9 de febrero de 2025

Índice general

Capítulo 1

Preliminares

La optimización binivel es un problema de optimización donde un subconjunto de variables debe ser la solución óptima de otro problema de optimización, parametrizado por las variables restantes. Este problema tiene dos niveles jerárquicos de decisión: el nivel superior (líder) y el nivel inferior (seguidor). Este tiene dos características principales: primero, el problema del nivel inferior actúa como una restricción para el nivel superior; segundo, la solución del nivel inferior depende de las variables del nivel superior, creando una interdependencia entre ambos niveles. Por ello, el líder debe anticipar la respuesta óptima del seguidor al tomar decisiones. En términos abstractos, la optimización binivel busca minimizar una función objetivo de nivel superior, $F(x, y)$, donde x son las variables de decisión del líder y y son las variables del seguidor.

En este capítulo se abordará los conocimientos necesarios para el desarrollo de esta tesis. La primera seccioón trata sobre Optimización Binivel: la definición de solucion basada en la existencia de una cierta cooperacion entre lideres y seguidores (caso optimista) y la reformulación KKT. Luego se presentan los conceptos fundamentales de solución, regularidad y estacionariedad para Problemas Matemáticos con Restricciones de Equilibrio (MPEC). La tercera sección contiene las principales transformaciones que se aplican a las restricciones de complementariedad para la solucion numérica del modelo. Por último, se presenta la modelación de los problemas binivel en el lenguaje de programación Julia y los métodos seleccionados implementados en este lenguaje para su resolución.

1.1. Optimización Binivel

El problema de optimización binivel en general se define de la siguiente manera

$$\begin{aligned} &\text{mín}_x \quad F(x,y) \\ &s.a \begin{cases} x \in \mathcal{T} \\ y \in S(x) = \arg\text{mín}_y \{f(x,y) \quad s.a \quad y \in \mathcal{H}\} \\ (x,y) \in \mathcal{M}^0 \end{cases} \end{aligned} \tag{1.1}$$

En la mayoría de las aplicaciones, los problemas de optimizacion del nivel inferior y superior son modelos de programación matemática, por lo que, sin pérdida de generalidad, se asume que $\mathcal{T} = \mathbb{R}^n$, $\mathcal{H}$ es $\{y \in \mathbb{R}^m \in v_j(x,y) \leq 0,\ j = 1,\dots,q\}$ y $\mathcal{M}^0 = \{(x,y) \in \mathbb{R}^{n+m},\ g_i(x,y) \leq 0,\ i = 1,\dots,q\}$. El problema es

$$\begin{aligned} &\min_x \ F(x,y) \\ &\text{sujeto a} \\ &g_i(x,y) \leq 0,\ i = 1,\dots,q, \\ &y \in \underset{y}{\text{argmin}} \{f(x,y) \mid v_j(x,y) \leq 0,\ j = 1,\dots,q\} \end{aligned} \tag{1.2}$$

donde $x \in R^n,\ y \in R^m,\ F(x,y) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R},\ g_i(x,y) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R},$, $i = 1,\dots,q,\ f(x,y) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R},\ v_j(x,y) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R},\ j = 1,\dots,s.$ En el enfoque optimista, se asume que el seguidor, que actúa en el nivel inferior, elegirá la solución más favorable para el líder, quien toma decisiones en el nivel superior. Este es considerado más tratable y, en ciertas situaciones favorables, puede simplificarse a un problema convexo. Además, en el contexto de múltiples objetivos, el enfoque optimista permite alcanzar el mejor frente de Pareto posible, ver **DempeyZemkoho2020**.

$$\begin{aligned} \min_x \ \inf_{y\in S(x)} F(x,y) \\ g_i(x,y) \leq 0, \ i=1,\ldots,q, \\ \text{con } S(x) = \operatorname{argmin}_y \{f(x,y) \mid v(x,y) \leq 0\} \end{aligned} \tag{1.3}$$

Figura 1.1. Problema de optimización bajo el enfoque optimista.

Por otro lado, el enfoque pesimista asume que el seguidor seleccionará la opción menos favorable para el líder entre las soluciones óptimas disponibles, el cual es más complejo de resolver y puede incluso no tener solución. A menudo, se requieren reformulaciones para abordar estos problemas, lo que lo convierte en un reto teórico y computacional significativo en situaciones de múltiples objetivos, conduce al peor frente de Pareto posible, ver **Sinha2017ARO**.

$$\begin{aligned} &\min_x \ \sup_{y\in S(x)} F(x,y) \\ &g_i(x,y) \leq 0, \ i=1,\ldots,q, \\ &\text{con } S(x) = \operatorname*{argmin}_y \{f(x,y) \mid v(x,y) \leq 0\} \end{aligned} \tag{1.4}$$

Figura 1.2. Problema de optimización bajo el enfoque pesimista.

