

Optimización con Restricciones

Algoritmos y Técnicas de Descomposición

Dr. Ing. Claudio Risso (crisso@fing.edu.uy)

Instituto de Computación (FING - UDELAR)

Curso Optimización Continua y Aplicaciones (OCA)
(Octubre 2023)

Resolver Condiciones KKT

Aunque escala pobremente, resolver las ecuaciones de Karush-Kuhn-Tucker es un camino para hallar el óptimo.

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

En este ejemplo, donde todas las funciones son convexas, sabemos que las condiciones son necesarias y suficientes si F es no-trivial, como en este caso, donde $(0,2)$ es interior.

En general, es buena idea verificar que las restricciones tienen algún efecto, para lo cual, se puede calcular el óptimo sin ellas.

Sabemos que para $f(x, y)$, ese óptimo es $\bar{z} = (\frac{26}{9}, \frac{2}{9})$ (ver BFGS), pero podemos descartarlo porque no cumple $g_1(x, y) = \frac{370}{81} \leq 0$.

Resolver Condiciones KKT

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

El cálculo directo pasa por resolver las condiciones de Kuhn-Tucker:

$$\nabla f_{\lambda} = \begin{bmatrix} 4x+2y-12 \\ 2x+10y-8 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2x \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 2y-4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2x \\ -1 \end{bmatrix} = \vec{0},$$
$$x^2 + y - 4 \leq 0, \quad -4x + y^2 - 4y \leq 0, \quad x^2 - y \leq 0, \quad \lambda_1(x^2 + y - 4) = 0,$$
$$\lambda_2(-4x + y^2 - 4y) = 0, \quad \lambda_3(x^2 - y) = 0, \quad \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Los pasos para resolver KKT son: i) elegir un *set de signos* para los λ 's; ii) resolver las igualdades resultantes; y iii) verificar si se cumplen las otras desigualdades restantes.

Resolver Condiciones KKT

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

El cálculo directo pasa por resolver las condiciones de Kuhn-Tucker:

$$\nabla f_{\lambda} = \begin{bmatrix} 4x+2y-12 \\ 2x+10y-8 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2x \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 2y-4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2x \\ -1 \end{bmatrix} = \vec{0},$$

$$x^2 + y - 4 \leq 0, \quad -4x + y^2 - 4y \leq 0, \quad x^2 - y \leq 0, \quad \lambda_1(x^2 + y - 4) = 0, \\ \lambda_2(-4x + y^2 - 4y) = 0, \quad \lambda_3(x^2 - y) = 0, \quad \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Si los λ 's son $(0, 0, 0)$, resolvemos $\nabla f_{\lambda} = \nabla f = \vec{0}$, que es la solución sin restricciones (ya vimos que no es factible).

Resolver Condiciones KKT

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

El cálculo directo pasa por resolver las condiciones de Kuhn-Tucker:

$$\nabla f_{\lambda} = \begin{bmatrix} 4x+2y-12 \\ 2x+10y-8 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2x \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 2y-4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2x \\ -1 \end{bmatrix} = \vec{0},$$
$$x^2 + y - 4 \leq 0, \quad -4x + y^2 - 4y \leq 0, \quad x^2 - y \leq 0, \quad \lambda_1(x^2 + y - 4) = 0,$$
$$\lambda_2(-4x + y^2 - 4y) = 0, \quad \lambda_3(x^2 - y) = 0, \quad \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Probando con $(+, 0, 0)$, hay que anular $\begin{bmatrix} 4x+2y-12+2\lambda_1x \\ 2x+10y-8+\lambda_1 \end{bmatrix}$,

además hay que satisfacer $x^2 + y - 4 = 0$. Usando $y = 4 - x^2$ y despejando de la segunda ecuación en ∇f_{λ} : $\lambda_1 = 10x^2 - 2x - 32$, y $10x^3 - 3x^2 - 30x - 2 = 0$, que tiene una única raíz para la que λ_1 es positivo ($\lambda_1 = 0.9609$).

Resolver Condiciones KKT

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

El cálculo directo pasa por resolver las condiciones de Kuhn-Tucker:

$$\nabla f_{\lambda} = \begin{bmatrix} 4x+2y-12 \\ 2x+10y-8 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2x \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 2y-4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2x \\ -1 \end{bmatrix} = \vec{0},$$
$$x^2 + y - 4 \leq 0, \quad -4x + y^2 - 4y \leq 0, \quad x^2 - y \leq 0, \quad \lambda_1(x^2 + y - 4) = 0,$$
$$\lambda_2(-4x + y^2 - 4y) = 0, \quad \lambda_3(x^2 - y) = 0, \quad \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Probando con $(+, 0, 0)$, hay que anular $\begin{bmatrix} 4x+2y-12+2\lambda_1x \\ 2x+10y-8+\lambda_1 \end{bmatrix}$,

además hay que satisfacer $x^2 + y - 4 = 0$. Sin embargo, ese punto $x = 1.9183$, $y = 0.3203$, no satisface $g_3(x, y) \leq 0$. Hay que seguir probando entonces otras combinaciones.

Resolver Condiciones KKT

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

El cálculo directo pasa por resolver las condiciones de Kuhn-Tucker:

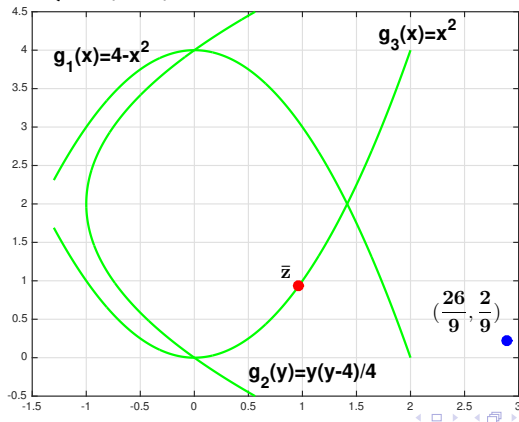
$$\nabla f_{\lambda} = \begin{bmatrix} 4x+2y-12 \\ 2x+10y-8 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2x \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 2y-4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2x \\ -1 \end{bmatrix} = \vec{0},$$
$$x^2 + y - 4 \leq 0, \quad -4x + y^2 - 4y \leq 0, \quad x^2 - y \leq 0, \quad \lambda_1(x^2 + y - 4) = 0,$$
$$\lambda_2(-4x + y^2 - 4y) = 0, \quad \lambda_3(x^2 - y) = 0, \quad \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Seguiríamos probando hasta $(0, 0, +)$, donde $y = x^2$,

$\lambda_3 = 2x + 10x^2 - 8$ y resolvemos $10x^3 + 3x^2 - 6x - 6 = 0$, que tiene $x = 0.9654$ como su única raíz real, de donde $y = 0.9319$ y $\lambda_3 = 3.2499 > 0$. El punto verifica las demás restricciones y es por tanto el óptimo, así que $\bar{z} = (0.9654, 0.9319)$ y $f(\bar{z}) = -1.0345$.

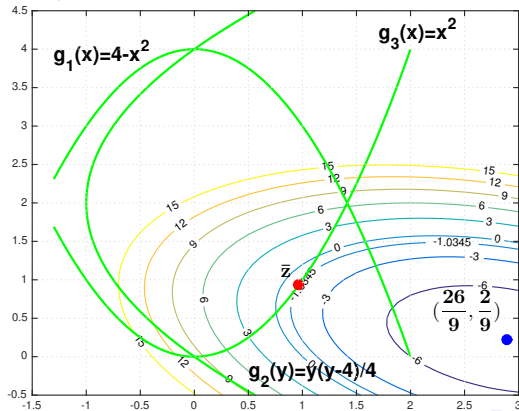
Resolver Condiciones KKT

Ejemplo: (P)
$$\begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$



Resolver Condiciones KKT

Ejemplo: (P)
$$\begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$



Resolver Condiciones KKT

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

El cálculo directo pasa por resolver las condiciones de Kuhn-Tucker:

$$\nabla f_{\lambda} = \begin{bmatrix} 4x+2y-12 \\ 2x+10y-8 \end{bmatrix} + \lambda_1 \begin{bmatrix} 2x \\ 1 \end{bmatrix} + \lambda_2 \begin{bmatrix} -4 \\ 2y-4 \end{bmatrix} + \lambda_3 \begin{bmatrix} 2x \\ -1 \end{bmatrix} = \vec{0},$$

$$x^2 + y - 4 \leq 0, \quad -4x + y^2 - 4y \leq 0, \quad x^2 - y \leq 0, \quad \lambda_1(x^2 + y - 4) = 0, \\ \lambda_2(-4x + y^2 - 4y) = 0, \quad \lambda_3(x^2 - y) = 0, \quad \lambda_1, \lambda_2, \lambda_3 \geq 0.$$

Verificar resultados usando *fmincon* (Matlab) o *sqp* (Octave).

OBS: Aplicar KKT puede requerir 2^m iteraciones.

La dificultad radica en las desigualdades. Sin ellas, el problema se reduce a un sistema de ecuaciones.

Active Set

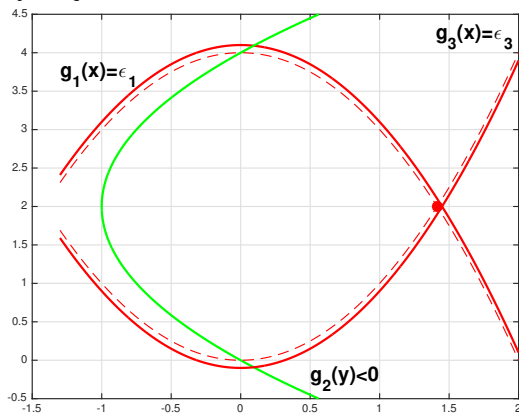
Se basa en la interpretación de los λ 's como *costos marginales* para un problema (P) $\begin{cases} \min f(x) \\ g_i(x) \leq 0 \end{cases}$. Si supiéramos de antemano cuáles son los λ 's positivos, concluiríamos que el punto es óptimo, porque perturbar las restricciones i asociadas tendría el efecto de descender el valor del objetivo, al ser $\frac{\partial \Psi(0)}{\partial \epsilon_i} = -\lambda_i < 0$.

