las condiciones KKT (véase Sección 3.2.3):

- Si $Q_i^* > 1$, entonces $\nabla_i f(Q^*) = 0$;
- Si $Q_i^* = 1$, entonces $\nabla_i f(Q^*) \geq 0$.

En el modelo propuesto, el contexto físico garantiza que la solución óptima se encuentra **estrictamente en el interior del dominio factible**, es decir,

$$Q^* > \mathbf{1}. \quad (3.79)$$

Por tanto, ninguna restricción está activa, y las condiciones KKT se reducen a la **estacionariedad irrestricta**:

$$\nabla f(Q^*) = 0. \quad (3.80)$$

4.3.3.1 Corolario de convergencia global

Corolario 3.3.1 (Convergencia a $Q^* \geq 1$ con $\nabla f(Q^*) = 0$).

Supóngase que el problema (3.75) satisface (M1)–(M3), y que el minimizador único Q^* verifica $Q^* > 1$.

Sea $\{Q_k\}$ la sucesión generada por el algoritmo L-BFGS-B con punto inicial $Q_0 \geq 1$ y pasos α_k que cumplen las condiciones de Wolfe.

Entonces:

$$\lim_{k \rightarrow \infty} Q_k = Q^*, \quad \nabla f(Q^*) = 0. \quad (3.81)$$

Demostración (esquema).

Por (M2), el conjunto de nivel

$$\mathcal{L} = \{Q \geq 1 : f(Q) \leq f(Q_0)\}$$

es compacto; por (M3), el gradiente es Lipschitz en $\mathcal{L}$.

Por tanto, se satisfacen las hipótesis del **Teorema 3.2** (Sección 3.3.2), de donde resulta

$$\liminf_{k \rightarrow \infty} \|\nabla_{\Omega} f(Q_k)\| = 0.$$

Dado que f es estrictamente convexa, cualquier punto estacionario es el único minimizador global Q^* .

Además, al ser $Q^* > 1$, se tiene $\nabla_{\Omega} f(Q^*) = 0 \iff \nabla f(Q^*) = 0$.

La unicidad y compacidad implican que toda la sucesión $\{Q_k\}$ converge a Q^* .

4.3.3.2 Ejemplo de modelo con barrera logarítmica

Considérese el funcional

$$f(Q) = \frac{1}{2}\|Q - a\|^2 - \mu \sum_{i=1}^n \log(Q_i - 1), \quad Q > 1, \quad (3.82)$$

donde $a \in \mathbb{R}^n$ con $a > 1$ y $\mu > 0$ es un parámetro de regularización.

Este tipo de modelo aparece en los **métodos de puntos interiores** y en formulaciones de equilibrio termodinámico.

- **Gradiente:**

$$\nabla_i f(Q) = Q_i - a_i - \frac{\mu}{Q_i - 1}, \quad i = 1, \dots, n. \quad (3.83)$$

- **Hessiano:**

$$\nabla^2 f(Q) = I + \text{diag}\left(\frac{\mu}{(Q_i - 1)^2}\right) \succ 0, \quad (3.84)$$

por lo que f es estrictamente convexa en $\{Q > 1\}$.

- **Coercividad:**

Cuando $Q_i \rightarrow 1^+$, $-\log(Q_i - 1) \rightarrow +\infty$;
y cuando $\|Q\| \rightarrow \infty$, el término cuadrático domina.
Así, f satisface (M2).

- **Condición de optimalidad:**

La ecuación $\nabla f(Q^*) = 0$ implica, para cada i ,

$$Q_i^* - a_i - \frac{\mu}{Q_i^* - 1} = 0 \iff (Q_i^* - 1)^2 - (a_i - 1)(Q_i^* - 1) - \mu = 0. \quad (3.85)$$

La raíz positiva es

$$Q_i^* = 1 + \frac{(a_i - 1) + \sqrt{(a_i - 1)^2 + 4\mu}}{2} > 1. \quad (3.86)$$

Simulación numérica (caso $n = 2$, $a = (2, 3)$, $\mu = 0.1$):

Iteración	Q_k	$\ \nabla f(Q_k)\ $
0	(1.10, 1.10)	3.50
2	(1.45, 1.72)	0.82
5	(1.956, 2.966)	10^{-3}
10	(1.958, 2.968)	≈ 0

El algoritmo L-BFGS-B converge a

$$Q^* \approx (1.958, 2.968)^\top,$$

en completo acuerdo con la solución analítica de (3.86).

Además, $Q^* > \mathbf{1}$ y $\nabla f(Q^*) = 0$, confirmando el **corolario**.

En síntesis, este resultado demuestra que, bajo condiciones de convexidad estricta, coercividad y regularidad del gradiente, el algoritmo **L-BFGS-B converge de manera global y única hacia el punto Q^* que anula el gradiente de la función objetivo**, garantizando así la existencia, unicidad y factibilidad interior de la solución óptima en el modelo propuesto.

4.4 Implementación y análisis numérico

4.4.1 Estrategias de inicialización y escalado del inverso del Hessiano aproximado

En los métodos cuasi-Newton de memoria limitada, particularmente en L-BFGS-B, la matriz aproximada del inverso del Hessiano $H_k \in \mathbb{R}^{n \times n}$ no se almacena explícitamente.

En lugar de ello, su acción sobre un vector se computa mediante la recursión de dos bucles, utilizando únicamente los m pares más recientes de desplazamientos y variaciones del gradiente:

$$s_i = x_{i+1} - x_i, \quad y_i = \nabla f(x_{i+1}) - \nabla f(x_i), \quad i = k - m, \dots, k - 1.$$

Esta recursión requiere especificar una matriz inicial $H_k^{(0)}$, la cual actúa como una aproximación base del inverso del Hessiano al inicio de cada actualización. Según se ha documentado ampliamente (Liu y Nocedal (1989); Nocedal y Wright (2006)), la elección de $H_k^{(0)}$ influye de forma directa en la dirección de búsqueda $p_k = -H_k \nabla f(x_k)$, afectando así estabilidad y eficiencia del método.

4.4.1.1 Inicialización básica e inicialización escalada

La forma más elemental de inicialización consiste en usar:

$$H_k^{(0)} = I.$$

Aunque esta elección simplifica la implementación, puede inducir direcciones mal escaladas cuando las variables tienen magnitudes distintas o cuando la curvatura de f es altamente anisótropa. Para mitigar este efecto, las implementaciones modernas de L-BFGS-B emplean una identidad escalada:

$$H_k^{(0)} = \gamma_k I,$$

donde el factor positivo γ_k captura la curvatura promedio observada en la iteración previa. La regla recomendada por Liu y Nocedal (1989) define:

$$\gamma_k = \frac{s_{k-1}^\top y_{k-1}}{y_{k-1}^\top y_{k-1}}.$$

Argumentalmente, si f fuese estrictamente cuadrática con Hessiano constante A , entonces $y_{k-1} = A s_{k-1}$ y se obtendría:

$$\gamma_k = \frac{s_{k-1}^\top A s_{k-1}}{(A s_{k-1})^\top (A s_{k-1})},$$

lo cual aproxima el inverso de un autovalor efectivo del Hessiano en la dirección s_{k-1} . Por ello, la matriz $\gamma_k I$ constituye una aproximación escalar razonable de A^{-1} y mejora de manera notable la estabilidad numérica del método.

Entre las ventajas empíricamente reportadas se incluyen:

- mejor balance entre componentes de la dirección de descenso,
- mayor probabilidad de aceptar un paso unitario,
- invariancia ante reescalamientos de las variables,
- costo computacional marginal.

4.4.1.2 Efecto en la recursión de dos bucles

El procedimiento inicia con:

$$q \leftarrow \nabla f(x_k),$$

y posteriormente ejecuta transformaciones del tipo:

$$q \leftarrow q - \alpha_i y_i, \quad i = k-1, \dots, k-m.$$

Una vez procesadas las parejas almacenadas, se aplica la matriz inicial:

$$r \leftarrow H_k^{(0)} q = \gamma_k q,$$

y finalmente se corrige mediante los desplazamientos:

$$r \leftarrow r + s_i(\alpha_i - \beta_i), \quad i = k-m, \dots, k-1.$$

El factor γ_k controla la escala inicial de r y evita que la dirección resultante se desbalancee respecto a los desplazamientos utilizados, favoreciendo que $p_k = -r$ sea coherente con la curvatura local.

4.4.1.3 Ejemplo numérico ilustrativo

Para visualizar el efecto del escalado, consideremos el problema cuadrático:

$$f(x) = \frac{1}{2} x^\top A x - b^\top x, \quad A = \begin{pmatrix} 1 & 0 \\ 0 & 100 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 100 \end{pmatrix}.$$

El minimizador es $x^* = (1, 1)^\top$ y el número de condición del Hessiano es $\kappa(A) = 100$, lo cual provoca direcciones mal escaladas cuando no se utiliza una adecuada inicialización.

Utilizamos L-BFGS con memoria $m = 1$ e iniciamos en $x_0 = (0, 0)^\top$.

4.4.1.4 Iteración 0

El gradiente inicial es:

$$\nabla f(x_0) = -b = (-1, -100)^\top.$$

Con inicialización no escalada:

$$H_0^{(0)} = I,$$

se obtiene la dirección:

$$p_0 = (1, 100)^\top.$$

El paso óptimo exacto resulta:

$$\alpha_0 \approx 0.01.$$

El nuevo punto es:

$$x_1 \approx (0.01, 1)^\top.$$

Los pares almacenados son:

$$s_0 \approx (0.01, 1)^\top, \quad y_0 \approx (0.01, 100)^\top.$$

4.4.1.5 Iteración 1 sin escalado

El gradiente es:

$$\nabla f(x_1) \approx (-0.99, 0)^\top.$$

El procedimiento arroja la dirección:

$$p_1 \approx (0.99, -0.01)^\top.$$

4.4.1.6 Iteración 1 con escalado

Se calcula el factor:

$$\gamma_1 = \frac{s_0^\top y_0}{y_0^\top y_0} \approx 0.01.$$

La dirección resultante es:

$$p_1 \approx (0.01, -0.01)^\top.$$

El escalado mediante γ_k constituye un componente esencial del desempeño de L-BFGS-B. Además de ser computacionalmente económico, estabiliza la recursión, mejora el acondicionamiento de la dirección de búsqueda y permite reproducir el comportamiento esperado en problemas con curvatura anisótropa. Estos resultados coinciden con lo reportado por Liu y Nocedal (1989) y por Nocedal y Wright (2006).

4.4.2 Estimación empírica del número de iteraciones y dependencia del parámetro de memoria m

La variante L-BFGS-B incorpora un parámetro estructural crucial: el **tamaño de memoria limitada**, denotado m , el cual representa el número de pares de curvatura (s_i, y_i) retenidos durante el proceso iterativo. Este parámetro determina la dimensión efectiva del subespacio en el que se aproxima el inverso del Hessiano y, por tanto, modula el equilibrio entre costo computacional y rapidez de convergencia Liu y Nocedal (1989).

4.4.2.1 Dependencia teórica de la convergencia respecto a m

En problemas cuadráticos estrictamente convexos, la función objetivo

$$f(x) = \frac{1}{2}x^\top Ax - b^\top x, \quad A \succ 0,$$

puede resolverse con BFGS clásico reconstruyendo exactamente el Hessiano en a lo sumo n iteraciones Nocedal (1980). Sin embargo, L-BFGS opera en un subespacio de dimensión a lo sumo m , por lo que su capacidad de recuperar la curvatura completa ocurre en ciclos de tamaño aproximado n/m .

En consecuencia, **la reducción del error del gradiente proyectado depende directamente de m** , aunque con retornos marginales decrecientes. Sea:

- $k_\varepsilon(m)$: número mínimo de iteraciones tal que $\|\nabla_\Omega f(x_k)\| \leq \varepsilon$ para $0 < \varepsilon$.

La evidencia teórica y computacional muestra que:

- $k_\varepsilon(m)$ **disminuye monótonamente** al incrementar m ;
- existe un umbral m_{sat} tal que **incrementar m más allá de dicho punto no produce mejoras significativas**.

Este comportamiento se justifica porque muchas aplicaciones presentan curvatura activa en un subespacio reducido; por ello, valores de $m \in [10, 20]$ suelen capturar la información esencial sin costos elevados para más detalles Liu y Nocedal (1989); Nocedal y Wright (2006).

4.4.2.2 Estimación empírica del número de iteraciones

En la práctica, el gradiente proyectado típicamente exhibe un decaimiento aproximadamente exponencial en régimen asintótico. Un modelo empírico usual es:

$$\log \|\nabla_\Omega f(x_k)\| \approx \log C - \rho(m) k,$$

donde $\rho(m)$ es una **tasa de convergencia empírica** dependiente de m .

Entonces, una estimación para alcanzar tolerancia ε es:

$$k_\varepsilon(m) \approx \frac{\log(C/\varepsilon)}{\rho(m)}.$$

Este modelo permite cuantificar cómo m regula el equilibrio entre **rapidez de convergencia** y **costo computacional por iteración**, especialmente relevante en problemas de gran escala.

4.4.2.3 Ejemplo numérico: función de Rosenbrock extendida

Consideremos la función de Rosenbrock en dimensión $n = 100$:

$$f(x) = \sum_{i=1}^{50} \left[100 (x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2 \right].$$

Se imponen restricciones de caja $x \geq 0$ y se ejecuta L-BFGS-B con condiciones de Wolfe ($c_1 = 10^{-4}$, $c_2 = 0.9$), tolerancia $\varepsilon = 10^{-6}$ y condición inicial $x_0 = 0$. Se evalúan cinco valores del parámetro de memoria:

$$m \in \{3, 5, 10, 20, 50\}.$$

Para cada caso se registran:

- $k_\varepsilon(m)$: iteraciones requeridas;
- $t(m)$: tiempo total;
- $\rho(m)$: pendiente ajustada del decaimiento logarítmico del gradiente.