Es relevante destacar que la mayoría de la literatura se centra en el enfoque optimista debido a su mayor facilidad de tratamiento. Sin embargo, el otro también tiene su utilidad, especialmente en la modelación de situaciones donde se considera la aversión al riesgo, ver **DempeyZemkoho2020**. En este contexto, los términos ”líder” y ”seguidor” se utilizan para describir los roles en el modelo a optimizar; el líder toma decisiones considerando las posibles reacciones del seguidor, quien a su vez reacciona seleccionando su mejor opción, ver **Sinha2017ARO**.

Dado que en la tesis trataremos sobre problemas binivel de enfoque optimista mostraremos algunos resultados referidos al modelo resultante.

Primeramente es importante notar el siguiente resultado, ver **Scmidtbiblio**.

Proposición 1 *El problema de optimización binivel optimista* (**??**) *se formula equivalentemente como:*

$$\begin{array}{cc} \text{mín}_{x,y} & F(x,y) \\ s.t. & g_i(x,y) \leq 0, i = 1 \dots q, \\ & y \in S(x), \end{array}$$

donde $S(x)$ es el conjunto de soluciones óptimas del problema parametrizado por x

$$\text{mín}_{y \in Y(x)} \quad f(x,y) s.t. \quad v_j(x,y) \leq 0, j = 1 \dots s,$$

De ahí que a partir de ahora se considerará el problema (**??**) y se le llamará indistintamente problema de dos niveles. Los problemas de dos niveles pueden ser reformulados en un problema de un solo nivel al reemplazar el problema del nivel inferior por las condiciones KKT de este en las restricciones del primer nivel.

$$\begin{array}{ll} \underset{x,y,\lambda_j}{\text{mín}} & F(x,y) \\ s.a & \begin{cases} g_i(x,y) \leq 0, i = 1 \dots q, \\ \nabla_y f(x,y) + \sum_{j=1}^{s} \nabla_y v_j(x,y) \lambda_j = 0, \\ v_j(x,y) \leq 0, j = 1 \dots s, \\ v_j(x,y)\lambda_j = 0, j = 1 \dots s, \\ \lambda_j \geq 0, j = 1 \dots s. \end{cases} \end{array} \quad (1.5)$$

Los tres últimos grupos de restricciones expresan que v y λ están restringidas en signo y que al menos una es 0. Estas condiciones son conocidas como **restricciones de complementariedad**. Estos modelos corresponden a la clase de problemas de programación matemática con restricciones de complementariedad (MPEC). A continuación, presentamos los resultados de esta área necesarios para el desarrollo de esta tesis.

1.2. Programación Matemáticos con Restricciones de Equilibrio (MPEC)

Un **problema de Programación Matemático con Restricciones de Equilibrio (MPEC)** es un tipo de problema de optimización no lineal que incluye restricciones de equilibrio, específicamente, restricciones de complementariedad.

$$
\begin{array}{ll}
\text{mín} & f(z) \\
\text{s.t.} & \begin{array}{rcl} g_i(z) & \leq & 0, \quad i = 1, \dots, q, \\ h_k(z) & = & 0, \quad k = 1 \dots, m, \\ G_j(z) & \geq & 0, \quad j = 1, \dots, s, \quad H_j(z) \;\; \geq \;\; 0, \quad j = 1, \dots, s, \end{array} \\
& G_j(z)^T H_j(z) = 0, \; j = 1, \dots, s.
\end{array}
\tag{1.6}
$$

Definición de MPEC

donde $f : \mathbb{R}^{\hat{n}} \to \mathbb{R}$, $g : \mathbb{R}^{\hat{n}} \to \mathbb{R}$, $h : \mathbb{R}^{\hat{n}} \to \mathbb{R}$, $G : \mathbb{R}^{\hat{n}} \to \mathbb{R}$, y $H : \mathbb{R}^{\hat{n}} \to \mathbb{R}$ son funciones continuamente diferenciables. El término de complementariedad se debe a la existencia de las restricciones homónimas

$$
G_j(z) \geq 0, \; H_j(z) \geq 0, \quad G_j(z)^T H_j(z) = 0, \quad j = 1, \dots, s. \tag{1.7}
$$

Debido a las restricciones de complementariedad, los MPEC no cumplen con las condiciones de regularidad estándar, lo que hace que las condiciones de KKT no sean directamente aplicables como condiciones de optimalidad de primer orden. Los MPEC son utilizados para modelar problemas donde existen restricciones de equilibrio, como problemas de ingeniería y economía, ver **Flegel2003AFJ**; **DempeyZemkoho2020**.