Por ejemplo, resolvemos el problema (P) anterior sin restricciones, para concluir que g_1 y g_3 no se cumplen, y ensayamos el caso donde ellas están activas y g_2 no (i.e. $\lambda_2 = 0$).

Resolvemos con alguna técnica numérica, o en este caso sustituimos $y = x^2$ en g_1 para llegar a $2x^2 = 4$, que tiene raíz $x = \sqrt{2}$, de donde: $y = 2, \lambda_1 = -7, \lambda_3 = 5 + 2\sqrt{2}$.

Active Set

Observar que cambiar $g_i(x) \leq 0$ por $g_i(x) \leq \epsilon_i$ extiende la región factible cuando $\epsilon_i > 0$. Así que expandiendo la región factible F en las restricciones con $\lambda_i > 0$ se mejora la solución. Por el contrario, si $\lambda_i < 0$ hay mejores valores hacia el interior de F .



Active Set

Active Set presupone un conjunto factible de restricciones activas $\mathcal{A} \subseteq \{1, \dots, m\}$, calcula los multiplicadores, y luego remueve del conjunto \mathcal{A} las restricciones con $\lambda_i < 0$, en el entendido que se gana avanzando hacia adentro de la región factible.

Buscamos resolver $(P) \begin{cases} \min f(x) \\ g_i(x) \leq 0 \end{cases}, 1 \leq i \leq m.$

- 1 Conseguir un conjunto $\mathcal{A} \subseteq \{1, \dots, m\}$ factible.
- 2 Resolver el problema que sólo tiene las restricciones en \mathcal{A} activas, agregando otras restricciones si ellas son violadas.
- 3 Calcular los multiplicadores asociados (λ 's) y remover de \mathcal{A} los índices de los mutiplicadores negativos.
- 4 Si no se alcanzó condición de salida, volver al punto 2.

Active Set

En el contexto de este curso podemos suponer que para \mathcal{A} dado, la resolución numérica de KKT se reduce a un problema de minimización sin restricciones.

$$\text{Ejemplo: } (P) \begin{cases} \min f(x, y) = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1(x, y) = x^2 + y - 4 \leq 0, \\ g_2(x, y) = -4x + y^2 - 4y \leq 0, \\ g_3(x, y) = x^2 - y \leq 0. \end{cases}$$

Para $\mathcal{A} = \{1, 3\}$ (ejemplo anterior), se tiene $\lambda_2 = 0$, $\nabla f_{\lambda_1, \lambda_3} = \vec{0}$, $g_1 = 0$ y $g_3 = 0$ si y sólo si $\min_{x, y, \lambda_1, \lambda_3} h(x, y, \lambda_1, \lambda_3) = 0$, donde: $h = (4x + 2y - 12 + 2x\lambda_1 + 2x\lambda_3)^2 + (2x + 10y - 8 + \lambda_1 - \lambda_3)^2 + (x^2 + y - 4)^2 + (x^2 - y)^2$. Verificar con `fminunc` que el óptimo en este caso es $\bar{z} = [1.4142, 2, -7, 7.8284]$.

Aplicar las reglas de Active Set y repetir el proceso para llegar a: $\bar{z} \approx [0.9654, 0.9319, 3.2499]$ con $\lambda_1 = \lambda_2 = 0$ ($\lambda_3 \approx 3.2499$).

Problemas con Igualdades

Cada iteración de KKT y Active Set lleva a un sistema de ecuaciones equivalente a un problema (P) $\begin{cases} \min f(x) \\ h_j(x) = 0 \end{cases}$.

Se sugería (slide previa) minimizar $\|\nabla f_\mu\|_2^2 + \sum_{j=1}^q h_j^2(x)$, pero eso dificulta el problema (e.g. duplica el grado en polinomios).

Cuando hay igualdades, se puede llevar (P') a una forma similar a la de (P) aunque con *variables de holgura* $y_i \geq 0$.

$$(P') \begin{cases} \min f(x) \\ g_i(x) \leq 0 \end{cases} \Rightarrow (P'') \begin{cases} \min f(x) \\ h_i(x) = g_i(x) + y_i = 0 \\ y_i \geq 0 \end{cases}$$

¿No hay algo mejor que elevar al cuadrado para resolver KKT?

¿No son más simples las restricciones de signo que las genéricas?

Programación Primal-Dual

Propiedad

Dado un problema (P) de programación matemática y su dual (D) $\{\max \Phi(\lambda, \mu) | \lambda \geq 0\}$, donde $\Phi(\lambda, \mu)$ es el mínimo de la relajación $(P_{\lambda, \mu})$ alcanzado en $\bar{x}_{\lambda, \mu}$, se cumple $\nabla \Phi(\lambda, \mu) = (g(\bar{x}_{\lambda, \mu}), h(\bar{x}_{\lambda, \mu}))$.

PRUEBA: El punto $\bar{x}_{\lambda, \mu}$ resuelve $\nabla f_{\lambda, \mu}(\bar{x}_{\lambda, \mu}) = \vec{0}$, porque es el óptimo de $(P_{\lambda, \mu})$ y $\Phi(\lambda, \mu) = f_{\lambda, \mu}(\bar{x}_{\lambda, \mu})$. Aplicando la regla de la

cadena a $f_{\lambda, \mu}(\bar{x}_{\lambda, \mu}) = f(\bar{x}_{\lambda, \mu}) + \sum_{i=1}^p \lambda_i g_i(\bar{x}_{\lambda, \mu}) + \sum_{j=1}^q \mu_j h_j(\bar{x}_{\lambda, \mu}) \Rightarrow$

$$\frac{\partial f_{\lambda, \mu}(\bar{x}_{\lambda, \mu})}{\partial \lambda_i} = (\nabla f(\bar{x}_{\lambda, \mu}) + \sum_{i=1}^p \lambda_i \nabla g_i(\bar{x}_{\lambda, \mu}) + \sum_{j=1}^q \mu_j \nabla h_j(\bar{x}_{\lambda, \mu}))^T \cdot \bar{x}'_i$$

+ $g_i(\bar{x}_{\lambda, \mu})$, donde \bar{x}'_i es la derivada de la curva $\bar{x}_{\lambda, \mu}$ al variar λ_i .

Como $\nabla f_{\lambda, \mu}(\bar{x}_{\lambda, \mu}) = \vec{0}$, se cumple $\frac{\partial f_{\lambda, \mu}(\bar{x}_{\lambda, \mu})}{\partial \lambda_i} = g_i(\bar{x}_{\lambda, \mu})$.

Programación Primal-Dual

Propiedad

Dado un problema (P) de programación matemática y su dual (D) $\{\max \Phi(\lambda, \mu) | \lambda \geq 0\}$, donde $\Phi(\lambda, \mu)$ es el mínimo de la relajación $(P_{\lambda, \mu})$ alcanzado en $\bar{x}_{\lambda, \mu}$, se cumple $\nabla \Phi(\lambda, \mu) = (g(\bar{x}_{\lambda, \mu}), h(\bar{x}_{\lambda, \mu}))$.

PRUEBA: El punto $\bar{x}_{\lambda, \mu}$ resuelve $\nabla f_{\lambda, \mu}(\bar{x}_{\lambda, \mu}) = \vec{0}$, porque es el óptimo de $(P_{\lambda, \mu})$ y $\Phi(\lambda, \mu) = f_{\lambda, \mu}(\bar{x}_{\lambda, \mu})$. Aplicando la regla de la

cadena a $f_{\lambda, \mu}(\bar{x}_{\lambda, \mu}) = f(\bar{x}_{\lambda, \mu}) + \sum_{i=1}^p \lambda_i g_i(\bar{x}_{\lambda, \mu}) + \sum_{j=1}^q \mu_j h_j(\bar{x}_{\lambda, \mu}) \Rightarrow$

$$\frac{\partial f_{\lambda, \mu}(\bar{x}_{\lambda, \mu})}{\partial \lambda_i} = (\nabla f(\bar{x}_{\lambda, \mu}) + \sum_{i=1}^p \lambda_i \nabla g_i(\bar{x}_{\lambda, \mu}) + \sum_{j=1}^q \mu_j \nabla h_j(\bar{x}_{\lambda, \mu}))^T \cdot \bar{x}'_i$$

+ $g_i(\bar{x}_{\lambda, \mu})$, donde \bar{x}'_i es la derivada de la curva $\bar{x}_{\lambda, \mu}$ al variar λ_i .

Con el mismo argumento se concluye $\frac{\partial f_{\lambda, \mu}(\bar{x}_{\lambda, \mu})}{\partial \mu_j} = h_j(\bar{x}_{\lambda, \mu})$.

Programación Primal-Dual

Si nuestro problema fuera $(P) \begin{cases} \min f(x) \\ h_j(x) = 0 \end{cases}$, podríamos tomar como *master* su dual $(D) \{ \max \Phi(\mu) \}$. El problema (D) no tiene restricciones en este caso, y lo podemos resolver con alguna de las técnicas vistas en la primera parte (e.g. SD, CG, BFGS).