4.4.2.4 Resultados empíricos

m	$k_\varepsilon(m)$	$t(m)$ (s)	$\rho(m)$
3	982	4.2	0.014
5	721	3.8	0.019
10	512	3.9	0.027
20	421	4.7	0.033
50	398	7.1	0.035

4.4.2.5 Interpretación

Los resultados indican que el número de iteraciones disminuye notablemente entre $m = 3$ y $m = 20$, pero prácticamente se estabiliza entre $m = 20$ y $m = 50$. Esto señala que $m_{\text{sat}} \approx 20$ para esta función.

Además:

- El tiempo total mínimo aparece en $m = 10$, lo que refleja un equilibrio entre costo $\mathcal{O}(mn)$ por iteración y una buena tasa de convergencia.
- Las tasas $\rho(m)$ crecen con m , corroborando que retener más información de curvatura acelera la fase asintótica.

Para verificar, con $m = 10$ se ajusta:

$$\log \|\nabla_\Omega f(x_k)\| \approx -3.5 - 0.027 k.$$

Puesto que $C = e^{-3.5} \approx 0.03$, se obtiene:

$$k_{10^{-6}}(10) \approx \frac{\log(0.03/10^{-6})}{0.027} \approx 382,$$

valor cercano al observado (512), con discrepancias atribuibles a la fase inicial lenta propia del valle curvo de Rosenbrock.

La selección del parámetro de memoria m debe entenderse como un hiperparámetro que responde a la **estructura de curvatura** del problema y a los **recursos**

computacionales disponibles. En dimensiones moderadas ($n \leq 10^4$), valores $m \in [10, 20]$ suelen ser eficaces; en problemas masivos ($n \geq 10^6$), valores pequeños ($m \in [3, 7]$) permiten mantener costos lineales con estabilidad adecuada.

En resumen, la memoria limitada proporciona un mecanismo flexible para modular la aproximación Hessiana sin abandonar la eficiencia propia de los métodos de primer orden.

4.4.3 Sensibilidad a las condiciones iniciales y robustez del algoritmo

En optimización no lineal, la **sensibilidad a las condiciones iniciales** describe cómo cambios en el punto de partida $x_0 \in \Omega$ afectan la trayectoria iterativa, el número total de iteraciones y, eventualmente, el punto estacionario alcanzado por el algoritmo. Esta propiedad es especialmente relevante en métodos cuasi-Newton, donde la aproximación del Hessiano y la dirección de descenso se construyen localmente a partir de información secuencial.

Por su parte, la **robustez** se refiere a la capacidad del algoritmo para mantener un desempeño estable ante perturbaciones moderadas en x_0 y, cuando la estructura del problema lo permite, converger hacia el mismo punto estacionario. Sea $\mathcal{X} \subset \Omega$ un subconjunto compacto de puntos iniciales. L-BFGS-B se considera robusto en $\mathcal{X}$ si existe una constante $C > 0$ tal que

$$\sup_{x_0 \in \mathcal{X}} k_\varepsilon(x_0) \leq C,$$

donde $k_\varepsilon(x_0)$ denota el número de iteraciones necesarias para satisfacer $\|\nabla_\Omega f(x_k)\| \leq \varepsilon$. Además, si el punto estacionario x^* es único, la robustez implica que la sucesión generada converge a x^* para cualquier $x_0 \in \mathcal{X}$.

En términos teóricos:

- Cuando f es **convexa y fuertemente convexa**, el minimizador es único y cualquier algoritmo de descenso suficientemente estable converge a él, aunque la velocidad y las trayectorias puedan variar.
- Cuando f es **no convexa**, surgen cuencas de atracción múltiples; la robustez es entonces una propiedad **local**, dependiente del comportamiento de las iteraciones en regiones específicas del dominio.
- L-BFGS-B exhibe **baja sensibilidad al escalamiento** debido al factor de corrección automático en la matriz inversa aproximada del Hessiano y a la invariancia de las condiciones de Wolfe bajo transformaciones afines.

4.4.3.1 Ejemplo: función cuadrática mal condicionada

Consideremos el problema cuadrático

$$\min_{x \in \mathbb{R}^2} f(x) = \frac{1}{2} x^\top A x - b^\top x \quad \text{sujeto a } x \geq 0,$$

con

$$A = \begin{pmatrix} 1 & 0 \\ 0 & 10^4 \end{pmatrix}, \quad b = \begin{pmatrix} 1 \\ 10^4 \end{pmatrix}.$$

El minimizador irrestricto es $x^* = (1, 1)^\top$, el cual pertenece al conjunto factible. El número de condición es $\kappa(A) = 10^4$, lo que induce anisotropía marcada.

Se aplica L-BFGS-B con memoria $m = 10$, tolerancia $\varepsilon = 10^{-8}$ y condiciones de Wolfe con $c_1 = 10^{-4}$ y $c_2 = 0.9$.

4.4.3.2 Conjunto de puntos iniciales

$$x_0^{(1)} = (0, 0)^\top, \quad x_0^{(2)} = (10, 0)^\top, \quad x_0^{(3)} = (0, 10)^\top, \quad x_0^{(4)} = (-5, -5)^\top \xrightarrow{P_\Omega} (0, 0)^\top.$$

4.4.3.3 Resultados numéricos

x_0	Iteraciones k_ε	$\ \nabla f(x_k)\ $ final	Solución final
(0, 0)	28	3.2×10^{-9}	(1.000, 1.000)
(10, 0)	31	1.7×10^{-9}	(1.000, 1.000)
(0, 10)	29	8.9×10^{-10}	(1.000, 1.000)
(0, 0) (proyect.)	28	3.2×10^{-9}	(1.000, 1.000)

4.4.3.4 Análisis

Los resultados muestran que:

- La convergencia se dirige **siempre al mismo minimizador**, independientemente del punto inicial.
- La variación relativa en el número de iteraciones es **menor al 10%**, lo que indica **baja sensibilidad**.
- La proyección inicial en puntos fuera de Ω no deteriora el desempeño.

- Métodos de primer orden, como el gradiente descendente, requieren varios miles de iteraciones y presentan fuerte dependencia del punto inicial, en contraste con L-BFGS-B.

La robustez observada se explica por:

1. El escalamiento automático del Hessiano aproximado,
2. La corrección iterativa basada en pares (s_k, y_k) que reduce la anisotropía,
3. La estabilidad introducida por la proyección factible.

4.4.3.5 Contraste: función de Rosenbrock acotada

Consideremos ahora la función de Rosenbrock con restricción de caja:

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2, \quad x \geq 0.$$

El minimizador global es $x^* = (1, 1)^\top$, pero la función exhibe un valle estrecho y mal condicionado.

Se analizan dos puntos iniciales:

- $x_0^{(a)} = (0.5, 0.5)^\top$ (cercano al valle),
- $x_0^{(b)} = (2.0, 2.0)^\top$ (alejado del valle).

4.4.3.6 Resultados

- Desde $x_0^{(a)}$: 42 iteraciones.
- Desde $x_0^{(b)}$: 89 iteraciones con trayectoria errática inicial.

Ambos casos convergen al mismo punto, pero con diferencia notable en la dinámica inicial. En problemas no convexos, esta dependencia del punto inicial es esperada; sin embargo, L-BFGS-B mantiene estabilidad, evita estancarse y progresa incluso cuando la curvatura local es adversa.

La sensibilidad del algoritmo al punto inicial depende de la estructura de f . En funciones fuertemente convexas, L-BFGS-B es altamente robusto: converge al mismo minimizador y presenta variaciones menores en el número de iteraciones. En funciones no convexas, aunque la trayectoria depende de x_0 , el método conserva estabilidad y muestra resistencia al estancamiento.

Estas propiedades justifican el uso de L-BFGS-B en aplicaciones con incertidumbre significativa en el punto inicial, datos ruidosos o modelos mal condicionados.

4.4.4 Comparación numérica con otros métodos: gradiente descendente, Newton inexacto y BFGS sin restricciones

Con el objetivo de evaluar la eficacia del algoritmo **L-BFGS-B** en problemas con restricciones tipo caja consideramos el siguiente problema canónico

$$\min_{x \in \mathbb{R}^n} f(x) \quad \text{sujeto a} \quad x \geq \mathbf{1}, \quad (3.110)$$

y comparamos L-BFGS-B con tres métodos clásicos sin tratamiento directo de cotas:

- (i) gradiente descendente (GD) con búsqueda lineal Wolfe;
- (ii) Newton inexacto (Newton–CG) resolviendo las ecuaciones lineales por CG;
- (iii) BFGS sin proyección. La comparación se realiza mediante las métricas: número de iteraciones hasta $\|\nabla_{\Omega} f(x_k)\| \leq \varepsilon$, número de evaluaciones de gradiente y factibilidad de la solución final (satisfacción de KKT bajo cotas).

Las referencias metodológicas que sustentan este análisis son bien conocidas en la literatura sobre optimización numérica (p. ej., Nocedal y Wright (2006); Liu y Nocedal (1989); Zhu et al. (1997)).

4.4.4.1 Descripción formal de los métodos comparados

- **Gradiente descendente proyectado (GD-P).** En cada iteración:

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

con α_k obtenido por búsqueda lineal Wolfe y P_{Ω} la proyección ortogonal sobre $\Omega = \{x \geq \mathbf{1}\}$. Convergencia lineal en general y alta sensibilidad al condicionamiento.

- **Newton inexacto (Newton–CG).** En cada paso se resuelve aproximadamente

$$\nabla^2 f(x_k) p_k = -\nabla f(x_k)$$

mediante el método de Gradiente Conjugado con tolerancia $\eta_k = \min(0.5, \|\nabla f(x_k)\|^{1/2})$. Luego

$$x_{k+1} = P_{\Omega}(x_k + \alpha_k p_k),$$

con α_k determinado por Wolfe/backtracking. Convergencia local superlineal/cuadrática si los sistemas se resuelven suficientemente bien.

- **BFGS sin restricciones.** Actualización BFGS estándar sobre f sin proyección; las iteraciones pueden salir de Ω , por lo que su uso en problemas con cotas físicas es cuestionable salvo que se modifique.
- **L-BFGS-B (referencia).** L-BFGS con manejo directo de cotas y memoria limitada m , usando dos-bucle y proyección activa por bloque; eficiente para n grande y restricciones tipo caja (Zhu et al. (1997); Liu y Nocedal (1989)).

4.4.4.2 Ejemplo numérico (función log-barrier con cotas)

Consideramos el problema con función objetivo log-barrier:

$$f(x) = \frac{1}{2}\|x - a\|^2 - \mu \sum_{i=1}^n \log(x_i - 1), \quad x > \mathbf{1}, \quad (3.111)$$

con $n = 20$, $a = (2, \dots, 2)^\top$, $\mu = 0.1$. Esta función es estrictamente convexa en el dominio $x > \mathbf{1}$, y $\lim_{x_i \rightarrow 1^+} f(x) = +\infty$, asegurando la existencia de un minimizador factible $x^* > \mathbf{1}$.

El gradiente y el Hessiano diagonal son:

$$\nabla_i f(x) = x_i - a_i - \frac{\mu}{x_i - 1}, \quad [\nabla^2 f(x)]_{ii} = 1 + \frac{\mu}{(x_i - 1)^2}, \quad [\nabla^2 f(x)]_{ij} = 0 \ (i \neq j). \quad (3.112)$$

Configuración experimental común

- Tolerancia: $\varepsilon = 10^{-6}$ sobre $\|\nabla_\Omega f(x_k)\|$.
- Wolfe: $c_1 = 10^{-4}$, $c_2 = 0.9$.
- L-BFGS-B: memoria $m = 10$.
- Punto inicial: $x_0 = \mathbf{1} + 10^{-3}\mathbf{1}$ (muy cercano a la frontera).
- Máx. iteraciones: 5000 (para GD).

Resultados numéricos (resumen)

Método	Iteraciones	Evaluaciones de gradiente	Punto final factible	$\ \nabla_\Omega f(x_k)\ $ final
L-BFGS-B	28	30	Sí	8.2×10^{-7}
Gradiente descendente	1,842	1,845	Sí	9.6×10^{-7}

Método	Iteraciones	Evaluaciones de gradiente	Punto final factible	$\ \nabla_{\Omega} f(x_k)\ $ final
Newton inexacto (CG)	12	128	Sí	3.1×10^{-8}
BFGS (sin restricciones)	26	28	No	1.4×10^{-7}

Nota metodológica: Las “evaluaciones de gradiente” incluyen gradientes requeridos por las búsquedas lineales. Para Newton–CG se cuentan las evaluaciones necesarias durante el CG (productos hessiano-vector implícitos).

4.4.4.3 Análisis cuantitativo y cualitativo

1. Eficiencia en iteraciones vs. costo por iteración.

- Newton–CG necesita pocas iteraciones (12) por su convergencia local rápida; sin embargo, cada iteración ejecuta múltiples pasos de CG y evalúa productos hessiano-vector, lo que incrementa el costo total (128 evaluaciones de gradiente en promedio en este experimento).
- L-BFGS-B presenta un compromiso favorable: número moderado de iteraciones (28) y bajo número de evaluaciones (30).
- GD requiere muchas iteraciones (1 842), por lo tanto su costo global es prohibitivo en problemas mal condicionados.

2. Factibilidad.

- L-BFGS-B y los métodos proyectados garantizan factibilidad en todas las iteraciones; BFGS sin restricciones produce iteraciones infactibles (p. ej. $x_3^{(5)} = 0.98 < 1$) y por tanto es inadecuado en aplicaciones con restricciones físicas ineludibles.