A continuación como en **Flegel2003AFJ**, se introduce las siguientes definiciones, útiles para la obtención de condiciones necesarias de optimalidad.

Definición 1 (Conjunto de Índices) *Dado un vector factible z^* del MPEC*

(**??**), *definimos los siguientes conjuntos de índices:*

$$\begin{aligned} J_G := & \quad J_G(z^*) := \{i | G_i(z^*) = 0, H_i(z^*) > 0\} \\ J_{GH} := & \quad J_{GH}(z^*) := \{i | G_i(z^*) = 0, H_i(z^*) = 0\} \\ J_H := & \quad J_H(z^*) := \{i | G_i(z^*) > 0, H_i(z^*) = 0\} \end{aligned} \tag{1.8}$$

Para definir condiciones de regularidad para MPEC, introducimos el siguiente problema, dependiente de z^*:

Definición 2 (Programa No Lineal Relacionado (RNLP)) *Un Programa No Lineal Relacionado $RNLP_{a,b} := RNLP_{a,b}(z^*)$ es:*

$$\begin{array}{ll} \text{mín} & f(z) \\ & \begin{array}{rcllrcll} g_i(z) & \leq & 0, & i = 1, \ldots, q, & & & & \\ h_k(z) & = & 0, & k = 1 \ldots, m, & & & & \\ G_j(z) & = & 0, & j \in J_G \cup J_a, & G_j(z) & \geq & 0, & j \in J_H, \\ H_j(z) & = & 0, & j \in J_H \cup J_b, & H_j(z) & \geq & 0, & j \in J_G, \end{array} \\ \text{s.t.} & \end{array} \tag{1.9}$$

Problema No Lineal Ajustado (TNLP)

donde $a \cup b = J_{GH}$.

El TNLP (**??**) puede usarse para definir variantes de regularidad y condiciones necesarias de optimalidad adecuadas para MPEC como son la restricción estándar de independencia lineal (LICQ en su forma abreviada) y la de punto de KKT .

Definición 3 (MPEC-LICQ) *El MPEC* (**??**) *se dice que satisface la MPEC-LICQ en un vector factible z^* si los correspondientes $RNLP_{a,b}(z^*)$ satisfacen la LICQ en ese vector z^*.*

Cabe destacar que si la LICQ se cumple en $RNLP_{a,b}(z^*)$ para una partición (a, b) se cumple para todas.

En el contexto de los MPEC en **Flegel2003AFJ** se exponen varios tipos de puntos estacionarios que son cruciales para analizar la optimalidad, los cuales son los siguientes:

Definición 4 (Punto Factible) *Un punto factible z^* del MPEC se llama débilmente estacionario si existe un multiplicador de Lagrange $(\mu, \alpha, \beta, \gamma)$ tal que se cumplen las siguientes condiciones:*

$$
\begin{aligned}
&\nabla f(z^*) + \sum_{i=1}^{q} \mu_i \nabla g_i(z^*) + \sum_{k=1}^{q_0} \alpha_k \nabla h_k(z^*) - \sum_{i=1}^{s} [\beta_j \nabla G_j(z^*) + \gamma_j \nabla H_j(z^*)] = \vec{0} \\
&\beta_j,\ j \in J_G \textit{ libre}, \quad \beta_j,\ j \in_{J_G \cup J_{GH}} \textit{ libre}, \quad \beta_j,\ j \in_{J_H} = 0, \\
&\gamma_j,\ j \in_{J_H} \textit{ libre}, \quad \gamma_j,\ j \in_{J_G \cup J_{GH}} \textit{ libre}, \quad \gamma_j,\ j \in_{J_G} = 0, \\
&g_i(z^*) \leq 0, \quad \mu_i \geq 0, \quad \mu_i g_i(z^*) = 0, i = 1, \ldots, q.
\end{aligned}
\tag{1.10}
$$

Este concepto de estacionariedad es equivalente al cumplimiento de la estacionariedad clásica en el problema $RNLP_{J_{GH}, J_{GH}}$. De ahí que es una condición relativamente débil. Existen conceptos más fuertes de estacionariedad que se derivan y estudian en otros lugares. En particular, se tienen las siguientes definiciones:

Definición 5 (Punto C-estacionario) *El punto z^* es: C-estacionario si, para cada $i \in J_{GH}$, $\lambda_i^G \lambda_i^H \geq 0$ se cumple.*