Buscamos el *máximo* de una función $\Phi : \mathbb{R}^m \rightarrow \mathbb{R}$, que es el resultado de optimizar $f_\mu : \mathbb{R}^n \rightarrow \mathbb{R}$. El esqueleto es el siguiente:

- 1 Se tiene un punto inicial μ_0 .
- 2 Dado μ_n , se minimiza $f_{\mu_n}(x)$ para encontrar \bar{x}_{μ_n} y $\Phi(\mu_n)$.
- 3 Se calcula $\nabla \Phi(\mu_n) = h(\bar{x}_{\mu_n})$, y con él se toma una dirección \vec{d}_n , que debe ser de *ascenso* en este caso.
- 4 Se busca μ_{n+1} en la dirección \vec{d}_n que cumpla $\Phi(\mu_{n+1}) > \Phi(\mu_n)$, y se actualiza el punto (i.e. $\mu_n \leftarrow \mu_{n+1}$, sobrescribir μ_n).
- 5 Si no se cumple la *condición de salida*, volver al punto 2.

Programación Primal-Dual

$$\text{Ejemplo: } (P) \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x^2 + y - 4 \leq 0, \\ -4x + y^2 - 4y \leq 0, \\ x^2 - y \leq 0 \end{cases}$$

Ya sabemos que en el óptimo, la restricción activa será la 3.

Programación Primal-Dual

$$\text{Ejemplo: } (P) \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x^2 - y = 0 \end{cases}$$

Ya sabemos que en el óptimo, la restricción activa será la 3.

Tomamos $\mu_0 = 0$ y resolvemos f_μ (e.g. con `fminunc`) para llegar a $\Phi(0) = -8.2222$ y $\Phi'(0) = 8.1235$.

Como $\Phi(\mu)$ está en \mathbb{R}^1 , el problema es una búsqueda lineal.

Con la primera etapa de Wolfe, vemos que $\bar{\mu}$ está en $[0, 8.1235]$.

Podemos usar bipartición y `fminunc` para llegar a que $\bar{\mu} = 3.2499$ con $[\bar{x}_\mu, \bar{y}_\mu] = [0.9654, 0.9319]$ y $\Phi'(\bar{\mu}) \approx 1.8e^{-5}$.

Programación Primal-Dual

$$\text{Ejemplo: } (P) \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x^2 + y - 4 \leq 0, \\ -4x + y^2 - 4y \leq 0, \\ x^2 - y \leq 0 \end{cases}$$

Ya sabemos que en el óptimo, la restricción activa será la 3.

Tomamos $\mu_0 = 0$ y resolvemos f_μ (e.g. con `fminunc`) para llegar a $\Phi(0) = -8.2222$ y $\Phi'(0) = 8.1235$.

Como $\Phi(\mu)$ está en \mathbb{R}^1 , el problema es una búsqueda lineal.

Con la primera etapa de Wolfe, vemos que $\bar{\mu}$ está en $[0, 8.1235]$.

Podemos usar bipartición y `fminunc` para llegar a que $\bar{\mu} = 3.2499$ con $[\bar{x}_\mu, \bar{y}_\mu] = [0.9654, 0.9319]$ y $\Phi'(\bar{\mu}) \approx 1.8e^{-5}$.

Como en general no sabemos qué desigualdades estarán activas, se necesita una forma de cumplir con $\lambda \geq 0$.

Programación Primal-Dual

Una forma pasa por controlar los signos de λ en la búsquedas lineales del problema (D) $\left\{ \begin{array}{l} \max \Phi(\lambda, \mu) \\ \lambda \geq 0 \end{array} \right.$.

Si $\{\mu_j\} = \emptyset$ y en la iteración n del *master* (D) tenemos: $\lambda^{(n)} = [2, 1, 3]$, $d^{(n)} = -[6, 4, 4]$ (dirección de crecimiento) y determinamos el paso $\tau = 0.5$, llegamos a $\lambda^{(n+1)} = [-1, -1, 1]$.

Para no escapar de la región factible de (D) hay que tomar un paso $\tau' < 0.5$. El problema es con λ_1 y λ_2 , así: $2 - 6\tau' = 0$ implica $\tau' = \frac{1}{3}$, y $1 - 4\tau' = 0$ implica $\tau' = \frac{1}{4}$. Hay que elegir el menor ($\tau' = \frac{1}{4}$), y con él llegamos a $\lambda^{(n+1)} = [\frac{1}{2}, 0, 2]$.

Ésta es la misma idea que usa Simplex para cambiar de vértice.

Hay que tener un mecanismo de protección para cuando el crecimiento es ilimitado por poder violar la frontera **¡cotas?**

Controlarlo en la misma búsqueda lineal es una solución.

Programación Primal-Dual

Ya en la frontera, necesitamos otro mecanismo adicional al de los signos de λ en el problema (D) $\left\{ \begin{array}{l} \max \Phi(\lambda, \mu) \\ \lambda \geq 0 \end{array} \right.$.

Si en la iteración n del *master* (D) tenemos $\lambda^{(n)} = [\frac{1}{2}, 0, 2]$ (ejemplo anterior), no podemos usar la dirección de crecimiento $d^{(n)} = [-1, -2, 1]$ porque λ_2 será negativo para todo $\tau > 0$.

Se puede proyectar $d^{(n)}$ sobre las restricciones activas. Para el gradiente, la nueva dirección $\tilde{d}^{(n)}$ también es de ascenso porque $\langle d^{(n)}, \tilde{d}^{(n)} \rangle \geq 0$, y si $\langle d^{(n)}, \tilde{d}^{(n)} \rangle = 0$ el punto era óptimo.

Siguiendo el ejemplo, para la búsqueda lineal se tomaría $\tilde{d}^{(n)} = [-1, 0, 1]$ que es ortogonal a $[0, 1, 0] \cdot \lambda \geq 0$.

Cuando $\{\mu_j\} \neq \emptyset$, hay que ajustar el μ con el mismo paso τ' que λ .

Programación Primal-Dual

$$\text{Ejemplo: } (P) \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x^2 + y - 4 \leq 0, \\ -4x + y^2 - 4y \leq 0, \\ x^2 - y \leq 0 \end{cases}$$

Tomamos $\lambda^{(0)} = [0, 0, 0]$. Con `fminunc` llegamos a $\Phi(\lambda^{(0)}) = -8.2222$ y $\nabla\Phi(\lambda^{(0)}) = [4.5679, -12.3951, 8.1235]$.
Luego $\tilde{d}^{(0)} = [4.5679, 0, 8.1235]$ y $\tau = 0.1772$.

Así, $\lambda^{(1)} = [0.8092, 0, 1.4391]$, $\Phi(\lambda^{(1)}) = -3.4791$,
 $\bar{x}^{(1)} = [1.2689, 0.6092]$ y $\nabla\Phi(\lambda^{(1)}) = [-1.7806, -7.1414, 1.0010]$.

Tomo $\tilde{d}^{(1)} = [-1.7806, 0, 1.0010]$ y llego a $\tau = 2.6725$, valor para el cual $\lambda^{(2)} = [-3.9495, 0, 4.1143]$, que no es factible. Se debe tomar $\tau' = 0.8092/1.7806 = 0.4545$ y $\lambda^{(2)} = [0, 0, 1.8941]$.

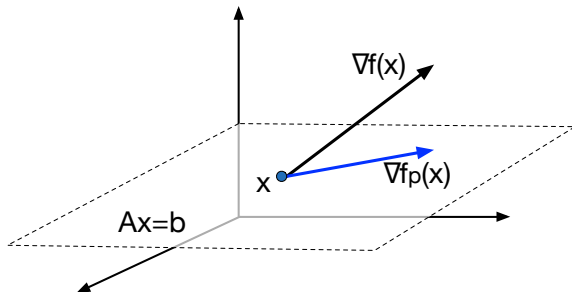
Luego, $\Phi(\lambda^{(2)}) = -1.6909$, $\nabla\Phi(\lambda^{(2)}) = [-1.4421, -7.7823, 1.1216]$,
 $\tilde{d}^{(2)} = [0, 0, 1.1216]$ y $\tau = 1.2088$. Así, $\lambda_1^{(3)} = \lambda_2^{(3)} = 0$, $\lambda_3^{(3)} = 3.2499$ y
 $\bar{x}^{(3)} = [0.9654, 0.9319]$, que es el óptimo.

Gradiente Projectado

En el esquema anterior modificamos la dirección del gradiente y el paso de la búsqueda lineal para no escapar de la región factible.

El gradiente proyectado tiene el mismo espíritu. Es especialmente simple cuando las restricciones son lineales.

Tomando como referencia $(P) \begin{cases} \min f(x) \\ Ax = b \end{cases}$, donde $A \in \mathbb{A}^{m \times n}$.



Gradiente Projectado

En el esquema anterior modificamos la dirección del gradiente y el paso de la búsqueda lineal para no escapar de la región factible.

El gradiente projectado tiene el mismo espíritu. Es especialmente simple cuando las restricciones son lineales.

Tomando como referencia $(P) \begin{cases} \min f(x) \\ Ax = b \end{cases}$, donde $A \in \mathbb{A}^{m \times n}$. Con $n > m = r(A)$, la expresión del gradiente $\nabla f(x)$ projectado en F es:

$$\nabla f_p(x) = (I - A^T(AA^T)^{-1}A)\nabla f(x)$$

Podemos usar $\nabla f_p(x)$ en las búsquedas lineales, incluso cuando tengamos restricciones de signo (misma idea que Primal-Dual).

Puede ser muy eficiente para m bajo (hay que invertir AA^T).

Gradiente Projectado

En el esquema anterior modificamos la dirección del gradiente y el paso de la búsqueda lineal para no escapar de la región factible.

$$\text{Ejemplo: } (P) \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x + y = 3 \end{cases}.$$

$$\text{Como } A = [1, 1], R = (I - A^T(AA^T)^{-1}A) = \begin{bmatrix} \frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & \frac{1}{2} \end{bmatrix}.$$

Tomamos $z_0 = (\frac{3}{2}, \frac{3}{2})$ factible, calculamos $\nabla f(z_0) = [-3, 10]^T$, y hacemos la búsqueda lineal usando $\nabla f_p(z_0) = \frac{13}{2} \begin{bmatrix} -1 \\ 1 \end{bmatrix}$.