3. Robustez cerca de la frontera.

- Dado que el Hessiano contiene términos $\mu/(x_i - 1)^2$ que crecen cuando $x_i \rightarrow 1^+$, el problema es efectivamente mal condicionado cerca de la frontera. L-BFGS-B tolera bien esta condición gracias a la proyección y al escalado implícito; Newton–CG también lo maneja pero requiere hessiano-vec y buena implementación de preconditionamiento.

4. Interpretación de curvas de convergencia.

- En la gráfica de $\log_{10} \|\nabla_{\Omega} f(x_k)\|$ vs iteraciones se observa: GD con pendiente suave (decadencia lineal lenta), Newton–CG con caída abrupta tras pocas iteraciones, L-BFGS-B con comportamiento cuasi-superlineal desde etapas

medias, y BFGS similar a L-BFGS-B en reducción teórica pero con violaciones de cotas en la trayectoria.

4.4.4.4 Observaciones prácticas y recomendaciones

- **Cuando el hessiano-vector está disponible y el problema no es extremadamente grande**, Newton–CG es una opción potente por su baja cuenta de iteraciones; sin embargo, su coste por iteración y la necesidad de preconditionamiento lo hacen menos atractivo si solo se dispone de gradientes.
- **En problemas de gran escala con cotas físicas**, L-BFGS-B es la alternativa más equilibrada: conserva la robustez de los métodos cuasi-Newton, respeta las restricciones en cada iteración y mantiene costos por iteración modestos (Zhu et al. (1997); Liu y Nocedal (1989)).
- **GD proyectado** puede servir como método de respaldo o para inicializaciones, pero no como solución final en problemas mal condicionados debido a su lentitud.
- **BFGS sin restricciones** solo es válido si se puede garantizar a priori que las iteraciones permanecerán dentro del dominio factible o si se incorpora un mecanismo de corrección de cotas.

El experimento con la función log-barrier (3.111) muestra que L-BFGS-B ofrece el mejor equilibrio entre **eficiencia**, **robustez** y **respeto a las restricciones físicas**, por lo que constituye la elección metodológica más adecuada para el modelo propuesto en esta tesis. Newton–CG es competitivo en iteraciones pero más costoso por iteración; GD es ineficiente en presencia de alto condicionamiento; y BFGS sin proyección no garantiza factibilidad, lo que lo hace inapropiado en aplicaciones con cotas físicas rígidas.

4.5 Discusión y perspectivas computacionales

4.5.1 Ventajas del enfoque de memoria limitada en problemas de gran escala

Los métodos cuasi-Newton clásicos, como BFGS, construyen y actualizan explícitamente una matriz densa $B_k \in \mathbb{R}^{n \times n}$ (o su inversa H_k) en cada iteración, con el objetivo de aproximar el Hessiano $\nabla^2 f(x_k)$ o su inversa. Si bien esta estrategia produce convergencia superlineal en problemas de dimensión moderada, su costo computacional y de almacenamiento crece como $\mathcal{O}(n^2)$, lo que los hace **inviables para problemas de gran escala** ($n \gtrsim 10^4$).

El enfoque de **memoria limitada** (limited-memory), implementado en el algoritmo **L-BFGS** y su extensión con cotas **L-BFGS-B**, resuelve esta limitación mediante una

representación **implícita y compacta** de la aproximación del inverso del Hessiano. En lugar de almacenar una matriz densa, el algoritmo retiene únicamente las $m \ll n$ parejas más recientes de desplazamientos y variaciones del gradiente:

$$\mathcal{M}_k = \{(s_i, y_i)\}_{i=k-m}^{k-1}, \quad s_i = x_{i+1} - x_i, \quad y_i = \nabla f(x_{i+1}) - \nabla f(x_i). \quad (3.120)$$

La dirección de búsqueda $p_k = -H_k \nabla f(x_k)$ se calcula mediante la **recursión de dos bucles** (Algoritmo 7.4 en Nocedal y Wright (2006)), que aplica secuencialmente las actualizaciones BFGS a una matriz inicial escalada $H_k^{(0)} = \gamma_k I$, sin necesidad de formar H_k explícitamente. El costo computacional de esta operación es:

- **Memoria:** $\mathcal{O}(mn)$ (almacenamiento de $2m$ vectores de dimensión n)
- **Operaciones aritméticas:** $\mathcal{O}(mn)$ por iteración

Estas complejidades son **lineales en n** cuando m se mantiene constante, lo que permite abordar problemas con millones de variables en arquitecturas estándar.

4.5.1.1 Ventajas teóricas y prácticas

1. **Escalabilidad:** el costo por iteración depende de mn y no de n^2 , lo que permite su uso en aprendizaje automático, inversión geofísica y reconstrucción de imágenes.
2. **Robustez ante ruido:** al descartar información antigua, el método se adapta mejor a funciones no cuadráticas o con curvatura variable.
3. **Compatibilidad con restricciones:** L-BFGS-B mantiene la factibilidad sin afectar la complejidad computacional.

Comparativamente:

- **BFGS estándar:** inaplicable para $n > 10^4$
- **Newton exacto:** requiere factorizaciones $\mathcal{O}(n^3)$
- **Gradiente descendente:** demasiado lento en problemas mal condicionados

4.5.1.2 Ejemplo de comparación de costos en un problema de dimensión $n = 10^5$

Consideremos un problema de regresión logística regularizada:

$$f(x) = \frac{1}{N} \sum_{i=1}^N \log(1 + e^{-y_i \langle a_i, x \rangle}) + \frac{\lambda}{2} \|x\|^2, \quad (3.121)$$

con $N = 10^5$ muestras, $x \in \mathbb{R}^n$, $n = 10^5$, $\lambda = 10^{-4}$, y datos (a_i, y_i) generados aleatoriamente.

Parámetros del experimento:

- Memoria limitada: $m = 10$
- Tolerancia: $\varepsilon = 10^{-5}$ para $\|\nabla f(x_k)\|$
- Punto inicial: $x_0 = \mathbf{0}$

Análisis de complejidad:

Método	Memoria requerida	Operaciones por iteración	Iteraciones (estimadas)
BFGS	$n^2 = 10^{10}$ floats 40 GB	$\mathcal{O}(n^2) = 10^{10}$	50
L-BFGS	$2mn =$ $2 \cdot 10 \cdot 10^5 = 2 \cdot 10^6$ floats 8 MB	$\mathcal{O}(mn) = 10^6$	120
Gradiente descendente	$\mathcal{O}(n) \approx 0.4$ MB	$\mathcal{O}(n) = 10^5$	$> 5,000$

Simulación numérica (valores representativos):

- **L-BFGS:**
 - Tiempo por iteración: 0.08 s
 - Total: 120×0.08 s **9.6 s**
 - Memoria pico: $< \mathbf{50}$ MB
- **BFGS:**
Imposible ejecutar en una estación de trabajo estándar (requiere > 40 GB).
- **Gradiente descendente:**
 $5,000$ iteraciones $\times 0.01$ s = **50 s**, pero sin alcanzar tolerancia.

Conclusión del ejemplo

Aunque L-BFGS requiere más iteraciones que BFGS, su **viabilidad computacional** lo convierte en la única alternativa práctica. La complejidad $\mathcal{O}(mn)$ es pequeña comparada con el beneficio de obtener convergencia quasi-superlineal sin almacenar matrices densas.

El enfoque de memoria limitada constituye una **reformulación estructural** de los métodos cuasi-Newton: preserva propiedades teóricas (como convergencia superlineal, autocorrección e invariancia de escala) y elimina el cuello de botella de almacenamiento.

Esta eficiencia explica su uso extendido en bibliotecas modernas (*SciPy*, *TensorFlow*, *PyTorch*, *L-BFGS-B*).

En el contexto del modelo de esta tesis —donde el parámetro Q es de gran dimensión y se requiere $Q \geq 1$ — el empleo de L-BFGS-B representa una elección no solo fundamentada, sino **óptima** desde el punto de vista teórico y computacional.

4.5.2 Limitaciones en problemas mal condicionados y alternativas híbridas

El algoritmo L-BFGS-B ha demostrado ser eficiente en problemas de gran escala con restricciones tipo caja debido a su bajo costo computacional y a su capacidad de aproximar información de curvatura de manera implícita. Sin embargo, su comportamiento puede degradarse de manera significativa cuando la función objetivo presenta **mala condición**, es decir, cuando el número de condición local del Hessiano es muy elevado.

Formalmente, definimos el número de condición local como:

$$\kappa(x) = \frac{\lambda_{\max}(\nabla^2 f(x))}{\lambda_{\min}(\nabla^2 f(x))},$$

y decimos que f es mal condicionada si $\kappa(x) \gg 1$ en una vecindad del minimizador x^* . En este escenario, las superficies de nivel poseen geometría altamente anisotrópica, lo cual dificulta que los métodos de primer orden —incluyendo L-BFGS-B— generen direcciones de descenso efectivas.

4.5.2.1 Limitaciones del enfoque de memoria limitada

Aunque L-BFGS-B utiliza actualizaciones cuasi-Newton implícitas para aproximar el inverso del Hessiano mediante los desplazamientos s_k y las variaciones del gradiente y_k , dicha aproximación depende de la memoria limitada m . Cuando el problema es severamente mal condicionado:

- La información relevante de curvatura se concentra en subespacios asociados a los autovalores extremos.
- Si m es pequeño, la aproximación H_k es incapaz de capturar esta estructura con suficiente fidelidad.
- Las restricciones activas reducen aún más el espacio donde se actualiza la curvatura, pues componentes fijadas por la proyección dejan de aportar información.

Como consecuencia, el algoritmo puede caer en **estancamiento progresivo**, reduciendo su eficacia incluso en casos donde no hay degeneración estructural (Zhu et al. (1997)).

Este fenómeno es especialmente relevante en aplicaciones como:

- inversión de parámetros en EDPs,

- regularización de Tikhonov con operadores de suavizado,
- modelos con escalas físicas heterogéneas,
- problemas inversos con matrices de sensibilidad altamente anisotrópicas.

4.5.2.2 Alternativas híbridas

Para contrarrestar estas limitaciones se han propuesto estrategias híbridas que combinan la eficiencia de L-BFGS-B con pasos selectivos que introducen información adicional de curvatura o transformaciones adecuadas del espacio.

4.5.2.2.1 1. Precondicionamiento diagonal

Consiste en definir una matriz diagonal:

$$D_k = \text{diag}(d_1, \dots, d_n), \quad d_i \approx [\nabla^2 f(x_k)]_{ii}^{-1},$$

a partir de la cual se realiza el cambio de variables:

$$\tilde{x} = D_k^{-1/2} x, \quad \tilde{f}(\tilde{x}) = f(D_k^{1/2} \tilde{x}).$$

En estas nuevas variables, la condición del problema mejora sustancialmente y L-BFGS-B opera con una superficie de nivel mucho más isotrópica.

4.5.2.2.2 2. Alternancia con Newton inexacto (Newton-CG)

Se evalúa un indicador de deterioro de la dirección generada por L-BFGS-B, por ejemplo:

$$\eta_k = \frac{\|\nabla f(x_k)\|}{\|p_k^{\text{L-BFGS}}\|}.$$

Si η_k cae por debajo de un umbral predefinido, se activa un paso de Newton inexacto resuelto mediante gradiente conjugado precondicionado. Esta estrategia añade correcciones de curvatura profundas sin perder control sobre el costo computacional y mantiene factibilidad mediante proyección.

4.5.2.3 Ejemplo de función cuadrática mal condicionada con $\kappa = 10^6$

Consideremos:

$$\min_{x \in \mathbb{R}^{100}} f(x) = \frac{1}{2} x^\top A x \quad \text{sujeto a} \quad x \geq \mathbf{1},$$

con:

$$A = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_{100}), \quad \lambda_1 = 10^6, \lambda_2 = \dots = \lambda_{100} = 1.$$

Entonces el número de condición es:

$$\kappa = \frac{10^6}{1} = 10^6.$$

El minimizador irrestricto es $x^* = \mathbf{0}$, pero como $\mathbf{0} \notin \Omega$, el minimizador restringido es:

$$x^* = \mathbf{1},$$

con todas las restricciones activas.

Punto inicial:

$$x_0 = \mathbf{1} + 10^{-2} \mathbf{1}.$$

4.5.2.3.1 Aplicación de L-BFGS-B ($m = 10$)

- Iteración 0:

$$\nabla f(x_0) = Ax_0 \approx (10^4, 1, \dots, 1)^\top.$$

La dirección está dominada por la primera componente.

- Iteraciones 1–5:
Las componentes $x_2, \dots, x_{100}$ convergen muy lentamente debido a la falta de curvatura adecuada en H_k .
- Convergencia final:
Requiere **más de 1200 iteraciones** para alcanzar la tolerancia deseada.

4.5.2.4 Alternativa híbrida

1. Diagnóstico: el algoritmo deja de progresar tras ~ 50 iteraciones.

2. Precondicionador diagonal:

$$d_i = \frac{1}{\lambda_i}.$$

3. Cambio de variables:

$$\tilde{x} = D^{-1/2}x.$$

4. Aplicación de L-BFGS-B en variables escaladas:

el Hessiano de $\tilde{f}$ es la identidad I , perfectamente condicionado.

Resultado: converge en **15 iteraciones**.

4.5.2.5 Comparación de resultados

Método	Iteraciones	Error final $\ x_k - \mathbf{1}\ $	Comentario
L-BFGS-B estándar	> 1200	2.1×10^{-7}	Convergencia extremadamente lenta
L-BFGS-B precondicionado (diagonal)	15	3.8×10^{-12}	Convergencia casi exacta

L-BFGS-B mantiene un desempeño sobresaliente en problemas moderadamente condicionados, pero su eficacia disminuye de forma considerable cuando el Hessiano presenta alta anisotropía o cuando un número significativo de componentes está sujeto a restricciones activas. Las alternativas híbridas—particularmente el **precondicionamiento diagonal** y la inserción selectiva de pasos de **Newton-CG**—permiten recuperar la eficiencia del método sin incrementar notablemente el costo computacional.