Definición 6 (Punto M-estacionario) *El punto z^* es: M-estacionario si, para cada $i \in J_{GH}$, o bien $\lambda_i^G, \lambda_i^H > 0 \vee \lambda_i^G \lambda_i^H = 0$.*

Definición 7 (Punto Fuertemente estacionario) *El punto z^* es: fuertemente estacionario si, para cada $i \in J_{GH}$, $\lambda_i^G, \lambda_i^H \geq 0$.*

La C y la M estacionariedad son condiciones necesarias de optimalidad bajo ciertas condiciones de regularidad, más débiles que la MPEC-LICQ. La fuerte conlleva al siguiente resultado:

Teorema 1 *Sea $z^* \in \mathbb{R}^n$ un mínimo local del MPEC (??). Si MPEC-LICQ se cumple en z^*, entonces existe un único multiplicador de Lagrange tal que $(z^*, \mu,^*)$ es fuertemente estacionario.*

O sea el punto z^* es un punto estacionario del problema relajado

$$\begin{array}{ll} \text{mín} & f(z) \\ \text{s.t.} & g_i(z) \leq 0,\ i = 1, \ldots, q, \\ & h_k(z) = 0,\ k = 1 \ldots, m, \\ & G_j(z) = 0,\ j \in J_G \quad G_j(z) \geq 0,\ j \in J_{GH},\quad G_j(z) \geq 0,\ j \in J_H, \\ & H_j(z) = 0,\ j \in J_H \quad H_j(z) \geq 0,\ j \in J_{GH},\quad H_j(z) \geq 0,\ j \in J_G, \\ & \text{Problema No Lineal Relajado} \end{array} \tag{1.11}$$

Los algoritmos buscan al menos converger a puntos de este tipo. En la próxima sección se revisarán las reformulaciones que se resuelven las bibliotecas de Julia.

1.3. Métodos de Reformulación para Optimización Binivel

La optimización binivel presenta desafíos particulares debido a su naturaleza jerárquica y las condiciones de complementariedad resultantes. A continuación, se presentan los principales métodos de reformulación implementados en la literatura, basados en la transformación del MPEC (??).

1.3.1. Método Big-M

El método Big-M (Fortuny-Amat y McCarl) es una técnica fundamental para reformular problemas de optimización binivel en problemas MPEC. Este método

aborda específicamente las condiciones de complementariedad que surgen en estas reformulaciones, transformando el problema original en un problema de programación lineal mixta entera (MILP).
La reformulación mediante Big-M introduce un parámetro M suficientemente grande y variables binarias para transformar las condiciones de complementariedad no lineales en restricciones lineales. Para cada condición de complementariedad $v_j(x,y)\lambda_j = 0$ en (**??**), el método introduce una variable binaria $\delta_j \in \{0,1\}$ y cotas superiores M_p, M_d bajo las siguientes restricciones:

$$\begin{aligned} v_j(x,y) &\geq -M_p(1-\delta_j) \\ \lambda_j &\leq M_d\delta_j \\ \delta_j &\in \{0,1\} \end{aligned} \tag{1.12}$$

donde M_p y M_d son valores grandes para las variables primales y duales, respectivamente. La efectividad del método depende crucialmente de la selección apropiada de estos valores, que deben ser suficientemente grandes para no excluir la solución óptima, pero no excesivamente grandes para evitar inestabilidades numéricas **BilevelJump**.

1.3.2. Método SOS1

El método de Conjuntos Ordenados Especiales tipo 1 (SOS1) evita el uso de parámetros Big-M mediante restricciones de tipo conjunto. Para cada par complementario (v_j, λ_j), $j = 1, \ldots, s$:

$$\begin{gathered} v_j(x,y) = s_j \\ [s_j; \lambda_j] \in \text{SOS1} \end{gathered} \tag{1.13}$$

donde $SOS1 = \{(a,b) \in \mathbb{R} : a, b \geq 0,\ ab = 0\}$

Esta restricción fuerza que al menos una variable en el par sea cero, preservando la no linealidad original sin necesidad de cotas. Note que SOS1 es un cono. Esta es una reformulación del MPEC como un problema de optimización cónica. Es particularmente eficaz en MPECs con restricciones del tipo $u \in K$, donde K es un cono, pues sigue siendo un problema de optimización cónica y se pueden aplicar algoritmos específicos para la resolucion de este tipo de modelos, **BilevelJump**. La dificultad principal es que el cono es no convexo. Una aplicación de este enfoque se encuentra en **SadddiquiNaturalGasSOS1**.