La búsqueda es $h(\tau) = f(z_0 - \nabla f_p(z_0)\tau)$ (tomando la dirección de búsqueda como la opuesta al gradiente proyectado).

Concluimos que el paso óptimo es $\tau = 0.2$, para el cual $z = (\frac{14}{5}, \frac{1}{5})$ y $f(z) = -\frac{41}{5}$, que coincide con el óptimo que ya conocíamos.

Gradiente Proyectado

En el esquema anterior modificamos la dirección del gradiente y el paso de la búsqueda lineal para no escapar de la región factible.

$$\text{Ejemplo: } (P') \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x + y + z = 3 \\ z \geq 0 \end{cases} .$$

También podríamos haber partido de $x + y \leq 3$ agregado la variable de holgura z y tomando $(0, 0, 3)$ como punto inicial porque $(0, 0)$ es factible en (P) (cuando esto pasa siempre se pueden tomar las variables de holgura así).

En este caso hay que repetir las consideraciones que vimos en el Primal-Dual: i) ajustar el paso τ si éste me saca de F ; ii) ajustar el gradiente (proyectado en este caso) si estando en la frontera hay una componente negativa incompatible.

Lagrangeano Aumentado

Retoma la idea que resolver KKT en (P) $\begin{cases} \min f(x) \\ h_j(x) = 0 \end{cases}$ es como minimizar $\|\nabla f_\mu\|_2^2 + \sum_{j=1}^q h_j^2(x) = \|\nabla f_\mu\|_2^2 + \|h(x)\|_2^2$.

En este caso se resuelve (P_c) $\begin{cases} \min f(x) + \frac{c}{2}\|h(x)\|_2^2 \\ h_j(x) = 0 \end{cases}$, siendo $c > 0$ una constante cualquiera, que tiene el mismo óptimo que (P) , al ser $h(x) = \vec{0}$ su región factible.

Se minimiza entonces $L(x, \mu) = f(x) + \mu^T h(x) + \frac{c}{2}\|h(x)\|_2^2$.

El Lagrangeano Aumentado tiene aplicaciones numéricas como método de *penalización/barrera exterior*, porque al aumentar c tiende a evitar el salirse de la región factible de (P) .

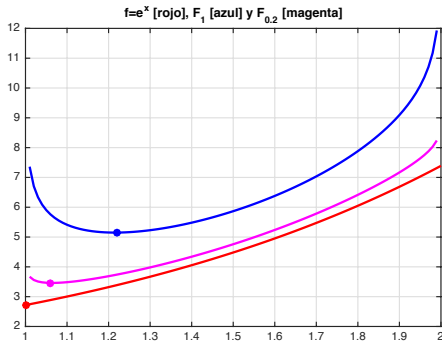
Pero es especialmente usado por sus propiedades teóricas, algunas de las que fundamentan las Condiciones de Segundo Orden, que son suficientes para KKT (si las funciones no son convexas).

Punto Interior

En los métodos de barrera, para un problema (P) $\begin{cases} \min f(x) \\ g_i(x) \geq 0 \end{cases}$, se usa una penalización en lugar de una relajación.

Si $F_\mu(x) = f(x) - \sum_{i=1}^m \mu_i \cdot \log(g_i(x))$ con $\mu = \{\mu_i > 0\}$ dado, tenemos un problema sin restricciones confinado a la región factible de (P) , donde podemos usar optimización sin restricciones.

$$(P) \begin{cases} \min e^x \\ 1 \leq x \leq 2 \end{cases}$$



Punto Interior

En los métodos de barrera, para un problema $(P) \begin{cases} \min f(x) \\ g_i(x) \geq 0 \end{cases}$, se usa una penalización en lugar de una relajación.

Si $F_\mu(x) = f(x) - \sum_{i=1}^m \mu_i \cdot \log(g_i(x))$ con $\mu = \{\mu_i > 0\}$ dado, tenemos un problema sin restricciones confinado a la región factible de (P) , donde podemos usar optimización sin restricciones.

Los algoritmos de descenso no pueden salirse de la región factible de (P) , porque la barrera logarítmica de la función $F_\mu(x)$ tiende a infinito al acercarse a la frontera.

Observar que $\nabla F_\mu(\bar{x}) = \nabla f(\bar{x}) - \sum_{i=1}^m \mu_i \frac{\nabla g_i(\bar{x})}{g_i(\bar{x})} = \vec{0}$ determina el óptimo $\bar{x}(\mu)$ y si usamos el cambio de variable $\mu_i = \lambda_i g_i(\bar{x})$, resulta $\nabla F_\mu(\bar{x}) = \nabla f(\bar{x}) - \sum_{i=1}^m \lambda_i \nabla g_i(\bar{x}) = \vec{0}$, lo que completa la idea que $\bar{x}(\mu)$ tiende al óptimo de (P) cuando $\mu \rightarrow 0$, porque los $\bar{x}(\mu)$ son siempre factibles en (P) , los $\nabla f_\lambda = 0$ y las condiciones de complementaridad tienden a 0.

Punto Interior

$$\text{Ejemplo: } (P) \begin{cases} \min 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ x^2 + y - 4 \leq 0, \\ -4x + y^2 - 4y \leq 0, \\ x^2 - y \leq 0. \end{cases}$$

Tomando $\mu_n = 1/2^{n-1}$ (todos los μ_i iguales en este caso) y usando BFGS conseguimos la secuencia: $(x_1, y_1) = (0.8546, 0.9947)$, $(x_2, y_2) = (0.9056, 0.9627)$, $(x_3, y_3) = (0.9343, 0.947)$, $(x_4, y_4) = (0.9495, 0.9394)$, $(x_5, y_5) = (0.9574, 0.9356)$, $(x_6, y_6) = (0.9614, 0.9338)$ y $(x_7, y_7) = (0.9634, 0.9328)$.

Recordar que el óptimo era $(0.9654, 0.9319)$.

Podemos usar $\lambda_i \approx \mu_k / g_i(x_k)$ para estimar cuáles son las restricciones que tienden a activarse. En este caso, $\lambda_3 \approx 3.2627$ para $k = 7$, y los demás tienden a 0.

El método se puede hibridizar con otros, como active-set.

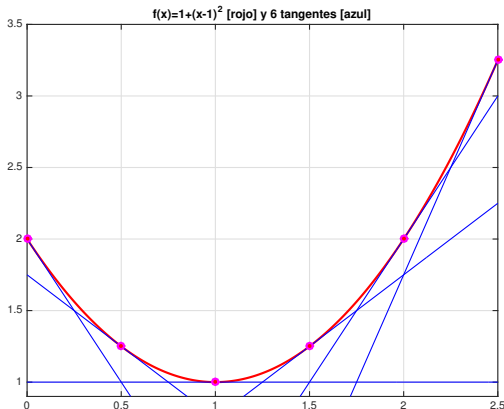
Estrategia de Cortes

En general, las funciones se pueden aproximar muy bien usando polinomios en distintos tramos o regiones.

Un ejemplo es el de las *splines*, cuya versión más notoria usa polinomios cúbicos, pero también se pueden usar polinomios de grado 1 y 2 (se pierde diferenciabilidad en las transiciones).

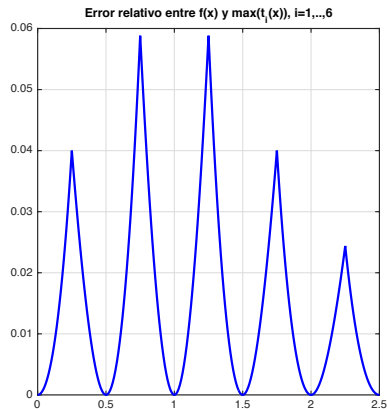
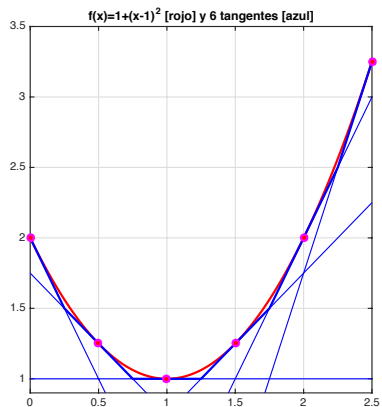
Estrategia de Cortes

Un ejemplo es el de las *splines*, cuya versión más notoria usa polinomios cúbicos, pero también se pueden usar polinomios de grado 1 y 2 (se pierde diferenciabilidad en las transiciones).



Estrategia de Cortes

Un ejemplo es el de las *splines*, cuya versión más notoria usa polinomios cúbicos, pero también se pueden usar polinomios de grado 1 y 2 (se pierde diferenciabilidad en las transiciones).



Estrategia de Cortes

En general, las funciones se pueden aproximar muy bien usando polinomios en distintos tramos o regiones.

Un ejemplo es el de las *splines*, cuya versión más notoria usa polinomios cúbicos, pero también se pueden usar polinomios de grado 1 y 2 (se pierde diferenciabilidad en las transiciones).

El error promedio en el ejemplo anterior es 1.47% y sólo se usaron 6 tangentes, que además, están uniformemente distribuidas.

Cuando una aproximación así se usa en el contexto de optimización sin restricciones, hay que introducir el subgradiente, que es análogo al cono normal $\mathcal{N}(x)$ usado en la prueba de KKT.

En optimización con restricciones, se elude la no-diferenciabilidad reformulando el problema.

Al usar planos tangentes se consigue una relajación del problema, y puede llegarse al óptimo si coinciden.

Estrategia de Cortes

En optimización con restricciones, se elude la no-diferenciabilidad reformulando el problema.

$$(P) \begin{cases} \min_x f(x) \\ x \in F \end{cases} \approx (P') \begin{cases} \min_x \max_i \{t_i(x)\} \\ x \in F \end{cases} \Leftrightarrow (P'') \begin{cases} \min_{x,y} y \\ y \geq t_i(x) \\ x \in F \end{cases}$$

El cambio de variables preserva la convexidad y la diferenciabilidad (cuando existían), pasando lo que era una no-diferenciabilidad de una función a un vértice de la nueva región factible.