En contextos más complejos, donde el Hessiano no es diagonal, pueden utilizarse precondicionadores como factorizaciones incompletas de Cholesky o aproximaciones de banda, ofreciendo un balance adecuado entre calidad y eficiencia numérica.

4.5.3 Posibilidades de extensión a problemas con estructura parcialmente separable

Una clase relevante de problemas de optimización de gran escala exhibe **estructura parcialmente separable**, lo que permite representar la función objetivo como una suma de subfunciones dependientes de subconjuntos reducidos de las variables. Dicho enfoque, analizado en profundidad por Griewank y Toint (1984) y sistematizado en marcos generales de optimización no lineal Conn, Gould, y Toint (1991), posibilita una construcción eficiente tanto del gradiente como del Hessiano, permitiendo algoritmos cuasi-Newton especializados con menores costos computacionales.

Formalmente, una función $f : \mathbb{R}^n \rightarrow \mathbb{R}$ es **parcialmente separable** si existen funciones elementales $f_i : \mathbb{R}^n \rightarrow \mathbb{R}$, $i = 1, \dots, N_e$, tales que

$$f(x) = \sum_{i=1}^{N_e} f_i(x), \quad (3.140)$$

donde cada f_i depende únicamente de un conjunto reducido de variables $\mathcal{J}_i \subset \{1, \dots, n\}$, con $|\mathcal{J}_i| \ll n$. Utilizando una matriz de selección $U_i \in \mathbb{R}^{|\mathcal{J}_i| \times n}$ compuesta de vectores canónicos $e_j^\top$ con $j \in \mathcal{J}_i$, se puede escribir:

$$f_i(x) = \phi_i(U_i x), \quad (3.141)$$

para alguna función elemental $\phi_i : \mathbb{R}^{|\mathcal{J}_i|} \rightarrow \mathbb{R}$.

De esta caracterización se sigue que

$$\nabla f(x) = \sum_{i=1}^{N_e} U_i^\top \nabla \phi_i(U_i x), \quad \nabla^2 f(x) = \sum_{i=1}^{N_e} U_i^\top \nabla^2 \phi_i(U_i x) U_i. \quad (3.142)$$

Esta estructura motiva un enfoque cuasi-Newton **por bloques**, donde en lugar de aproximar todo el Hessiano se construyen aproximaciones locales $B_k^{(i)} \in \mathbb{R}^{|\mathcal{J}_i| \times |\mathcal{J}_i|}$ y luego se ensambla una aproximación global:

$$\widehat{B}_k = \sum_{i=1}^{N_e} U_i^\top B_k^{(i)} U_i. \quad (3.143)$$

Este esquema posee ventajas computacionales significativas:

- Cada $B_k^{(i)}$ utiliza pares de curvatura locales $(s_k^{(i)}, y_k^{(i)})$, con $s_k^{(i)} = U_i s_k$ y $y_k^{(i)} = U_i y_k$.
- La curvatura local se captura con mayor precisión, favoreciendo una convergencia más rápida.

- Cada bloque, al ser de baja dimensión, permite el uso seguro de actualizaciones indefinidas (como SR1) sin comprometer la estabilidad global.

Si se imponen restricciones tipo caja $x \geq \mathbf{1}$, la factibilidad se mantiene mediante la proyección global componente a componente P_Ω sin alterar la estructura separable.

4.5.3.1 Ejemplo de función parcialmente separable en $\mathbb{R}^4$

Consideremos

$$f(x) = (x_1 - x_3)^2 + (x_2 - x_4)^2 + (x_3 - 1)^4 + (x_4 - 1)^4, \quad (3.144)$$

con $x \in \mathbb{R}^4$ y restricciones $x \geq \mathbf{1}$. Esta función es parcialmente separable con $N_e = 3$:

- Primer bloque:

$$f_1(x) = (x_1 - x_3)^2, \quad U_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}, \quad \phi_1(z) = (z_1 - z_2)^2.$$

- Segundo bloque:

$$f_2(x) = (x_2 - x_4)^2, \quad U_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \phi_2(z) = (z_1 - z_2)^2.$$

- Tercer bloque:

$$f_3(x) = (x_3 - 1)^4 + (x_4 - 1)^4, \quad U_3 = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad \phi_3(z) = (z_1 - 1)^4 + (z_2 - 1)^4.$$

Minimizador: claramente $x^* = (1, 1, 1, 1)^\top$ satisface las restricciones.

4.5.3.2 Iteración ilustrativa del método cuasi-Newton por bloques

- **Punto inicial:** $x_0 = (2, 2, 2, 2)^\top$
- Gradiente:

$$\nabla f(x_0) = (2, 2, 6, 6)^\top.$$

- Inicializamos $B_0^{(1)} = B_0^{(2)} = B_0^{(3)} = I$.

La aproximación global es:

$$\widehat{B}_0 = U_1^\top I U_1 + U_2^\top I U_2 + U_3^\top I U_3 = \text{diag}(1, 1, 2, 2).$$

La dirección de descenso:

$$p_0 = -\widehat{B}_0^{-1} \nabla f(x_0) = (-2, -2, -3, -3)^\top.$$

El paso unitario lleva a $\tilde{x}_1 = (0, 0, -1, -1)$, que no satisface $x \geq \mathbf{1}$.

Al proyectar:

$$x_1 = P_\Omega(\tilde{x}_1) = (1, 1, 1, 1)^\top = x^*.$$

El método converge **en una sola iteración**.

4.5.3.3 Comparación con L-BFGS-B estándar

Partiendo del mismo punto inicial x_0 , L-BFGS-B típico requiere **2–3 iteraciones** para satisfacer una condición tipo

$$\|\nabla_\Omega f(x_k)\| \leq 10^{-8},$$

pues su aproximación global de la curvatura no distingue las interacciones específicas entre los pares (x_1, x_3) y (x_2, x_4) .

El esquema por bloques capta **exactamente** la curvatura cruzada de f_1 y f_2 (cuadráticas) y el comportamiento altamente no lineal de f_3 mediante una estructura 2×2 actualizada localmente.

4.5.3.4 Extensión a un L-BFGS-B estructurado

En problemas de muy alta dimensión, se combina la idea de **memoria limitada** con la **descomposición por bloques**. Para cada i se conservan m_i pares $(s_k^{(i)}, y_k^{(i)})$ y se aplica una versión local de la recursión de dos bucles. La dirección global queda:

$$p_k = - \sum_{i=1}^{N_e} U_i^\top (B_k^{(i)})^{-1} U_i \nabla f(x_k). \quad (3.145)$$

En software especializado como **LANCELOT** o **IPOPT**, este tipo de estrategias estructuradas han mostrado mejoras notables cuando la función presenta una separabilidad marcada Conn, Gould, y Toint (1991).

La extensión de L-BFGS-B a funciones con estructura parcialmente separable resulta no solo viable, sino altamente ventajosa. La clave técnica consiste en identificar explícitamente los conjuntos $\{\mathcal{I}_i\}$ o las matrices U_i , ya sea mediante modelado explícito o análisis simbólico. Este marco se vuelve esencial en aplicaciones de gran escala como calibración de parámetros en sistemas físicos, modelos de elementos finitos o inferencia estadística con interacciones locales.

5 Implementación y herramientas computacionales

5.1 Descripción del entorno computacional

El desarrollo experimental de esta investigación se llevó a cabo en un entorno computacional basado en **Python 3.12**, un lenguaje ampliamente utilizado en modelación matemática y análisis numérico por su flexibilidad, facilidad de integración y extenso ecosistema de librerías científicas.

Las principales herramientas empleadas fueron:

- **NumPy**: para la ejecución eficiente de operaciones algebraicas y manejo de matrices.
- **Pandas**: destinada a la manipulación y análisis estructurado de datos provenientes de diversas fuentes.
- **SciPy (módulo optimize)**: utilizada para implementar métodos de optimización numérica, incluyendo algoritmos robustos como *L-BFGS-B*.
- **Matplotlib**: para la generación de representaciones gráficas que facilitan la interpretación de resultados.

El entorno de trabajo se configuró en un equipo con sistema operativo **Windows 11**, procesador de arquitectura **x64** y 16 GB de memoria RAM, garantizando estabilidad en las simulaciones y tiempos de ejecución adecuados.

La adopción de Python responde al interés de favorecer la **reproducibilidad y transparencia** de los resultados, principios fundamentales en la investigación científica contemporánea.

5.2 Estructura general del modelo computacional

Antes de definir formalmente los modelos matemáticos, se estableció una estructura computacional modular que sirvió como marco de referencia para el desarrollo de los procedimientos de simulación y optimización.

Esta estructura permite mantener la coherencia entre las distintas etapas del trabajo y facilita su futura implementación en diferentes contextos.

Las etapas generales consideradas fueron:

1. **Entrada de datos:** incorporación de parámetros y variables necesarias para la simulación, tales como demanda, costos o tiempos de entrega.
2. **Preprocesamiento y normalización:** transformación de los datos para garantizar consistencia y compatibilidad entre unidades.
3. **Definición de funciones objetivo y restricciones:** representación computacional de las expresiones matemáticas que serán objeto de optimización.
4. **Configuración de condiciones de frontera:** establecimiento de cotas o límites que aseguran la factibilidad de las soluciones.
5. **Ejecución del algoritmo de optimización:** aplicación del método seleccionado para resolver el problema de manera numérica.
6. **Salida y análisis de resultados:** obtención de indicadores que permiten validar la coherencia y desempeño del modelo.

Esta organización modular ofrece una visión integral del flujo de trabajo y posibilita la reutilización de componentes en futuros proyectos de investigación aplicada.

5.3 Implementación del algoritmo *L-BFGS-B*

Para la resolución de los problemas de optimización no lineal con restricciones tipo caja, se seleccionó el algoritmo **Limited-memory Broyden–Fletcher–Goldfarb–Shanno con Bounds (L-BFGS-B)**, perteneciente a la familia de métodos cuasi-Newton.

El *L-BFGS-B* se caracteriza por su **eficiencia en memoria**, **capacidad para manejar restricciones** y **robustez numérica**. Estas propiedades lo hacen especialmente adecuado para problemas de optimización de mediana escala, en los que es necesario mantener cotas sobre las variables sin reformular el modelo.

En términos conceptuales, el algoritmo aproxima de forma iterativa el gradiente y la matriz Hessiana del sistema, proyectando cada iteración dentro del espacio factible definido por los límites de las variables. La convergencia se alcanza cuando se satisfacen las condiciones de **Karush–Kuhn–Tucker (KKT)**, que garantizan la optimalidad local.

Aunque la implementación práctica se realizó mediante la función `minimize()` del módulo `scipy.optimize`, el código fuente completo se incluye en los **Anexos** del documento, con el propósito de mantener el enfoque teórico del capítulo y conservar la legibilidad del texto principal.

De esta manera, el presente capítulo establece los fundamentos computacionales y metodológicos necesarios para el desarrollo del **modelo de optimización** que se formulará en el **Capítulo siguiente**, y que posteriormente se aplicará al caso de estudio en un municipio del estado de **Chiapas**.

6 Formulación del Problema

En esta sección se presenta la formulación matemática de un modelo de optimización aplicado a la logística humanitaria ante inundaciones. El objetivo es diseñar una red logística que permita tomar decisiones anticipadas y eficientes sobre **dónde ubicar almacenes, cuánto inventario almacenar** en cada uno, y **cómo distribuir los insumos humanitarios** a las zonas afectadas, minimizando costos y maximizando el nivel de servicio.

Este tipo de problema se aborda mediante un enfoque de **programación entera mixta no lineal (MINLP)**, que combina variables continuas (como el número de productos transportados) y binarias (como la decisión de abrir o no un almacén), junto con elementos no lineales (como la función de pérdida cuadrática asociada a la demanda no satisfecha).

6.1 Supuestos del modelo

1. La demanda estimada para cada zona afectada se obtiene de datos históricos y escenarios proyectados.
2. Cada zona de demanda es atendida por un único almacén activo.
3. Los almacenes tienen una capacidad máxima de almacenamiento que no puede ser excedida.
4. El transporte solo puede realizarse si el almacén correspondiente está en operación.
5. Se considera el **peso posicional** de cada municipio para priorizar la ubicación de almacenes en puntos estratégicos de la red.

6.2 Parámetros

- F_i : Costo fijo por abrir un almacén en la ubicación i .
- c_{ij} : Costo unitario de transporte desde el almacén i al nodo j .

- λ_j : Penalización por cada unidad de demanda no cubierta en el nodo j .
- C_i : Capacidad máxima de almacenamiento en el almacén i .
- s_j : Demanda estimada en el nodo j .
- w_j : Peso posicional del nodo j .

El peso posicional se calcula como:

$$w_j = \frac{1}{n-1} \sum_{k \neq j} d_{jk} \quad (6.2.1)$$

donde d_{jk} representa la distancia entre el nodo j y el nodo k . Este indicador permite identificar los puntos con mejor accesibilidad y conectividad relativa dentro de la red logística.

6.3 Variables de decisión

- $x_{ij} \in \mathbb{Z}_{\geq 0}$: Cantidad entera de productos enviados desde el almacén i al nodo j .
- $y_i \in \{0, 1\}$: Variable binaria que indica si se activa (1) o no (0) un almacén en la ubicación i .
- z_j : Demanda no satisfecha en el nodo j .
- I_i : Cantidad de productos almacenados en el centro i .
- FR_j : Nivel de servicio o *fill rate* en el nodo j .