1.3.3. Método ProductMode

El método ProductMode representa un enfoque directo para manejar las condiciones de complementariedad en su forma de producto original. Este método es particularmente útil cuando se trabaja con solucionadores de programación no lineal (NLP), aunque no garantiza la optimalidad global.
La implementación del ProductMode mantiene la restricción de complementariedad en su forma original:

$$v_j(x, y) \cdot \lambda_j \leq t \tag{1.14}$$

donde $t > 0$ es un parámetro de regularización pequeño. Esta formulación, aunque no satisface las condiciones de calificación de restricciones estándar, es útil para obtener soluciones iniciales y puede ser especialmente efectiva cuando se combina con solucionadores NLP, ver **BilevelJump**. sin embargo, bajo condiciones naturales, la sucesión que genera converge a puntos C estacionarios y bajo regularidad hasta fuertemente estacionarios, ver **scholtes12**.
La Tabla **??** resume los requisitos y características de cada método:

Método	**Solver Requerido**	**Ventajas**
Big-M	MIP	Estabilidad numérica controlada
SOS1	MIP con SOS1	Sin parámetros ad-hoc
ProductMode	NLP	Manejo de no linealidades

Cuadro 1.1. Comparación de métodos de reformulación

Para casos con restricciones no lineales $v_j(x, y)$, se recomienda combinar ProductMode con técnicas de linealización por tramos **BilevelJump**. Todos estos métodos están implementados en BilevelJuMP.jl, permitiendo experimentación con reformulaciones que se pueden encontrqr en **BilevelJump**.

1.4. Modelación en Julia

Este paquete permite abordar una amplia variedad de tipos de problemas. En este trabajo usaremos las bibliotecas $JuMP$ y BilevelJuMP.

JuMP

Esta biblioteca permite resolver problemas de programación matemática no lineal y discreta. O sea

$$\min F(z)$$

sujeto a:

$$g_i(z) \leq 0, \qquad i = 1, \ldots, q, \tag{1.15}$$

$$h_k(z) = 0, \qquad i = 1, \ldots, q_0, \tag{1.16}$$

$$\tag{1.17}$$

donde $z \in \mathbb{R}^p \times \mathbb{R}^{\hat{n}-p}$. JuMP permite gestionar una variedad de casos diferenciando como si las variables son continuas o discretas y las restricciones cuadráticas, lineales, de cotas o no lineales y, similarmente, si la funci 'on objetivo es cuadrática, lineal o no lineal. También permite definir restricciones de complementariedad y ofrece métodos para resolver MPECs como ******. Pqra más detalles sobre lq biblioteca, ver **JuMPPaper**.

BilevelJuMP

BilevelJuMP.jl es un paquete de Julia diseñado para modelar y resolver problemas de **optimización binivel**, también conocidos como problemas de optimización de dos niveles o jerárquica, para más detalles ver **BilevelJump**.
BilevelJuMP.jl facilita la modelación ya que el problema del seguidor se representa usando la sintaxis de JuMP que, como se mencionó en la parte anterior, permite incluir de forma diferenciada las restricciones lineales y no lineales, las variables continuas y enteras, y diferentes tipos de funciones objetivos. Para el problema del nivel superior gestiona sus restricciones y la funcion objetivo de forma análoga.
Los usuarios pueden experimentar con diversas reformulaciones para las restricciones de complementariedad en los problemas MPEC, incluyendo SOS1 y McCarl (Big-M), entre otros. BilevelJuMP.jl puede utilizar tanto solucionadores de **programación lineal mixta entera (MIP)** como solucionadores de **programación no lineal (NLP)**, dependiendo de las características del problema y la reformulación elegida.
A pesar de sus capacidades, BilevelJuMP.jl presenta algunas limitaciones, entre las cuales se encuentran:

- Enfrentar dificultades en problemas altamente no lineales o con estructuras de optimización complejas que no se puedan representar adecuadamente en la sintaxis de JuMP.

- Existencia de ciertas restricciones que podrían no ser compatibles o que requieren transformaciones adicionales que podrían complicar el modelo.

- El problema es de gran escala afectando el rendimiento del solver, el cual puede ser un factor crítico.
- La formulación y resolución de problemas muy específicos o especializados podrían no estar completamente optimizadas en el paquete.

Teneindo en cuenta la posibilidad de que los métodos de solución converjan a puntos estacionarios de distinto tipo, resulta interesante comprobar si, conocido un punto de esta naturaleza, el algoritmo logra obtener un punto con mejor evaluación de la funcion objetivo. Esta es la motivación para el dasarrollo del generador de problemas prueba, objeto de esta tesis y que se presentar ' en el pr oximo capítulo. beginequation