Se agrega una variable (no hace diferencia) y tantas restricciones como tangentes (son planos en general).

Lo último requiere agregar muchas restricciones si se busca una aproximación fina, pero son todas lineales y escalan muy bien.

Estrategia de Cortes

¿Cuándo se puede aplicar la aproximación y a qué costo?

	$f(x)$ convexa	$f(x)$ cóncava
$\min f(x)$	Fácil	Combinatorio
$\max f(x)$	Combinatorio	Fácil

El máximo de convexas es convexo (se logra minimizando en y) y el mínimo de cóncavas es cóncavo (se logra maximizando en y).

Como forma de embeber la aproximación del ejemplo en un problema de maximización, puedo usar variables booleanas $\{y_i\}$ y tomar $f(x) = 2 - 2x + y_1(x-0.25) + y_2(x-0.75) + y_3(x-1.25) + y_4(x-1.75) + y_5(x-2.25)$, con $y_1=1 \Leftrightarrow x \geq 0.25$, $y_2=1 \Leftrightarrow x \geq 0.75$, $y_3=1 \Leftrightarrow x \geq 1.25$, $y_4=1 \Leftrightarrow x \geq 1.75$ y $y_5=1 \Leftrightarrow x \geq 2.25$.

Estrategia de Cortes

También se puede escribir como MIP agregando variables z_i reales.

$$\left\{ \begin{array}{l} \max_{x, y_i, z_i} \quad 2 - 2x + z_1 - 0.25y_1 + z_2 - 0.75y_2 + z_3 - 1.25y_3 \\ \quad + z_4 - 1.75y_4 + z_5 - 2.25y_5 \\ y_1 \geq \frac{x-0.25}{2.5}, \quad z_1 \leq x, \quad z_1 \leq 2.5y_1 \\ y_2 \geq \frac{x-0.75}{2.5}, \quad z_2 \leq x, \quad z_2 \leq 2.5y_2 \\ y_3 \geq \frac{x-1.25}{2.5}, \quad z_3 \leq x, \quad z_3 \leq 2.5y_3 \\ y_4 \geq \frac{x-1.75}{2.5}, \quad z_4 \leq x, \quad z_4 \leq 2.5y_4 \\ y_5 \geq \frac{x-2.25}{2.5}, \quad z_5 \leq x, \quad z_5 \leq 2.5y_5 \\ 0 \leq x \leq 2.5, y_i \in \{0, 1\}, z_i \text{ libre} \end{array} \right.$$

Cuando la función no es convexa ni cóncava, el problema es combinatorio, y por tanto difícil en general.

Estrategia de Cortes

La estrategia de cortes se usa de muchas formas (e.g. Benders).
Una aplicación típica es maximizar el dual, con una sucesión (D_k) .

$$\text{Ejemplo: } (P) \begin{cases} \min f = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1 = x^2 + y - 4 \leq 0, \\ g_2 = -4x + y^2 - 4y \leq 0, \\ g_3 = x^2 - y \leq 0 \end{cases}$$

Construimos $(D_k) = \{ \max y \mid A_k \begin{bmatrix} \lambda \\ y \end{bmatrix} \leq b_k, \lambda \geq 0 \}$ (es LP).

Tomamos x_0 y λ_0 factibles y llegamos a Φ_0 y $\nabla\Phi_0$.

La primera restricción de (D_0) es: $y \leq f(x_0)$ (la actualizaremos).

Como (D) es cóncavo: $\Phi(\lambda) \leq \Phi_0 + \nabla\Phi_0^T(\lambda - \lambda_0)$ entonces
agrego la restricción $-\nabla\Phi_0^T\lambda + y \leq \Phi_0 - \nabla\Phi_0^T\lambda_0$ (ésta queda).

$$\text{Así: } A_0 = \begin{bmatrix} 0, \dots, 0, 1 \\ -\nabla\Phi_0^T, 1 \end{bmatrix} \text{ y } b_0 = \begin{bmatrix} f(x_0) \\ \Phi_0 - \nabla\Phi_0^T\lambda_0 \end{bmatrix}.$$

Estrategia de Cortes

La estrategia de cortes se usa de muchas formas (e.g. Benders).
Una aplicación típica es maximizar el dual, con una sucesión (D_k) .

$$\text{Ejemplo: } (P) \begin{cases} \min f = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1 = x^2 + y - 4 \leq 0, \\ g_2 = -4x + y^2 - 4y \leq 0, \\ g_3 = x^2 - y \leq 0 \end{cases}$$

Construimos $(D_k) = \{\max y \mid A_k \begin{bmatrix} \lambda \\ y \end{bmatrix} \leq b_k, \lambda \geq 0\}$ (es LP).

Resuelvo (D_0) (e.g. simplex), con λ_1 resuelvo P_{λ_1} (e.g. BFGS) para llegar a \bar{x}_1 . Si $f(\bar{x}_1)$ es menor que $f(\bar{x}_0)$ actualizo cota. En cualquier caso agrego $-\nabla\Phi_1^T \lambda + y \leq \Phi_1 - \nabla\Phi_1^T \lambda_1$ para (D_1) .
Si registro el más alto de los Φ_k tengo una cota inferior del óptimo.
El más bajo de los $f(\bar{x}_k)$ con \bar{x}_k factible es un cota superior.

Estrategia de Cortes

$$\text{Ejemplo: } (P) \begin{cases} \min f = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1 = x^2 + y - 4 \leq 0, \\ g_2 = -4x + y^2 - 4y \leq 0, \\ g_3 = x^2 - y \leq 0 \end{cases}$$

Tomamos $\bar{x}_0 = [0, 2]$ (factible) y $\lambda_0 = [0, 0, 0]$, para llegar a $\Phi_0 = -8.2222$ y $\nabla\Phi_0 = [4.5679, -12.3951, 8.1235]$.

Resolvemos (D_0): $\lambda_1 = [107.27, 100.68, 108.95]$, $\Phi_1 = -1014.8$ y $\nabla\Phi_1 = [-1.1716, -7.7621, -1.0561] < 0$ y como $f(\bar{x}_1) = 7.457$, actualizamos la cota superior (era $f(0, 2) = 14$).

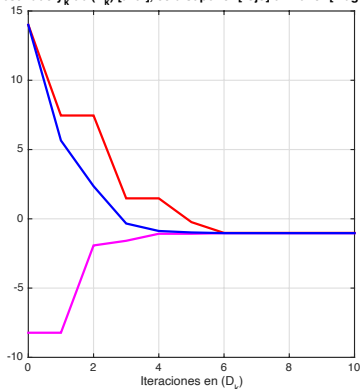
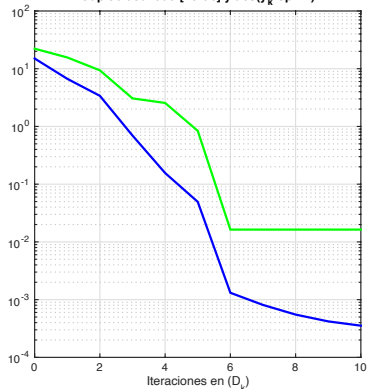
Resolvemos (D_1): $\lambda_2 = [0, 0, 1.7080]$, calculamos $\Phi_2 = -1.9223$ (mejor cota) y $\nabla\Phi_2 = [-1.2607, -8.0028, 1.3712] \not\leq 0$.

Resolvemos (D_2): $\lambda_3 = [0, 0, 4.8289]$, $\Phi_3 = -1.5857$ (mejor cota) y $\nabla\Phi_3 = [-2.3531, -6.1078, -0.6342] \leq 0$ (mejora cota a 1.4767).

Al resolver (D_3): $\lambda_4 = [0, 0, 2.8628]$, $\Phi_4 = -1.0782$ (mejora cota) y $\nabla\Phi_4 = [-2.0140, -6.9508, 0.2350] \not\leq 0$.

Estrategia de Cortes

$$\text{Ejemplo: } (P) \begin{cases} \min f = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1 = x^2 + y - 4 \leq 0, \\ g_2 = -4x + y^2 - 4y \leq 0, \\ g_3 = x^2 - y \leq 0 \end{cases}$$

Resultado y_k de (D_k) [azul], cota superior [rojo] e inferior [magenta]Gap de dualidad [verde] y $\text{abs}(y_k - \text{optim})$ 

Estrategia de Cortes

$$\text{Ejemplo: } (P) \begin{cases} \min f = 2x^2 + 5y^2 + 2xy - 12x - 8y + 10 \\ g_1 = x^2 + y - 4 \leq 0, \\ g_2 = -4x + y^2 - 4y \leq 0, \\ g_3 = x^2 - y \leq 0 \end{cases}$$

En solo 10 iteraciones (11 si contamos la 0), conseguimos estimar el óptimo de (P) con un error relativo de 0.031%.

Hay que recordar que la “*optimización cara*” es la de (P_{λ_k}) , que en el ejemplo resolvimos con `fminunc`.

Al optimizar (D_k) usamos `linprog`, y para la última iteración, la matriz $A \in \mathbb{R}^{13 \times 4}$. Tamaños así en un problema lineal son ridículos.

En la técnica Primal-Dual no-lineal (ver en Derivados de Propiedades Teóricas) tenía menos iteraciones, para debíamos evaluar (P_{λ_k}) varias veces en cada búsqueda lineal.

Programación Cuadrática y Complementaridad Lineal

Definición

Dados $M \in \mathbb{R}^{n \times n}$ y $q \in \mathbb{R}^n$, el Problema de Complementaridad Lineal (LCP) consiste en hallar $z \in \mathbb{R}^n$ tal que: $z \geq 0$, $Mz + q \geq 0$ y $z^T(Mz + q) = 0$ (i.e. $z_i \cdot (Mz + q)_i = 0$, con $1 \leq i \leq n$).