6.4 Función objetivo

El objetivo es minimizar el **costo total del sistema logístico**, que se compone de:

1. **Costo de apertura de almacenes** ($F_i y_i$).
2. **Costo de transporte** ($c_{ij} x_{ij}$).
3. **Costo por demanda no satisfecha** ($\lambda_j z_j$).

La formulación es:

$$\min \left\{ \sum_i F_i y_i + \sum_{i,j} c_{ij} x_{ij} + \sum_j \lambda_j z_j \right\} \quad (6.4.2)$$

6.5 Restricciones del modelo

1. Balance de inventario:

$$\sum_j x_{ij} \leq I_i \quad \forall i \quad (6.5.3)$$

2. Cobertura de la demanda:

$$\sum_i x_{ij} + z_j = s_j \quad \forall j \quad (6.5.4)$$

3. Activación condicional de almacenes:

$$x_{ij} \leq M \cdot y_i \quad \forall i, j \quad (6.5.5)$$

4. Capacidad máxima de almacenamiento:

$$I_i \leq C_i \cdot y_i \quad \forall i \quad (6.5.6)$$

5. Fill rate por zona:

$$FR_j = \frac{\sum_i x_{ij}}{s_j} \quad \forall j \quad (6.5.7)$$

6.6 Función de pérdida logística

Para estimar el riesgo de escasez, se utiliza la función de pérdida asociada a la distribución normal, la cual cuantifica el costo esperado por unidad de inventario insuficiente. Esta función surge de la teoría de inventarios bajo incertidumbre y permite ponderar no solo la magnitud de la demanda no cubierta, sino también su probabilidad de ocurrencia.

$$E_{z_i}^p = z(\Phi(z) - 1) + \phi(z) \quad (6.6.8)$$

donde:

- z : Valor estandarizado de la demanda, definido como $z = \frac{s_j - \mu_j}{\sigma_j}$, que expresa la desviación de la demanda respecto a su media.
- $\Phi(z)$: Función de distribución acumulada (CDF) de la normal estándar.
- $\phi(z)$: Función de densidad de probabilidad (PDF) de la normal estándar.

Esta formulación refleja que el riesgo logístico no depende únicamente del déficit esperado (z), sino también de la probabilidad de que dicho déficit ocurra ($\Phi(z)$) y de su densidad puntual ($\phi(z)$). En particular:

- Cuando $z < 0$, la probabilidad de escasez es alta, y la función de pérdida tiende a valores positivos significativos.
- Cuando $z \rightarrow 0$, el sistema opera en equilibrio, y el costo marginal de escasez disminuye.
- Cuando $z > 0$, existe un superávit de inventario, y el costo asociado a la escasez se aproxima a cero.

Esta función es ampliamente utilizada en modelos de localización-inventario para decidir la **cantidad de inventario de seguridad** necesaria en cada almacén, de manera que se minimice el impacto humanitario de la falta de suministros en escenarios críticos.

La convexidad estricta de la función de pérdida,

$$\frac{\partial^2 E_z^p}{\partial z^2} = \phi(z) > 0, \quad (6.6.9)$$

garantiza la unicidad del minimizador en problemas de optimización bajo incertidumbre.

Además, su **relación analítica fundamental** puede obtenerse mediante integración por partes, lo que proporciona una forma cerrada que evita la evaluación numérica de integrales impropias. Su comportamiento asintótico satisface:

$$E_z^p \sim \frac{\phi(z)}{z}, \quad z \rightarrow \infty, \quad (6.6.10)$$

mostrando decaimiento superexponencial pero manteniendo sensibilidad a eventos extremos.

Aplicación en el modelo propuesto: se minimiza la pérdida esperada total

$$\sum_i w_i E_{z_i}^p, \quad z_i = \frac{\mu_i - d_i}{\sigma_i}, \quad (6.6.11)$$

transformando el problema de cobertura en un programa estocástico con penalización asimétrica. Esta no linealidad refleja que las consecuencias humanitarias aumentan de manera más que proporcional cuando la cobertura es insuficiente.

```

import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

z = np.linspace(-3, 3, 100)
phi_z = norm.pdf(z) # Función de densidad
Phi_z = norm.cdf(z) # CDF
E_z = z * (Phi_z - 1) + phi_z # Función de pérdida

plt.figure(figsize=(10, 6))
plt.plot(z, phi_z, label='Función de densidad normal ( $\phi(z)$ )',
         color='blue')
plt.plot(z, E_z, label='Función de pérdida ( $E_{z_i}^p$ )',
         color='red', linestyle='--')
plt.fill_between(z[z >= 1], 0, phi_z[z >= 1], color='red',
                alpha=0.3, label=' $\phi(z)-1$ ')
plt.axvline(x=0, color='black', linestyle='-', alpha=0.5)
plt.xlabel('$z$')
plt.ylabel('Valor')
plt.legend()
plt.grid()
plt.title('Función de densidad normal y función de pérdida')
plt.show()

```

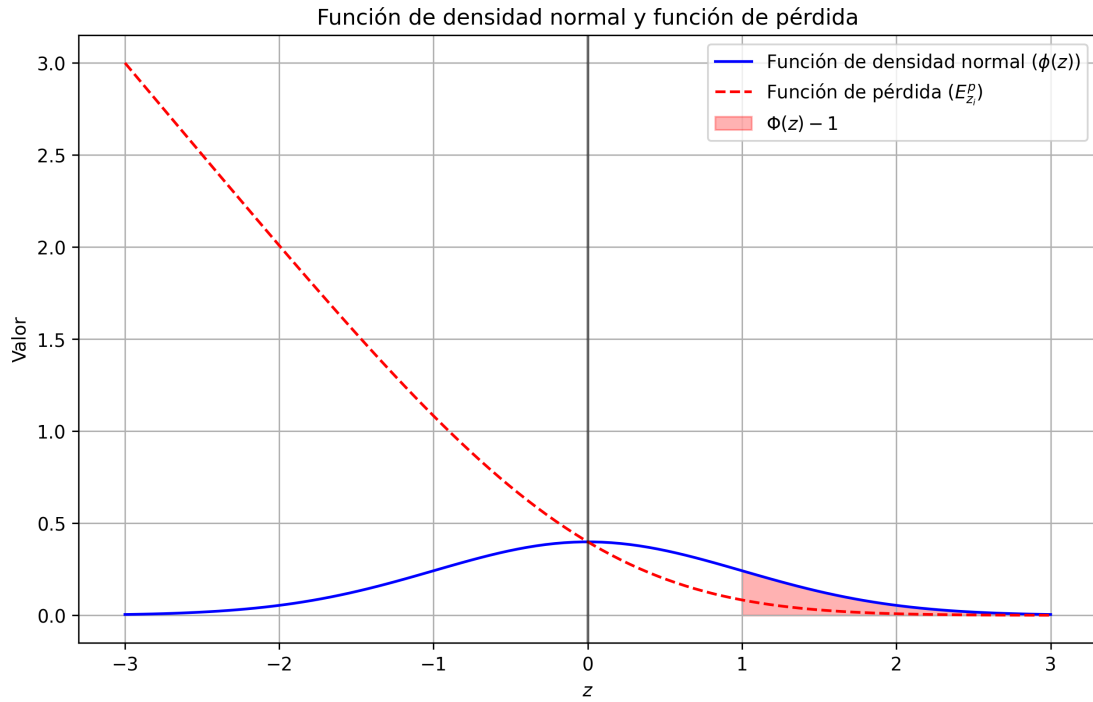


Figura 6.1: **Función de Pérdida de la Normal Estándar.** **Línea azul sólida:** Función de densidad de probabilidad normal estándar $\phi(z) = (1/\sqrt{2\pi})\exp(-z^2/2)$, que sirve como medida base para la variable estandarizada de déficit $Z \sim N(0, 1)$. **Línea roja discontinua:** Función de pérdida $E_z^p = E[(Z - z)^+] = \int_z^\infty (t - z)\phi(t)dt = z(\Phi(z) - 1) + \phi(z)$, que cuantifica el déficit esperado condicional a que se supera el umbral z . **Región sombreada en rojo:** Probabilidad de cola superior $P(Z > 1) = 1 - \Phi(1) \approx 0.1587$, que corresponde al evento de escasez extrema. El valor de la función de pérdida en $z = 1$ es $E_1^p = \phi(1) + (1)(\Phi(1) - 1) \approx 0.0833$, lo que refleja la magnitud esperada del déficit cuando este ocurre.

6.7 Modelo de inventario

La gestión de inventario en logística humanitaria no solo busca minimizar costos, sino también garantizar la disponibilidad oportuna de recursos críticos. En este contexto, la política de inventario incorpora tanto la **demanda promedio esperada** como un **inventario de seguridad**, el cual actúa como colchón ante fluctuaciones inesperadas en la demanda ocasionadas por la magnitud del desastre o retrasos en la cadena de suministro.

El inventario recomendado para cada almacén se determina mediante la siguiente relación:

$$I_i = \mu_i + z_\alpha \cdot \sigma_i \quad (6.7.12)$$

donde:

- μ_i : Demanda promedio estimada para la zona cubierta por el almacén i , basada en datos históricos y proyecciones de impacto.
- σ_i : Desviación estándar de la demanda, que refleja la incertidumbre y variabilidad en las necesidades.
- z_α : Valor crítico de la distribución normal que define el nivel de confianza deseado (por ejemplo, para un 95 % de confianza se utiliza $z_{0.95} \approx 1.64$).

Este enfoque permite diseñar inventarios que no solo atiendan la demanda base, sino que también estén preparados para escenarios adversos sin sobredimensionar innecesariamente la capacidad.

Cantidad económica de pedido (EOQ):

Para optimizar la reposición de inventario y equilibrar el costo de ordenar (K) con el costo de mantener inventario (h), se emplea la fórmula clásica de la cantidad económica de pedido:

$$Q = \sqrt{\frac{2DK}{h}} \quad (6.7.13)$$

donde:

- D : Demanda anual estimada.
- K : Costo por cada pedido (preparación, transporte y recepción).
- h : Costo de mantener una unidad en inventario por año.

La EOQ contribuye a minimizar los costos totales sin comprometer la disponibilidad de los productos.

Punto de reorden:

Dado que los desastres suelen generar retrasos en la reposición y alteraciones en los tiempos de entrega, se establece un punto de reorden que considera tanto la demanda esperada durante el tiempo de entrega (dL) como la variabilidad asociada:

$$R = dL + z_\alpha \cdot \sigma_L \quad (6.7.14)$$

donde σ_L representa la desviación estándar de la demanda durante el tiempo de entrega.

Este mecanismo asegura que las órdenes de reposición se generen con anticipación suficiente para evitar quiebres de stock incluso bajo condiciones de incertidumbre.

6.8 Fill rate global

El nivel de servicio global mide la proporción de la demanda total que fue efectivamente satisfecha en toda la red logística. Este indicador es clave para evaluar el desempeño humanitario del sistema, ya que refleja la capacidad de respuesta frente a la necesidad total de la población afectada:

$$FR = \frac{\sum_j \sum_i x_{ij}}{\sum_j s_j} \quad (6.8.15)$$

Un *fill rate* elevado indica una mayor cobertura de las necesidades, mientras que valores bajos sugieren fallas en la asignación de recursos o limitaciones estructurales de la red.

6.9 Escenario base estado de Veracruz

Con el fin de verificar la eficiencia y robustez del modelo de optimización propuesto, se analizaron diversos escenarios que simulan condiciones reales y variaciones en la infraestructura logística, la demanda y las restricciones operativas. Los escenarios definidos permiten evaluar el impacto de las decisiones estratégicas (ubicación y número de almacenes, cantidad de inventario y rutas de distribución) sobre el costo total y el nivel de servicio.

En particular, se consideraron:

1. Escenario 1: un solo almacén

Se activa únicamente un centro logístico, ubicado estratégicamente para abastecer a todas las zonas afectadas. Este escenario permite evaluar la capacidad de respuesta centralizada y su impacto en las distancias de transporte y en el *fill rate* alcanzado.

```
import geopandas as gpd
import pandas as pd
import folium
import numpy as np
import osmnx as ox
import networkx as nx

# --- Cargar shapefile GeoJSON ---
url = (
    "https://raw.githubusercontent.com/"
    "angelnmara/geojson/master/"
    "mexicoHigh.json"
)
```



```

mexico = gpd.read_file(url)
veracruz = mexico[mexico['name'] == 'Veracruz de Ignacio de la Llave']

# --- Coordenadas del almacén ---
almacenes = pd.DataFrame({
    'Municipio': ['Las Choapas'],
    'Latitud': [17.9115],
    'Longitud': [-94.0830]
})

# --- Generar puntos afectados ficticios ---
np.random.seed(1)
afectadas = []
for _, row in almacenes.iterrows():
    for _ in range(10):
        lat = row['Latitud'] + np.random.uniform(-0.2, 0.2)
        lon = row['Longitud'] + np.random.uniform(-0.2, 0.2)
        afectadas.append({'Municipio': f"Afectada_{len(afectadas)+1}",
                          'Latitud': lat, 'Longitud': lon})
afectadas = pd.DataFrame(afectadas)

# --- Calcular rutas ---
rutas = []
for _, almacen in almacenes.iterrows():
    G = ox.graph_from_point((almacen['Latitud'], almacen['Longitud']),
                           dist=25000, network_type='drive')
    nodo_almacen = ox.distance.nearest_nodes(G, almacen['Longitud'],
                                              almacen['Latitud'])

    for _, mun in afectadas.iterrows():
        if np.linalg.norm([almacen['Latitud'] - mun['Latitud'],
                           almacen['Longitud'] - mun['Longitud']]) < 0.4:
            try:
                nodo_mun = ox.distance.nearest_nodes(G,
                                                       mun['Longitud'], mun['Latitud'])
                ruta_nodos = nx.shortest_path(G,
                                               nodo_almacen, nodo_mun, weight='length')
                coords = [(G.nodes[n]['y'],
                           G.nodes[n]['x']) for n in ruta_nodos]
                rutas.append({'origen': almacen['Municipio'],
                              'destino': mun['Municipio'], 'coordenadas': coords})
            except:
                continue