Es un caso particular del problema de programación cuadrática:

$$(QP) \begin{cases} \min \frac{1}{2}x^T Qx + c^T x \\ Ax \geq b \\ x \geq 0 \end{cases},$$

porque $f_\lambda(x) = \frac{1}{2}x^T Qx + (c - A^T \lambda)^T x + b^T \lambda$ y KKT queda:

$$\begin{cases} Qx - A^T \lambda + c \geq 0, \\ Ax - b \geq 0, x \geq 0, \lambda \geq 0, \\ x^T(Qx - A^T \lambda + c) = 0, \\ \lambda^T(Ax - b) = 0 \end{cases}$$

Programación Cuadrática y Complementaridad Lineal

Definición

Dados $M \in \mathbb{R}^{n \times n}$ y $q \in \mathbb{R}^n$, el Problema de Complementaridad Lineal (LCP) consiste en hallar $z \in \mathbb{R}^n$ tal que: $z \geq 0$, $Mz + q \geq 0$ y $z^T(Mz + q) = 0$ (i.e. $z_i \cdot (Mz + q)_i = 0$, con $1 \leq i \leq n$).

$$\begin{cases} Qx - A^T \lambda + c \geq 0, \\ Ax - b \geq 0, x \geq 0, \lambda \geq 0, \\ x^T(Qx - A^T \lambda + c) = 0, \\ \lambda^T(Ax - b) = 0 \end{cases}$$

Que es el LCP cuando $M = \begin{bmatrix} Q & -A^T \\ A & 0 \end{bmatrix}$, $q = \begin{bmatrix} c \\ -b \end{bmatrix}$, si

usamos $z = \begin{bmatrix} x \\ \lambda \end{bmatrix}$.

Programación Cuadrática y Complementaridad Lineal

Teorema (Frank-Wolfe)

Todo problema (QP) acotado inferiormente en su región factible tiene óptimo en ella.

El teorema de Frank-Wolfe generaliza la propiedad probada en el práctico a partir del Simplex.

Corolario

Todo (QP) con Q definida positiva tiene óptimo y el (LCP) asociado tiene solución.

Cuando Q es semidefinida positiva pueden existir casos patológicos donde no se cumple.

Podemos asumir en general que Q es simétrica. En caso contrario, basta tomar $\tilde{Q} = (Q + Q^T)/2$.

Sequential Quadratic Programming (SQP)

Hemos visto estrategias varias, que combinadas con técnicas de optimización sin restricciones son estándares para resolver problemas de programación cuadrática:

$$(QP) \begin{cases} \min \frac{1}{2}x^T Qx + c^T x \\ Ax \leq b \end{cases}$$

Si Q es definida positiva hay existencia del óptimo, y además hay técnicas eficientes para resolver (QP) . Si no, el problema es NP-Hard en general.

En las condiciones de *eficiencia*, el dual está bien definido y hay expresiones analíticas simples para todos los elementos, por lo que la aproximación primal-dual genera un framework que permite resolver los problemas a gran escala.

¿Qué hacer en otros casos convexos no cuadráticos?

Sequential Quadratic Programming (SQP)

Para un problema (P) $\left\{ \begin{array}{l} \min f(x) \\ g_i(x) \leq 0 \\ h_j(x) = 0 \end{array} \right.$, el método SQP propone

generar un sucesión $\{x_n\}$ de aproximaciones de segundo orden, equivalentes al método de Newton pero con restricciones.

A partir de un punto (x_n, λ_n, μ_n) , resuelve el problema de anular el gradiente de $f_{\lambda, \mu}(x)$ con las restricciones, usando una aproximación de segundo orden para $f_{\lambda, \mu}(x)$ y de primero para $g(x)$ y $h(x)$.

$$(SQP_n) \left\{ \begin{array}{l} \min \nabla f_{\lambda, \mu}(x_n) \delta_n + \frac{1}{2} \delta_n^T H[f_{\lambda_n, \mu_n}(x_n)] \delta_n \\ g(x_n) + \nabla^T g(x_n) \delta_n \leq 0 \\ h(x_n) + \nabla^T h(x_n) \delta_n = 0 \end{array} \right.$$

El δ_n se usa como dirección de descenso factible para encontrar x_{n+1} (se concreta con un búsqueda lineal). Para λ_{n+1} y μ_{n+1} se toman los multiplicadores.

Método Simplex

Ajusta ideas del Active-Set para un problema lineal. Se usa la

forma estándar $(P) \begin{cases} \min c^T x \\ Ax = b \\ x \geq 0 \end{cases} (D) \begin{cases} \max b^T \lambda \\ A^T \lambda \leq c \end{cases}$, que es general.

Si tuviéramos que $A_{i1}x_1 + \dots + A_{in}x_n \leq b_i$, agregamos una variable $y_i \geq 0$ de holgura. Las otras excepciones son como en la canónica.

Asumimos $A \in \mathbb{R}^{m \times n}$, con $n > m$, $r(A) = m$ (rango completo) y también (por el momento) que (P) y (D) tienen óptimo.

Todo par (x, λ) factible en (P) y (D) cumple $c^T x \geq b^T \lambda$.

Además $x^T (A^T \lambda - c) = 0$ para que el objetivo de (D) sea $b^T \lambda$, lo que implica: $c^T x = c^T x + (b - Ax)^T \lambda = b^T \lambda + (c - A^T \lambda)^T x = b^T \lambda$.

En la forma estándar, doble factibilidad y la complementaridad $x^T (A^T \lambda - c) = 0$ son necesarias y suficientes para la optimalidad.

Método Simplex

Sabemos que el óptimo de (P) se alcanza en un vértice, y en todo vértice deben haber $n - m$ x 's nulas (se dicen variables no-básicas). Resolviendo $Ax = b$ para las m restantes (variables básicas) encontramos el vértice en cuestión.

Asumimos que x es factible, en particular, que $x_i > 0$ para los i 's que están en la base (caso no-degenerado). Si $x \not\geq 0$ lo descarto. Busco complementaridad con el sistema $(A^T \lambda - c)_i = 0$ para los i 's en la base. El sistema tiene m ecuaciones y m incógnitas. Al resolverlo determinamos λ (no importa el signo).

Por construcción, x es P-factible y $c^T x = b^T \lambda$, así que solamente resta verificar que $A^T \lambda \leq c$, en cuyo caso tengo el óptimo.

!!Probarlos todos requeriría $\binom{n}{n-m} = \binom{n}{m}$ intentos!!

Método Simplex

Sabemos que el óptimo de (P) se alcanza en un vértice, y en todo vértice deben haber $n - m$ x 's nulas (se dicen variables no-básicas). Resolviendo $Ax = b$ para las m restantes (variables básicas) encontramos el vértice en cuestión.

Asumimos que x es factible, en particular, que $x_i > 0$ para los i 's que están en la base (caso no-degenerado). Si $x \not\geq 0$ lo descarto. Busco complementaridad con el sistema $(A^T \lambda - c)_i = 0$ para los i 's en la base. El sistema tiene m ecuaciones y m incógnitas. Al resolverlo determinamos λ (no importa el signo).

Por construcción, x es P-factible y $c^T x = b^T \lambda$, así que solamente resta verificar que $A^T \lambda \leq c$, en cuyo caso tengo el óptimo.

En general no es mandatorio probarlos todos. El método Simplex propone una estrategia greedy que funciona bien en media.

Método Simplex

Si en una iteración tenemos (x, λ) no óptimo, como $c^T x = b^T \lambda$ y x es P-factible, lo que está pasando es que $b^T \lambda$ es demasiado alto, mayor al óptimo de (D) y por eso λ no es factible.

Como existe $(c - A^T \lambda)_j < 0$ con j no-básica (en las básicas era 0), y $c^T x = b^T \lambda + (c - A^T \lambda)_j x_j$ bajamos $c^T x$ (i.e. $b^T \lambda$) aumentando x_j .

Para mantener la consistencia también hay que variar las variables básicas a efectos de mantener $Ax = b$, verificando que $x \geq 0$.

Si podemos aumentar x_j indefinidamente al tiempo que mantenemos x factible en (P) , el problema (D) nunca se hará factible, y eso se debe a que (P) era no-acotado inferiormente.

Si no, una variable x_k positiva (básica) se anulará, con lo que llegamos a otro vértice, reconfigurando la base (i.e. $j \leftrightarrow k$).

No repito vértices porque en cada iteración bajamos el objetivo. Si (P) es acotado el algoritmo converge (número de vértices finito).

Método Simplex (fase-2)

Buscamos resolver (P) $\begin{cases} \min c^T x \\ Ax = b \\ x \geq 0 \end{cases}$ con una base B factible, con

$B \subset \{1, \dots, n\}$. El conjunto $N = \{1, \dots, n\} \setminus B$ son las no-básicas.

- 1 Resuelvo $A_B x_B = b$ ($x_B > 0$ porque es factible), donde A_B son las columnas de A que están en las base B ($x_i = 0$ si $i \in N$).
- 2 Hallo λ con $(c - A^T \lambda)_B = 0$ (filas básicas de $c - A^T \lambda$).
- 3 Si $A^T \lambda \leq c$ estamos en el óptimo (fin algoritmo).
- 4 Si no, tomo $j \in N$ tal que $(c - A^T \lambda)_j < 0$ y aumento x_j en el sistema $Ax = b$ hasta que se anule $x_i \in B$.
- 5 Si ningún $x_i \in B$ se anula, es porque (P) no está acotado inferiormente. Si anulo $x_i \in B$, actualizo B y N y vuelvo a 1.

El método supone que parto de una base factible.

Método Simplex (fase-1)

Cuando no hay una base evidente, recurrimos al Simplex fase-1.

Buscamos una base para (P) $\left\{ \begin{array}{l} \min c^T x \\ Ax = b \\ x \geq 0 \end{array} \right.$. Asumimos que $b \geq 0$,

ya que si $b_i < 0$ multiplicamos por -1 las filas i de A y b .

Considero el problema (P') $\left\{ \begin{array}{l} \min \sum_{i=1}^m y_i \\ Ax + y = b \\ x, y \geq 0 \end{array} \right.$, donde $x = 0$ y $y = b$

siempre es un vértice factible y podemos correr la fase-2. Observar que (P') es mayor o igual a 0, así que el algoritmo converge.

Si el óptimo de (P') es 0, tengo una base factible en (P) porque debe ser $y = 0$ y por tanto $Ax = b$.

Si el óptimo de (P') es mayor que 0, entonces no existía un punto factible en (P) .

Método Simplex (Ejemplo)

$$\text{Considero } (P) \begin{cases} \min x + y \\ 2x + y = 2 \\ x + 2y = 3 \\ x, y \geq 0 \end{cases} \text{ y tomo } (P') \begin{cases} \min u + v \\ 2x + y + u = 2 \\ x + 2y + v = 3 \\ x, y, u, v \geq 0 \end{cases}.$$

En la base inicial, $x=y=0$, $u=2$ y $v=3$ ($B=\{3, 4\}$ y $N=\{1, 2\}$).

Además, $(c - A^T \lambda)_B = 0$ equivale a $\lambda_1 = \lambda_2 = 1$. Por tanto, $(c - A^T \lambda)_N = [-3, -3]^T$. Elegimos x para sacar de la base:

$$\begin{bmatrix} 2 \\ 3 \end{bmatrix} = x \begin{bmatrix} 2 \\ 1 \end{bmatrix} + u \begin{bmatrix} 1 \\ 0 \end{bmatrix} + v \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \text{ de donde: } \begin{cases} u = 2 - 2x \\ v = 3 - x \end{cases},$$

que anula u para $x=1$. El nuevo vértice es $(1, 0, 0, 2)$ ($B=\{1, 4\}$).

$$\text{El } \lambda \text{ sale de } \begin{cases} 2\lambda_1 + \lambda_2 = 0 \\ \lambda_2 = 1 \end{cases}, \lambda_1 = -\frac{1}{2} \text{ y } (c - A^T \lambda)_N = \begin{bmatrix} -3/2 \\ 3/2 \end{bmatrix}.$$

Sacamos y de la base y llegamos a que $x = \frac{1}{3}$, $y = \frac{4}{3}$ es el punto factible inicial. Verificar que ese punto es también (casualmente) el óptimo en (P) .

Descomposición de Problemas

Las técnicas de descomposición explotan la estructura particular de algunos problemas. Por ejemplo, si a y b positivos en:

$$(P) \left\{ \begin{array}{l} \min \sum_{j=1}^n \frac{x_j^2}{2} \\ \sum_{j=1}^n a_j x_j \geq b, \\ x_j \geq 0. \end{array} \right. ,$$

la relajación $(P_\lambda) \{ \min \sum_{j=1}^n \frac{x_j^2}{2} + \lambda(b - \sum_{j=1}^n a_j x_j) \mid x_j \geq 0 \}$, descompone el problema en n independientes.

Cuando esos problemas son más simples, se puede implementar un primal-dual muy eficiente, incluso en problemas grandes.

Descomposición de Problemas

Otro ejemplo es el Knapsack en su versión continua (la versión entera es NP-Hard):

$$(P) \left\{ \begin{array}{l} \max \sum_{j=1}^n a_j x_j \\ \sum_{j=1}^n b_j x_j \leq b, \\ 0 \leq x_j \leq 1 \end{array} \right.$$

Se asume $\{a_j\}$, $\{b_j\}$ y b son positivos. Relajando la restricción que los vincula se descompone el problema, y se concluye que el óptimo es asignar 1 a las variables según orden decreciente de $\frac{a_j}{b_j}$, hasta llenar la mochila (greedy). El último se puede fraccionar.

Programación Estocástica

Entre los problemas que presentan estructuras favorables están los de programación estocástica.

La idea es que el problema tiene E etapas (por ejemplo 3).

La primera es la actual, y no hay incertidumbre. Las restantes abren S escenarios (por ejemplo 5) de probabilidades conocidas.

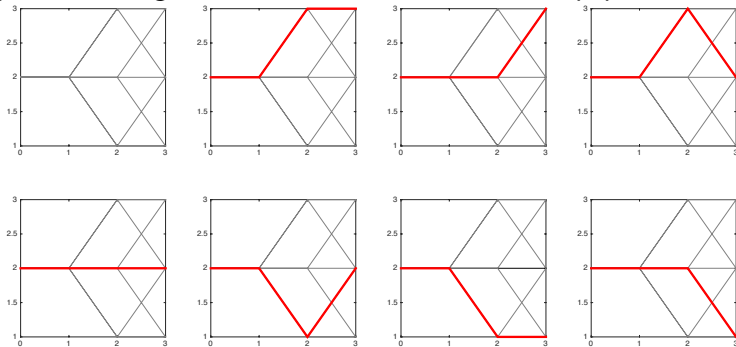
En cada etapa $e \in E$ se sabe en qué grupo o escenario $s_e \in S$ se está, y hay variables de control asociadas a los casos, pero no se conoce cuál será el escenario de la próxima etapa.

Se conoce el árbol que lleva de la etapa inicial a la final.

Se busca un “plan óptimo”: un conjunto de decisiones que minimicen el valor esperado entre las trayectorias posibles, haciendo uso en cada etapa solamente de los datos conocidos.

Programación Estocástica

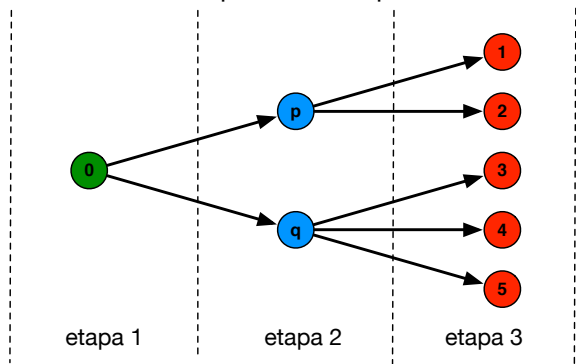
Un ejemplo de cómo confeccionar escenarios es la parte c) de la Opción-1, Obligatorio 5. Si las transiciones son equiprobables.



Tomando t (arriba), m (medio) y b (abajo), las 7 trayectorias con sus probabilidades son: $mmtt$ ($\frac{1}{6}$), $mmmt$ ($\frac{1}{9}$), $mmtm$ ($\frac{1}{6}$), $mmmm$ ($\frac{1}{9}$), $mmbm$ ($\frac{1}{6}$), $mmbb$ ($\frac{1}{6}$) y $mmmb$ ($\frac{1}{9}$).

Programación Estocástica

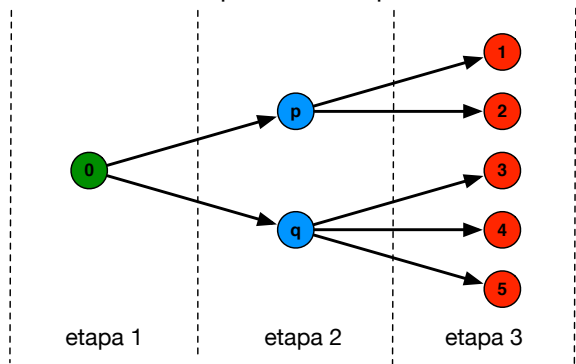
Con 5 escenarios en 3 etapas, el árbol podría ser:



Esto representa que si en la etapa 2 se da p , en la 3 solamente pueden suceder 1 y 2. Habrían 8 tipos de variables en el plan, que denotamos: $x^{(0)}$, $x^{(p)}$, $x^{(q)}$, $x^{(1)}$, $x^{(2)}$, $x^{(3)}$, $x^{(4)}$, $x^{(5)}$.

Programación Estocástica

Con 5 escenarios en 3 etapas, el árbol podría ser:



La variable $x^{(0)}$ tiene las decisiones que hay que tomar hoy, mientras que en $x^{(p)}$ se sabe que el escenario va a ser 1 o 2, pero no cuál. Finalmente, $x^{(i)}$ ($1 \leq i \leq 5$) representa qué hacer en el escenario en cuestión.

Programación Estocástica

En un problema lineal, las restricciones podrían tener la estructura:

$$[A|b] = \left[\begin{array}{c|c|c|c} A_{(1)}^{(0)} & & & b_{(1)}^{(0)} \\ \hline A_{(1)}^{(pq)} & A_{(2)}^{(p)} & & b_{(2)}^{(p)} \\ A_{(1)}^{(pq)} & & A_{(2)}^{(q)} & b_{(2)}^{(q)} \\ \hline A_{(1)}^{(*)} & \bar{A}_{(2)}^{(p)} & A_{(3)}^{(1)} & b_{(3)}^{(1)} \\ A_{(1)}^{(*)} & \bar{A}_{(2)}^{(p)} & & A_{(3)}^{(2)} & b_{(3)}^{(2)} \\ A_{(1)}^{(*)} & & \bar{A}_{(2)}^{(q)} & & A_{(3)}^{(3)} & b_{(3)}^{(3)} \\ A_{(1)}^{(*)} & & \bar{A}_{(2)}^{(q)} & & & A_{(3)}^{(4)} & b_{(3)}^{(4)} \\ A_{(1)}^{(*)} & & \bar{A}_{(2)}^{(q)} & & & & A_{(3)}^{(5)} & b_{(3)}^{(5)} \end{array} \right]$$

El subíndice identifica la etapa, y el supraíndice identifica a el o los nodos comprendidos en el árbol. El problema puede ser enorme, pero la mayoría de los valores de A son 0.