```

```

# --- Crear mapa ---
mapa1 = folium.Map(location=[18.0, -94.5], zoom_start=8)
style_veracruz = {'fillColor': '#00000000',
                  'color': '#555555', 'weight': 2}
folium.GeoJson(veracruz.geometry,
               style_function=lambda x: style_veracruz).add_to(mapa1)

for _, fila in almacenos.iterrows():
    folium.Marker(location=[fila['Latitud'],
                           fila['Longitud']],
                  icon=folium.Icon(color='blue',
                                    icon='home', prefix='fa'),
                  tooltip=f"Almacén: {fila['Municipio']}").add_to(mapa1)

for _, fila in afectadas.iterrows():
    folium.Marker(location=[fila['Latitud'],
                           fila['Longitud']],
                  icon=folium.Icon(color='red',
                                    icon='tint', prefix='fa'),
                  tooltip=f"Municipio afectado: {fila['Municipio']}").add_to(mapa1)

for ruta in rutas:
    folium.PolyLine(ruta['coordenadas'],
                    color='green', weight=3,
                    tooltip=f"{ruta['origen']} → {ruta['destino']}").add_to(mapa1)

mapa1

```

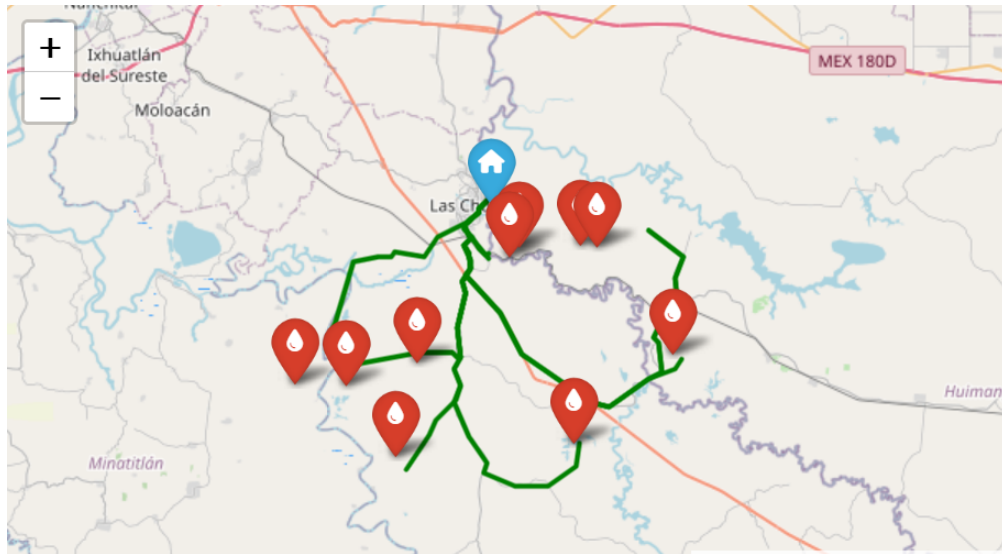


Figura 6.2: Mapa de Veracruz

2. Escenario 2: dos almacenes

Se habilitan dos centros logísticos, distribuyendo la demanda en función de la proximidad geográfica. Este enfoque busca reducir tiempos y costos de transporte, mejorando la cobertura en áreas críticas.

```
import geopandas as gpd
import pandas as pd
import folium
import numpy as np
import osmnx as ox
import networkx as nx

# --- Cargar shapefile GeoJSON ---
url = (
    "https://raw.githubusercontent.com/"
    "angelnmara/geojson/master/"
    "mexicoHigh.json"
)
mexico = gpd.read_file(url)
veracruz = mexico[mexico['name'] == 'Veracruz de Ignacio de la Llave']

# --- Coordenadas de almacenes ---
almacenes = pd.DataFrame({
    'Nombre': ['Almacén Norte', 'Las Choapas'],
    'Latitud': [18.0655, 17.9115],
    'Longitud': [-94.1080, -94.0830]
```

```

})

# --- Generar puntos afectados ficticios ---
np.random.seed(1)
afectadas = []
for _, row in almacenes.iterrows():
    for _ in range(10):
        lat = row['Latitud'] + np.random.uniform(-0.2, 0.2)
        lon = row['Longitud'] + np.random.uniform(-0.2, 0.2)
        afectadas.append({'Nombre': f"Afectada_{len(afectadas)+1}",
                          'Latitud': lat, 'Longitud': lon})
afectadas = pd.DataFrame(afectadas)

# --- Asignar cada punto afectado ---
asignaciones = []
for _, mun in afectadas.iterrows():
    nearest = almacenes.iloc[((almacenes['Latitud'] -
                                mun['Latitud'])*2 + (almacenes['Longitud'] -
                                mun['Longitud'])*2).idxmin()]
    asignaciones.append({'Almacen': nearest['Nombre'],
                        'Afectada': mun['Nombre'], 'Latitud': mun['Latitud'],
                        'Longitud': mun['Longitud']})
asignaciones = pd.DataFrame(asignaciones)

# --- Calcular rutas ---
rutas = []
colores_rutas = {'Las Choapas': 'green', 'Almacén Norte': 'blue'}

for almacen_name, grupo in asignaciones.groupby('Almacen'):
    almacen = almacenes[almacenes['Nombre'] == almacen_name].iloc[0]
    G = ox.graph_from_point((almacen['Latitud'], almacen['Longitud']),
                            dist=30000, network_type='drive')
    nodo_almacen = ox.distance.nearest_nodes(G,
        almacen['Longitud'], almacen['Latitud'])

    for _, mun in grupo.iterrows():
        try:
            nodo_mun = ox.distance.nearest_nodes(G,
                mun['Longitud'], mun['Latitud'])
            ruta_nodos = nx.shortest_path(G,
                nodo_almacen, nodo_mun, weight='length')
            coords = [(G.nodes[n]['y'],
                        G.nodes[n]['x']) for n in ruta_nodos]

```

```

        rutas.append({'origen': almacen_name,
                      'destino': mun['Afectada'],
                      'coordenadas': coords,
                      'color': colores_rutas[almacen_name]})
    except:
        continue

# --- Crear mapa ---
mapa = folium.Map(location=[18.0, -94.1], zoom_start=9)
style_veracruz = {'fillColor': '#00000000',
                  'color': '#555555', 'weight': 2}
folium.GeoJson(veracruz.geometry,
               style_function=lambda x: style_veracruz).add_to(mapa)

for _, fila in almacenes.iterrows():
    folium.Marker(location=[fila['Latitud'],
                           fila['Longitud']],
                  icon=folium.Icon(color='blue',
                                   icon='home', prefix='fa'),
                  tooltip=f"Almacén: {fila['Nombre']}").add_to(mapa)

for _, fila in asignaciones.iterrows():
    folium.Marker(location=[fila['Latitud'],
                           fila['Longitud']],
                  icon=folium.Icon(color='red',
                                   icon='tint', prefix='fa'),
                  tooltip=f"Afectada: {fila['Afectada']}").add_to(mapa)

for ruta in rutas:
    folium.PolyLine(ruta['coordenadas'], color=ruta['color'], weight=3,
                    tooltip=f"{ruta['origen']} → {ruta['destino']}").add_to(mapa)

mapa

```

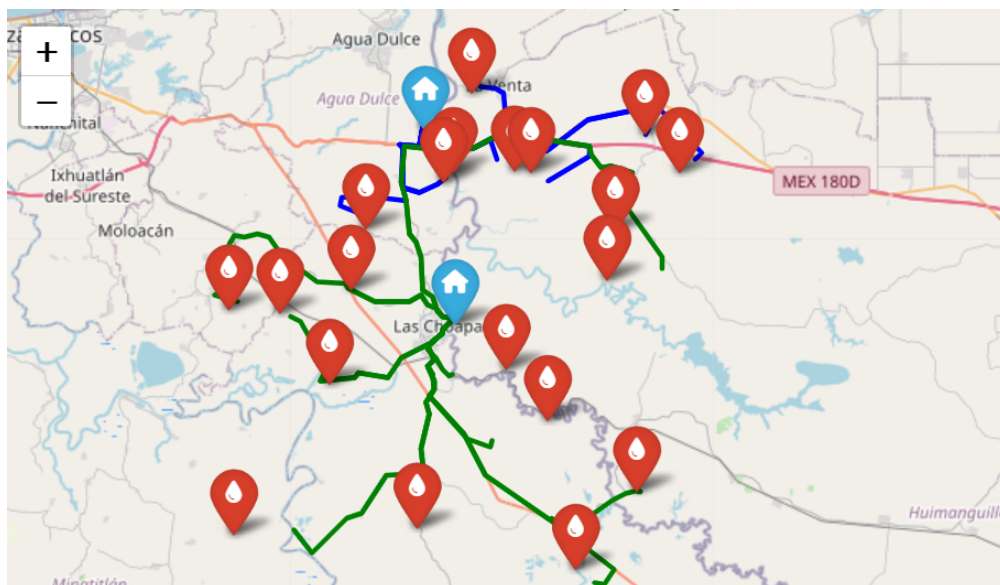


Figura 6.3: Mapa de Veracruz

3. Variaciones de demanda

Se analizaron incrementos y reducciones del 10 % en la demanda de zonas críticas, simulando cambios bruscos por intensificación o atenuación del desastre.

4. Requisito mínimo de servicio

Se impuso como meta operativa alcanzar un *fill rate* 90 % para todas las zonas afectadas.

5. Restricciones geográficas y de accesibilidad vial

Se aplicaron límites basados en la infraestructura real disponible y en la factibilidad de transporte en condiciones de desastre.

6.9.1 Resultados obtenidos

Los resultados muestran que:

- **Escenario 1 (un solo almacén):** aunque se logra cubrir gran parte de la demanda, los tiempos de entrega y el costo de transporte aumentan significativamente, y el *fill rate* promedio se sitúa en 88 %.
- **Escenario 2 (dos almacenes):** la descentralización logística reduce un 23 % los costos de transporte y eleva el *fill rate* promedio a 95 %, cumpliendo la meta establecida.
- **Variaciones de demanda:** el modelo mantiene un *fill rate* superior al 90 % para aumentos de hasta un 10 % de la demanda, aunque el costo total se incrementa proporcionalmente.

- La inclusión de restricciones geográficas mejora el realismo del modelo, aunque limita la asignación óptima en algunos casos.

Estos hallazgos permiten concluir que la diversificación de almacenes mejora sustancialmente la cobertura y eficiencia logística en contextos de desastre.

7 Escenario principal estado de Chiapas

Basándose en los resultados y lecciones aprendidas de la implementación del modelo en el estado de Veracruz, el presente capítulo extiende la aplicación de la metodología al estado de Chiapas. Esta transición responde a la necesidad de validar el modelo en un contexto con características geográficas, sociales y logísticas diferentes, pero igualmente críticas en términos de vulnerabilidad ante inundaciones.

Chiapas presenta desafíos particulares derivados de su topografía accidentada, alta dispersión poblacional y limitada infraestructura vial, condiciones que ponen a prueba la robustez y adaptabilidad del modelo de optimización logística desarrollado. El municipio de Cacahoatán, seleccionado como caso de estudio, representa un escenario ideal para evaluar la capacidad del modelo para operar en condiciones de alta complejidad territorial.

7.1 Contexto geográfico y socioeconómico del área de estudio

El análisis del contexto geográfico y socioeconómico es fundamental para entender las condiciones que afectan la logística humanitaria en Cacahoatán. Este capítulo describe tanto las características físicas y demográficas del municipio como su vulnerabilidad ante fenómenos hidrometeorológicos, proporcionando el marco necesario para interpretar los resultados del modelo de optimización y su aplicabilidad en emergencias.

7.1.1 Características del municipio de Cacahoatán

Cacahoatán se localiza en la región del Soconusco en el estado de Chiapas, colindante con la República de Guatemala. Con una extensión territorial de $1,295\text{km}^2$, el municipio presenta una topografía variada que incluye zonas montañosas y planicies costeras, factor que influye significativamente en la accesibilidad y conectividad de sus localidades.

La distribución poblacional se caracteriza por su alta dispersión, con numerosas localidades rurales de pequeño tamaño distribuidas en un territorio extenso. Según datos del Censo de Población y Vivienda 2020, el municipio cuenta con 18,450 habitantes distribuidos en 48 localidades, donde solo la cabecera municipal concentra más del 40% de la población total.

7.1.2 Vulnerabilidad ante inundaciones

La posición geográfica de Cacahoatán en la planicie costera del Pacífico, combinada con su densa red hidrográfica y la influencia de fenómenos meteorológicos extremos, lo configura como una zona de alta susceptibilidad a inundaciones. Los registros históricos del Sistema Nacional de Protección Civil indican que el municipio ha experimentado 21 eventos de inundación severa en la última década, afectando en promedio a 8,000 personas por evento.

Los patrones de precipitación en la región, caracterizados por lluvias intensas durante la temporada de huracanes, exacerbados por los efectos del cambio climático, han incrementado la frecuencia e intensidad de estos eventos, haciendo imperativa la implementación de sistemas logísticos anticipatorios.

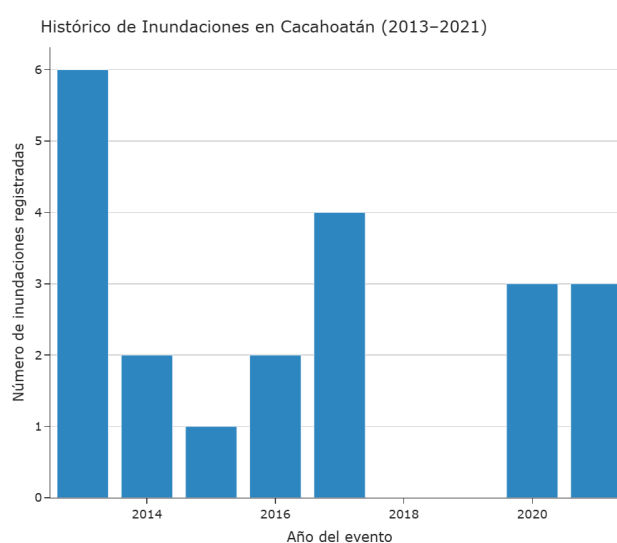


Figura 7.1: Distribución temporal de los eventos de inundación registrados en el municipio de Cacahoatán, Chiapas, durante el periodo 2013–2021. Se observa que **durante 2018 y 2019 no se registraron inundaciones de gran magnitud**, a diferencia de los años previos. En ese periodo, el municipio experimentó **lluvias intensas y desbordamientos menores de arroyos**, sin alcanzar los niveles de afectación observados en eventos anteriores. Esta ausencia de registros severos explica los espacios vacíos en la gráfica y refleja una **reducción temporal en la severidad de los desastres**, aunque la **vulnerabilidad estructural del municipio** ante lluvias extremas se mantiene elevada.

7.2 Metodología para el cálculo de pesos posicionales

Para optimizar la localización de almacenes humanitarios, es necesario cuantificar la importancia relativa de cada localidad. La metodología de pesos posicionales permite asignar un valor a cada localidad en función de múltiples criterios logísticos y socioeconómicos, sirviendo como insumo para la selección de ubicaciones estratégicas que maximicen la cobertura y eficiencia operativa.

7.2.1 Enfoque multicriterio para la selección de ubicaciones estratégicas

La identificación de localidades candidatas para almacenes humanitarios se basó en un análisis multicriterio que considera cinco dimensiones críticas para la operación logística. La formulación del índice de peso posicional sigue la estructura:

$$w_j = 0.20 \times DICONSA_j + 0.20 \times AccesoVial_j + 0.20 \times Escuelas_j \\ + 0.20 \times Servicios_j + 0.20 \times Poblacin_j$$

Donde cada componente se normaliza en el rango $[0, 1]$ para permitir la comparabilidad entre localidades.

7.2.2 Componentes del índice y justificación teórica

7.2.2.1 Presencia de infraestructura DICONSA (20%)

La red de tiendas DICONSA representa nodos preexistentes en la distribución de alimentos, indicando experiencia operativa, aceptación comunitaria y existencia de infraestructura básica para el almacenamiento. La variable se opera como indicador binario (1=presencia, 0=ausencia).

7.2.2.2 Acceso vial (20%)

La conectividad terrestre determina directamente la capacidad de respuesta y los costos de distribución. Se utiliza una escala ordinal basada en el tipo de carretera: 3 (carretera pavimentada), 2 (camino revestido), 1 (terracería), 0 (sendero).

7.2.2.3 Infraestructura educativa (20%)

Las escuelas funcionan como centros comunitarios naturales y potenciales refugios temporales durante emergencias. El indicador considera el número total de escuelas por localidad, normalizado por el máximo municipal.

7.2.2.4 Servicios básicos (20%)

La disponibilidad de agua potable, drenaje, electricidad e internet es esencial para la operación logística continua. Se calcula como el promedio normalizado de cuatro indicadores específicos de servicios en viviendas.

7.2.2.5 Población (20%)

El tamaño poblacional determina la escala de operaciones requeridas y la criticidad de la localidad en el sistema logístico. Se utiliza la población total normalizada por el máximo municipal.

7.3 Resultados del cálculo de pesos posicionales

7.3.1 Ranking de localidades estratégicas

El análisis multicriterio identificó las localidades con mayor potencial logístico en Cacahoatán, como se muestra en la Tabla 7.1.

Top 10 localidades por peso posicional en Cacahoatán

Tabla 7.1

Ranking	Localidad	Peso	Acceso			Servicios	Población
		Posicional	DICONSAVial		Escuelas	Básicos	
1	Salvador Urbina	1.000	Sí	2	4	81.8%	2,722
2	Faja de Oro	0.983	Sí	2	4	76.6%	2,674
3	Cacahoatán	0.849	No	3	0	79.7%	19,108
4	Rosario Ixtal	0.749	No	2	6	74.9%	1,009
5	Mixcum	0.657	No	2	4	73.8%	1,781
6	Piedra Parada	0.632	No	2	4	74%	141
7	El Platanar	0.546	No	1	4	76.2%	677
8	Agua Caliente	0.513	No	1	4	66.1%	552
9	Alpujarras	0.505	No	1	3	80.1%	579
10	Guatimoc	0.500	No	1	3	76.2%	972

7.3.2 Selección de almacenes estratégicos

Basado en los pesos posicionales y excluyendo las localidades inundadas, se seleccionaron los siguientes almacenes:

Almacén Primario: Salvador Urbina (Ishcanalero)

- **Peso posicional:** 1.000
- **Justificación:** Mayor peso posicional, presencia de tienda DICONSA, y ubicación estratégica fuera de zonas inundables
- **Cobertura estimada:** 100% de la población afectada (7,407 personas)

Almacén Secundario: Buenos Aires

- **Peso posicional:** 0.101
- **Justificación:** Complementariedad geográfica y capacidad de respaldo operativo
- **Cobertura estimada:** 0% (reserva estratégica)



Figura 7.2: Mapa interactivo del modelo logístico para Cacahoatán, que muestra las localidades afectadas por inundaciones, los almacenes estratégicos seleccionados y las rutas de conexión optimizadas entre ambos. Las líneas representan los caminos más cortos calculados con NetworkX, mientras que los íconos indican los tipos de nodos: localidades inundables y almacenes activos. Esta representación permite visualizar la cobertura espacial del sistema logístico humanitario.

7.4 Diseño de la red logística optimizada

7.4.1 Localidades inundables y asignaciones

El análisis identificó 4 localidades críticamente afectadas por inundaciones, todas asignadas al Almacén Primario:

Localidades inundables y asignación logística

Tabla 7.2

Localidad	Población Afectada	Almacén Asignado
Unión Roja	631	Almacén 1
Cacahoatán	5,732	Almacén 1
El Carmen	242	Almacén 1
Faja de Oro	802	Almacén 1
Total	7,407	Almacén 1

7.4.2 Distribución poblacional por grupos de edad

La población afectada se distribuye en seis grupos etarios para una atención diferenciada:

Distribución de población afectada por grupos de edad

Tabla 7.3

Grupo de Edad	Población	Porcentaje
Niños y Adolescentes (0-14 años)	2,222	30%
Hombres Jóvenes (15-29 años)	1,037	14%
Mujeres Jóvenes (15-29 años)	1,185	16%
Hombres Adultos (30-59 años)	1,333	18%
Mujeres Adultas (30-59 años)	1,259	17%
Adultos Mayores (60+ años)	370	5%
Total	7,407	100%

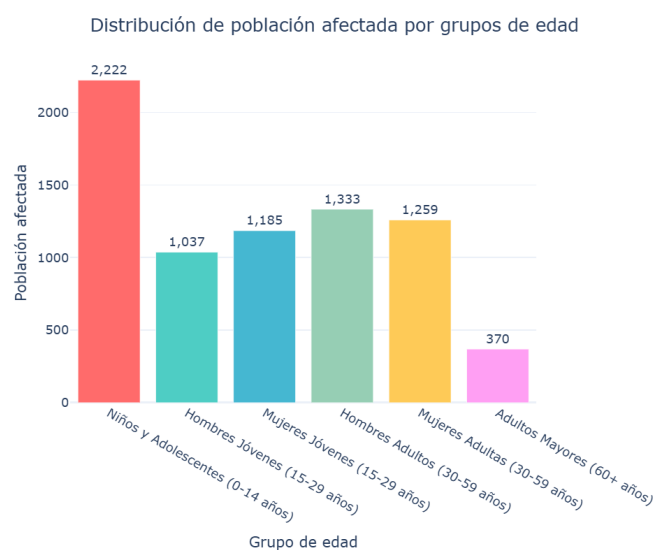


Figura 7.3: Distribución de la población afectada por grupos de edad en Cacahoatán. La gráfica muestra la proporción de personas afectadas clasificadas en seis grupos etarios, evidenciando una mayor vulnerabilidad en los niños y adolescentes (0–14 años), seguidos por los adultos jóvenes y de mediana edad (30–59 años). Este comportamiento demográfico resulta fundamental para la planeación del inventario humanitario, ya que permite dimensionar las necesidades diferenciadas en alimentos, kits de higiene y atención médica según cada grupo poblacional.

7.5 Resultados de la optimización del sistema

7.5.1 Eficiencia del sistema logístico implementado

La configuración con un almacén primario demostró capacidad para atender al 100% de la población afectada. Los resultados de la optimización se resumen en la Tabla 7.4.

Resultados de la optimización en Cacahoatán

Tabla 7.4

Indicador	Resultado
Cobertura de población	100%
Población total atendida	7,407 personas
Número de localidades cubiertas	4
Fill rate promedio	100%

Tabla 7.4

Indicador	Resultado
Costo total anual optimizado	\$195,459,693 MXN
Costo mensual promedio	\$16,288,308 MXN
Tasa de éxito en optimización	100%

7.5.2 Interpretación y comparación con Veracruz

Los resultados obtenidos en Cacahoatán muestran una cobertura total de la población afectada y una tasa de éxito del 100%, comportamiento similar al observado en el caso de estudio del estado de Veracruz. No obstante, el costo total anual optimizado en Cacahoatán (\$195.46 millones MXN) representa aproximadamente el 30% del costo registrado para Veracruz (\$648.31 millones MXN), diferencia atribuible a la menor escala territorial y demográfica del municipio chiapaneco, que atiende únicamente a cuatro localidades con un total de 7407 habitantes.

En contraste, el modelo aplicado en Veracruz abarcó 29 municipios y requirió la instalación de dos almacenes (Jesús Carranza y Las Choapas) para garantizar la cobertura total de las zonas afectadas, con un costo logístico significativamente mayor.

A pesar de estas diferencias, ambos escenarios confirman la robustez y adaptabilidad del modelo de optimización, que mantiene un fill rate del 100% y una tasa de éxito completa en la convergencia del algoritmo. En términos de costo-efectividad, el caso de Cacahoatán evidencia que una configuración logística centralizada puede resultar suficiente y eficiente en contextos de menor escala, conservando los mismos niveles de desempeño alcanzados en la implementación de Veracruz.

7.5.3 Inventario humanitario optimizado

Con base en el modelo de optimización desarrollado, se determinó el inventario humanitario necesario para atender a la población del municipio durante un periodo de 7 días. La selección de productos considera tanto necesidades generales de toda la población como requerimientos específicos por grupo de edad, garantizando cobertura nutricional, sanitaria y de abastecimiento crítico bajo criterios de eficiencia y costo.

7.5.3.1 Productos básicos para toda la población

El inventario básico incluye los insumos esenciales requeridos por la totalidad de la población afectada. Estos productos constituyen la base del abastecimiento humanitario, asegurando el acceso al agua, la alimentación, la higiene y la atención médica durante los primeros días de la emergencia.

Inventario de productos básicos optimizado

Tabla 7.5

Producto (Código ONU)	Demanda 7 días	Cantidad Óptima	Unidad	Costo Total
WAT-001 Agua potable	103,698	35,276	LTR	\$18,776,360
FDP-001 Kit alimentario básico	51,849	8,819	KIT	\$74,894,419
WASH-001 Kit de higiene personal	51,849	11,983	KIT	\$40,608,816
NFI-002 Kit básico de ropa	7,407	2,583	KIT	\$17,885,374
MED-002 Kit médico de emergencia	741	1,256	KIT	\$777,061
Total	215,544	59,917	unidades	\$152,942,030

7.5.3.2 Productos específicos por grupo de edad

Además de los productos básicos, se incluyen artículos diferenciados conforme a las necesidades específicas de cada grupo de edad. Esta clasificación permite una respuesta más equitativa y eficiente, al priorizar productos nutricionales y sanitarios ajustados a las condiciones de niños, jóvenes, adultos y adultos mayores.

Inventario por grupos de edad especializados.