Programación Estocástica

En general, el objetivo es minimizar el valor esperado, aunque hay variantes, como minimizar el peor caso.

De ser el primer caso, la función objetivo sería:

$$c_1^T x^{(0)} + \sum_{j \in \{p, q\}} \bar{p}_j c_{2j}^T x^{(j)} + \sum_{1 \leq i \leq 5} p_i c_{3i}^T x^{(i)},$$

donde p_i son las probabilidades de los escenarios ($\sum p_i = 1$) y las de los nodos intermedios: $\bar{p}_p = p_1 + p_2$ y $\bar{p}_q = p_3 + p_4 + p_5$.

Naturalmente, también se cumple que $\bar{p}_p + \bar{p}_q = 1$.

Descomposición de Benders

Tomamos como referencia inicial al problema

$$(P) \begin{cases} \min_{x_1, x_2} c_1^T x_1 + c_2^T x_2 \\ \text{s.t.} & A_{11}x_1 = b_1 \\ & A_{21}x_1 + A_{22}x_2 = b_2 \\ & x_1, x_2 \geq 0 \end{cases}$$

La idea sería separar problemas según sus variables:

$$(P1) \begin{cases} \min_{x_1} c_1^T x_1 + \alpha(x_1) \\ \text{s.t.} & A_{11}x_1 = b_1 \\ & x_1 \geq 0, x_1 \in \beta \end{cases} \quad (P2) \begin{cases} \min_{x_2} c_2^T x_2 \\ \text{s.t.} & A_{22}x_2 = (b_2 - A_{21}x_1) \\ & x_2 \geq 0 \end{cases}$$

Para un x_1 dado, llamamos $\alpha(x_1)$ al resultado de $(P2)$ (x_1 es un parámetro en este problema). Agregamos $x_1 \in \beta$ simbolizando que pueden haber $x_1 \geq 0$ para los cuales $(P2)$ no tenga solución.

Descomposición de Benders

$$\text{Ejemplo: } (P) \left\{ \begin{array}{l} \min 5x_1 - 3x_2 + x_3 + 2x_4 \\ x_1 + x_2 = 1 \\ 2x_1 + x_3 + 2x_4 = 2 \\ 3x_2 + 2x_3 + x_4 = 2 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array} \right. \text{ . Tomo}$$

$$(P2) \left\{ \begin{array}{l} \min x_3 + 2x_4 \\ x_3 + 2x_4 = (2 - 2x_1) \\ 2x_3 + x_4 = (2 - 3x_2) \\ x_3, x_4 \geq 0 \end{array} \right. , \text{ cuyo óptimo es } \frac{1}{3} \begin{bmatrix} 2 + 2x_1 - 6x_2 \\ 2 - 4x_1 + 3x_2 \end{bmatrix}$$

$$\text{de donde } \alpha(x_1, x_2) = 2(1 - x_1) \text{ y } (P1) \left\{ \begin{array}{l} \min 2 + 3x_1 - 3x_2 \\ x_1 + x_2 = 1 \\ x_1, x_2 \geq 0 \end{array} \right. ?$$

Descomposición de Benders

$$\text{Ejemplo: } (P) \left\{ \begin{array}{l} \min 5x_1 - 3x_2 + x_3 + 2x_4 \\ x_1 + x_2 = 1 \\ 2x_1 + x_3 + 2x_4 = 2 \\ 3x_2 + 2x_3 + x_4 = 2 \\ x_1, x_2, x_3, x_4 \geq 0 \end{array} \right. . \text{ Tomo}$$

$$(P2) \left\{ \begin{array}{l} \min x_3 + 2x_4 \\ x_3 + 2x_4 = (2 - 2x_1) \\ 2x_3 + x_4 = (2 - 3x_2) \\ x_3, x_4 \geq 0 \end{array} \right. , \text{ cuyo óptimo es } \frac{1}{3} \begin{bmatrix} 2 + 2x_1 - 6x_2 \\ 2 - 4x_1 + 3x_2 \end{bmatrix}$$

de donde $\alpha(x_1, x_2) = 2(1 - x_1)$ y $(P1) \left\{ \begin{array}{l} \min 2 + 3x_1 - 3x_2 \\ x_1 + x_2 = 1 \\ x_1, x_2 \geq 0 \end{array} \right. ?$

El óptimo de $(P1)$ sería $[0,1]$, para el cual $x_3 = -\frac{4}{3}$.

Hay que agregar β (restricciones de factibilidad).

Descomposición de Benders

$$\text{Tomando } (P2) \begin{cases} \min x_3 + 2x_4 \\ x_3 + 2x_4 = (2-2x_1) \\ 2x_3 + x_4 = (2-3x_2) \\ x_3, x_4 \geq 0 \end{cases}, \text{ cuyo óptimo es}$$

$$\frac{1}{3} \begin{bmatrix} 2 + 2x_1 - 6x_2 \\ 2 - 4x_1 + 3x_2 \end{bmatrix} \text{ de donde } \alpha(x_1, x_2) = 2(1 - x_1) \text{ si}$$

$$\begin{cases} -x_1 + 3x_2 \leq 1 \\ 4x_1 - 3x_2 \leq 2 \end{cases} \quad (x_{3,4} \geq 0) \quad \text{y } (P1) \begin{cases} \min 2 + 3x_1 - 3x_2 \\ x_1 + x_2 = 1 \\ -x_1 + 3x_2 \leq 1 \\ 4x_1 - 3x_2 \leq 2 \\ x_1, x_2 \geq 0 \end{cases},$$

cuya solución es $[\frac{1}{2}, \frac{1}{2}]$, de donde $\bar{x} = [\frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}]$.

Descomposición de Benders

$$\text{Tomemos } (P2) \begin{cases} \min x_3 + 2x_4 \\ x_3 + 2x_4 = (2-2x_1) \\ 2x_3 + x_4 = (2-3x_2) \\ x_3, x_4 \geq 0 \end{cases}, \text{ cuyo óptimo es}$$

$$\frac{1}{3} \begin{bmatrix} 2 + 2x_1 - 6x_2 \\ 2 - 4x_1 + 3x_2 \end{bmatrix} \text{ de donde } \alpha(x_1, x_2) = 2(1 - x_1) \text{ si}$$

$$\begin{cases} -x_1 + 3x_2 \leq 1 \\ 4x_1 - 3x_2 \leq 2 \end{cases} \quad (x_{3,4} \geq 0) \quad \text{y } (P1) \begin{cases} \min 2 + 3x_1 - 3x_2 \\ x_1 + x_2 = 1 \\ -x_1 + 3x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{cases},$$

cuya solución es $[\frac{1}{2}, \frac{1}{2}]$, de donde $\bar{x} = [\frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}]$.

Podríamos de hecho haber eliminado la segunda desigualdad, o agregar restricciones sólo cuando se violan condiciones.

Descomposición de Benders

En general no se puede conseguir una expresión de $\alpha(x_1)$ y $\beta(x_1)$ para poner en (P1), así que buscamos una aproximación numérica.

Para $\alpha(x_1)$ surge de (P2) $\left\{ \begin{array}{l} \min_{x_2} c_2^T x_2 \\ \text{s.t. } A_{22}x_2 = (b_2 - A_{21}x_1) \\ x_2 \geq 0 \end{array} \right.$, ya que por

dualidad $\alpha(x_1) = \max_{\lambda_2} \{ b_2^T \lambda_2 - \lambda_2^T A_{21}x_1 \mid A_{22}^T \lambda_2 \leq c_2 \}$.

Las restricciones no dependen de x_1 . Como $\alpha(x_1)$ es el máximo de un conjunto de funciones convexas en x_1 (afines en realidad) debe ser convexa, y sabemos que estas funciones se aproximan eficientemente por hiperplanos.

La reconstrucción de $\alpha(x_1)$ se hará iterativamente, agregando *cortes de optimalidad* (como en el primal-dual del obligatorio 5).

La de $\beta(x_1)$ se hará agregando *cortes de factibilidad*, solamente si se detecta una inconsistencia, para evitarla en la próxima iteración.

Descomposición de Benders

Buscamos el mínimo del problema (P) resolviendo iterativamente

$$(P_1^{(k)}) \left\{ \begin{array}{l} \min_{x_1, \alpha} \quad c_1^T x_1 + \alpha \\ \text{s.a.} \quad A_{11} x_1 = b_1 \\ \quad \quad I^{(k)} \cdot [x_1; \alpha] \geq b^{(k)} \\ \quad \quad x_1 \geq 0, \alpha \geq lb \end{array} \right. \quad \text{y } (P2).$$

El esquema general del método de Benders es el siguiente:

- 1 Se toma $I^{(0)} = \emptyset$, $b^{(0)} = \emptyset$ y una cota inferior lb inicial.
- 2 En iter k , se resuelve $(P_1^{(k)})$ para hallar x_1^k , que luego se usa en $(P2)$, para hallar λ_2 .
- 3 Se agrega corte de optimalidad $I_{k+1} = [I_k; \lambda_2^T A_{21}, 1]$, $b^{(k+1)} = [b^{(k)}; b_2^T \lambda_2]$.
- 4 Se actualizan cotas y si no se cumple la *condición de salida*, volver al punto 2.

Benders y Programación Estocástica Lineal

- En un Programación Estocástica Lineal con dos etapas y s escenarios, se usan $\alpha_1, \dots, \alpha_s$ variables.
- El problema original se descompone en un master ($P_1^{(k)}$) y s subproblemas independientes (P_2^i), con $1 \leq i \leq s$.
- Se puede paralelizar fácilmente la ejecución de los subproblemas, sincronizando las soluciones entre iteraciones.
- De haber más etapas, se puede usar *Benders Anidado*: una descomposición recursiva que aplicar el método de Benders de una etapa a la siguiente.