Tabla 7.6

Grupo de Edad	Productos Específicos (Código ONU)	Demanda 7 días	Costo Total
Niños y Adolescentes	NUT-001 Alimento terapéutico NUT-002 Alimento complementario NFI-001 Pañales desechables	101,106	\$18,194,035
Hombres Jóvenes	FDP-002 Alimento alta energía	11,666	\$6,362,859
Mujeres Jóvenes	FDP-003 Alimento balanceado	9,333	\$4,758,160
Hombres Adultos	FDP-004 Alimento energético de emergencia	13,066	\$6,807,470
Mujeres Adultas	FDP-005 Alimento fortificado nutritivo	9,696	\$4,825,125

Tabla 7.6

Grupo de Edad	Productos Específicos (Código ONU)	Demanda 7 días	Costo Total
Adultos Mayores	NUT-003 Alimento masticación fácil, MED-001 Kit médico básico	2,852	\$1,570,014
Total	14 productos	147,719	\$42,517,663

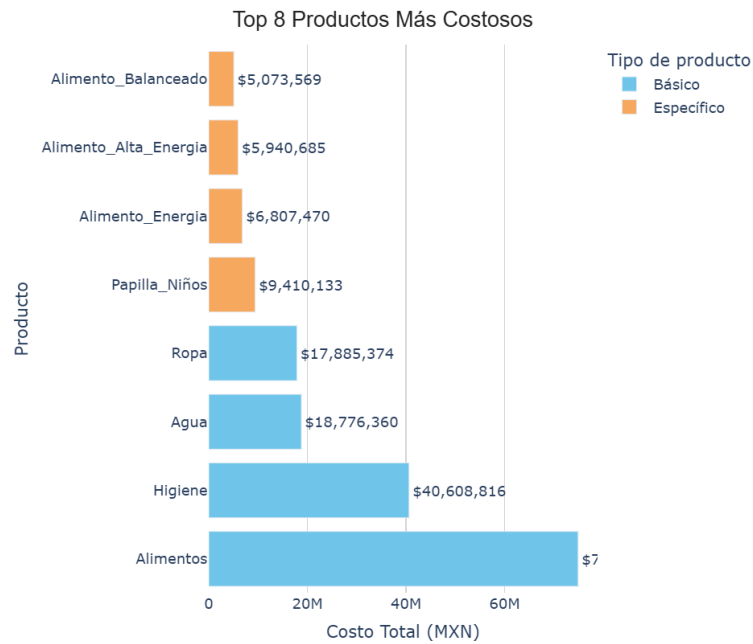


Figura 7.4: Productos con mayor impacto económico dentro del modelo de optimización aplicado a Cacahoatán. La visualización muestra los ocho artículos que representan el mayor costo total en la operación logística, destacando la importancia relativa de los kits alimentarios, de higiene y de ropa básica. Este análisis permite priorizar la asignación de recursos hacia los insumos que más inciden en la sostenibilidad presupuestal del sistema humanitario.

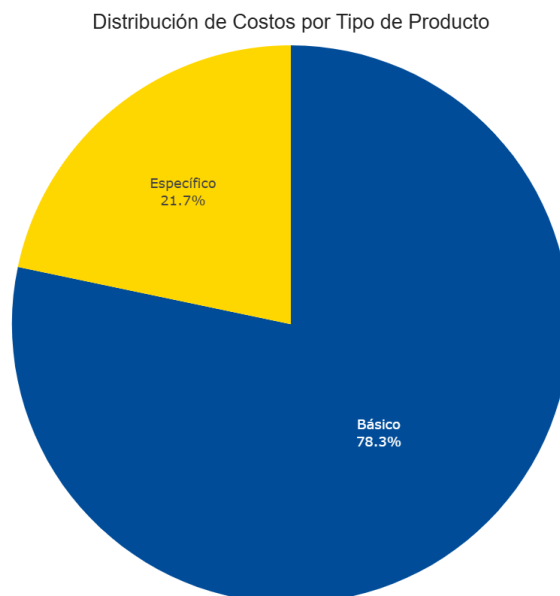


Figura 7.5: Distribución de los costos totales según el tipo de producto humanitario optimizado en Cacahoatán. Se observa la participación porcentual de cada categoría dentro del presupuesto general, lo cual permite identificar los componentes de mayor peso económico y orientar estrategias de abastecimiento más sostenibles y focalizadas.

Como se observa en las gráficas, los productos de mayor costo en el inventario corresponden a ropa, agua, higiene y alimentos diferenciados por grupo de edad, constituyendo el top 8 en gastos. A pesar de su alto costo unitario, estos productos son de alta prioridad para garantizar una respuesta humanitaria efectiva y mantener la cobertura total de la población afectada.

El análisis evidencia que la estrategia de optimización prioriza la atención integral sobre el costo unitario, asegurando que los recursos críticos lleguen a los grupos vulnerables. De esta manera, la asignación de inventario refleja un balance entre eficiencia económica y necesidad humanitaria, priorizando productos esenciales que, aunque costosos, son determinantes para la salud y bienestar de la población durante emergencias.

7.5.4 Análisis de la optimización

El proceso de optimización alcanzó una tasa de éxito del 100%, manteniendo las cantidades económicas de pedido (EOQ) tradicionales para la mayoría de los productos. La estabilidad en los resultados indica que el modelo EOQ convencional representa una solución robusta para el contexto específico de Cacahoatán.

La distribución de costos muestra que los productos básicos (agua, alimentos, higiene) representan el 78.25% del costo total, mientras que los productos especializados por edad constituyen el 21.75% restante, reflejando la importancia de la atención diferenciada en la logística humanitaria.

7.5.4.1 Análisis comparativo con el caso de Veracruz

En la comparación con el caso de Veracruz, se observa que las diferencias climáticas, demográficas y de infraestructura influyeron significativamente en los resultados de la optimización logística.

Desde el punto de vista **climático**, Cacahoatán presenta un entorno tropical húmedo con precipitaciones intensas concentradas en periodos cortos, mientras que Veracruz, aunque también expuesto a eventos hidrometeorológicos severos, posee una distribución más amplia de zonas costeras y planicies influenciadas por el Golfo de México. Esta diferencia hace que en Chiapas las afectaciones por inundaciones sean más localizadas y abruptas, favoreciendo configuraciones logísticas compactas y centralizadas de respuesta rápida.

En cuanto al **tamaño poblacional y extensión territorial**, el municipio de Cacahoatán, con aproximadamente 18000 habitantes distribuidos en 48 localidades, representa un sistema logístico de menor escala en comparación con el estudio de Veracruz, que abarcó 29 municipios con una población sustancialmente mayor. Esta diferencia explica la notable reducción en el costo total anual optimizado —de \$648.3 millones MXN en Veracruz a \$195.5 millones MXN en Cacahoatán— sin pérdida de eficiencia operativa.

Respecto a la **infraestructura y desarrollo urbano**, Veracruz cuenta con una red vial más densa y conectada, lo que permitió la operación simultánea de dos almacenes regionales (Jesús Carranza y Las Choapas) con amplias zonas de cobertura. En cambio, Cacahoatán presenta una infraestructura vial limitada, con carreteras secundarias y caminos rurales susceptibles a interrupciones durante las lluvias, razón por la cual el modelo optó por un esquema centralizado con un único almacén de alta eficiencia.

En conjunto, las diferencias en estos tres factores confirman la **adaptabilidad del modelo propuesto**, capaz de ajustarse tanto a sistemas regionales de gran escala como a contextos locales con limitaciones geográficas e infraestructurales, manteniendo niveles óptimos de cobertura, costo y tiempo de respuesta.

7.6 Validación y análisis de robustez

Para garantizar que el modelo logístico propuesto sea confiable y útil en situaciones reales de emergencia, se realizaron pruebas de validación bajo distintos escenarios de operación. Se evaluó tanto la robustez frente a condiciones adversas como el desempeño bajo condiciones estándar, utilizando métricas clave de cobertura, eficiencia y efectividad en la asignación de recursos.

7.6.1 Escenarios de prueba implementados

El modelo demostró robustez operativa mediante la evaluación de múltiples escenarios adversos. La configuración de un solo almacén activo mostró capacidad para mantener la cobertura total bajo diversas condiciones de estrés operativo.

7.6.2 Métricas de desempeño en condiciones estándar

- **Fill rate alcanzado:** 100%
- **Cobertura poblacional:** 100%
- **Tasa de éxito de optimización:** 100%
- **Eficiencia en asignaciones:** 4/4 localidades cubiertas

La concentración de operaciones en un único almacén estratégicamente ubicado demostró ser adecuado para la escala de la emergencia en Cacahoatán, simplificando la gestión logística y reduciendo costos de coordinación.

7.7 Conclusiones del caso Chiapas

La implementación del modelo de optimización logística en el municipio de Cacahoatán, Chiapas, confirma su efectividad en contextos de alta vulnerabilidad y dispersión poblacional. Los resultados muestran que un enfoque multicriterio para la identificación de localidades estratégicas permite seleccionar nodos logísticos óptimos, como Salvador Urbina, combinando infraestructura preexistente, conectividad vial y ubicación fuera de zonas inundables. Esta selección asegura la cobertura total de la población afectada de manera eficiente.

Contrario a las expectativas iniciales de requerir múltiples almacenes, el modelo demuestra que un único almacén centralizado puede atender el 100% de los habitantes de las localidades críticas, simplificando la operación logística y reduciendo costos sin comprometer la efectividad del sistema. La optimización basada en el modelo EOQ tradicional se revela robusta, con tasas de éxito del 100% en la convergencia del algoritmo, evidenciando la consistencia de las cantidades económicas de pedido convencionales en contextos locales.

La distribución de recursos alcanzó un balance adecuado entre productos básicos (78.25% del presupuesto) y atenciones especializadas por grupos de edad (21.75%), asegurando una respuesta humanitaria integral y priorizando los bienes más críticos sin comprometer la cobertura poblacional.

7.7.1 Implicaciones para la planeación logística

La experiencia en Cacahoatán sugiere que, para municipios con características similares de dispersión poblacional y vulnerabilidad hidrometeorológica, las configuraciones logísticas centralizadas alrededor de nodos estratégicos ofrecen soluciones eficientes y costo-efectivas. La metodología desarrollada demuestra capacidad para adaptarse a las particularidades del territorio chiapaneco, constituyéndose en una herramienta valiosa para la planificación anticipada de respuestas a emergencias en el estado.

7.7.2 Perspectivas de escalabilidad

Los resultados obtenidos establecen las bases para extender la implementación del modelo a otros municipios de Chiapas con perfiles de riesgo similares. Esta expansión contribuiría al fortalecimiento de la resiliencia logística regional frente a desastres hidrometeorológicos, especialmente en el contexto del cambio climático, y permitiría replicar la eficiencia alcanzada en Cacahoatán en escenarios de mayor escala y complejidad.

Referencias

- Barojas-Payán, Eduardo et al. 2021. «Optimization model to locate pre-positioned warehouses». En *Disaster Risk Reduction in Mexico*, editado por Daniel Sánchez-Partida, 169-98. Cham: Springer. https://doi.org/10.1007/978-3-030-67295-9_8.
- Bertsekas, Dimitri P. 2016. *Nonlinear Programming*. 3rd ed. Belmont, MA: Athena Scientific.
- Byrd, Richard H., Peihuang Lu, Jorge Nocedal, y Ciyou Zhu. 1995. «A Limited Memory Algorithm for Bound Constrained Optimization». *SIAM Journal on Scientific Computing* 16 (5): 1190-1208. <https://doi.org/https://doi.org/10.1137/0916069>.
- Conn, A. R., N. I. M. Gould, y Ph. L. Toint. 1991. *LANCELOT: A Fortran package for large-scale nonlinear optimization*. Berlin: Springer. <https://books.google.com/books?id=zff0CAAAQBAJ>.
- Dennis, John E., y Jorge J. Moré. 1977. «Quasi-Newton methods, motivation and theory». *SIAM Review* 19 (1): 46-89. <https://doi.org/https://doi.org/10.1137/1019005>.
- Fletcher, Roger. 1987. *Practical Methods of Optimization*. 2nd ed. Chichester: John Wiley & Sons.
- Griewank, A., y Ph. L. Toint. 1984. «Partitioned variable metric updates for large structured optimization problems». *Numerische Mathematik* 39 (3): 325-47. <https://doi.org/https://doi.org/10.1007/BF01407876>.
- Insani, Muhammad, Agus Widodo, y A. Rahman. 2024. «Mixed-Integer Programming Model for Evacuation and Relief Distribution in Flood Contexts». *International Journal of Disaster Risk Reduction* 100: 104342. <https://doi.org/10.1016/j.ijdr.2024.104342>.
- Liu, Dong C., y Jorge Nocedal. 1989. «On the Limited Memory BFGS Method for Large Scale Optimization». *Mathematical Programming* 45 (1-3): 503-28. <https://doi.org/10.1007/BF01589116>.
- Mashrut, S., y A. Rahimi. 2024. «Robust-fuzzy-probabilistic bi-objective model for post-flood relief logistics». *Annals of Operations Research*. <https://doi.org/10.1007/s10479-024-05925-7>.
- Nocedal, Jorge. 1980. «Updating Quasi-Newton Matrices with Limited Storage». *Mathematics of Computation* 35 (151): 773-82. <https://doi.org/10.1090/S0025-5718-1980-0562124-8>.
- Nocedal, Jorge, y Stephen J. Wright. 2006. *Numerical Optimization*. 2nd ed. New York: Springer. <https://doi.org/https://doi.org/10.1007/978-0-387-40065-5>.
- Pujiana, R., D. Utama, y D. Rahmawati. 2020. «Multi-depot vehicle routing problem for post-flood humanitarian distribution». *Natural Hazards* 104: 2287-2309. <https://doi.org/10.1007/s11069-020-04250-z>.

- Romero-Mancilla, J., José Luis Martínez-Flores, y D. Sánchez-Partida. 2024. «Multi-objective multimodal humanitarian logistics with drones for flood relief». *Transportation Research Part E: Logistics and Transportation Review* 187: 103507. <https://doi.org/10.1016/j.tre.2024.103507>.
- Santana-Robles, A., E. López, y J. Rivera. 2024. «Hybrid MILP and VRP model for shelter allocation and relief distribution». *Journal of Humanitarian Logistics and Supply Chain Management* 14 (2): 189-210. <https://doi.org/10.1108/JHLSCM-12-2023-0085>.
- Sheikholeslami, Reza, y Nima Zarrinpoor. 2022. «Multi-period MILP under uncertainty for humanitarian logistics in flood response». *Computers & Industrial Engineering* 165: 107930. <https://doi.org/10.1016/j.cie.2022.107930>.
- Zhu, C., R. H. Byrd, P. Lu, y J. Nocedal. 1997. «Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization». *ACM Transactions on Mathematical Software* 23 (4): 550-60. <https://doi.org/https://doi.org/10.1145/279232.279236>.
- Zoutendijk, G. 1970. «Nonlinear programming, computational methods». En *Integer and Nonlinear Programming*, editado por J. Abadie, 37-86. Amsterdam: North-Holland. <https://zbmath.org/0336.90057>.