Programación Lineal
Algoritmo de Simplex
Clase 16

Investigación Operativa UTN FRBA 2021
Curso: I4051
Docente: Martín Palazzo

Programación Lineal

Un problema genérico de optimización a ser resuelto por programación lineal se estructura matricial y vectorialmente de la siguiente forma:

$$\begin{aligned} \mathbf{c}^T\mathbf{x} &= \mathbf{z} \longrightarrow \text{Max} \\ A\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq 0 \end{aligned}$$

Donde **c** es un vector [n,1] de 'n' coeficientes, **x** es un vector [n,1] de 'n' variables de decisión, **A** es una matriz [r,n] de coeficientes (tecnológicos) sobre las 'n' variables de decisión **x** en las 'r' restricciones **b**. Tanto **c**, **b** y **A** son conocidos mientras que **x** es la incógnita multidimensional a averiguar. Notar que cuando n>3 nos encontramos con un problema de alta dimensión que no puede ser resuelto por el método gráfico.

Solucionar P.L. con Simplex

El método gráfico de resolución es útil para ganar intuición del problema y es fácil de explicar aunque es un método que no puede generalizar para más de 3 dimensiones, además queda sujeto a dibujos que pueden ser poco exactos.

De todos modos el método gráfico nos deja un aprendizaje: **la solución óptima siempre cae en uno de los vértices** (en casos no-especiales) de la región de soluciones básicas factibles (SBF) cuando no consideramos los casos especiales.

El algoritmo/método del SIMPLEX es una solución algebraica caracterizada por encontrar una solución **óptima SBFO** luego de **'t' iteraciones**. El SIMPLEX se basa en la idea de buscar la solución únicamente en los vértices del área de posibles SBF.

Programación Lineal

En caso que n = 2 y r = 2 el planteo del problema tendría la siguiente forma:

$$
\begin{aligned}
c_1x_1 + c_2x_2 &= \mathbf{z} \\
a_{11}x_1 + a_{12}x_2 &\leq b_1 \\
a_{21}x_1 + a_{22}x_2 &\leq b_2 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

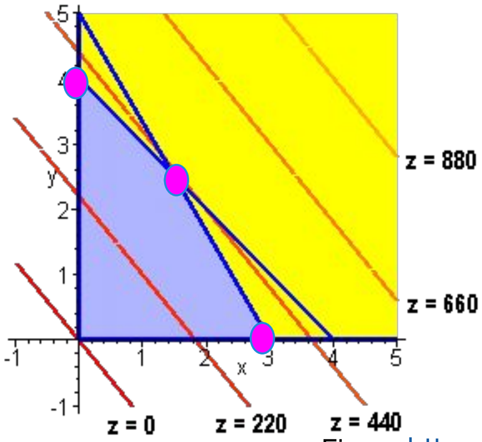


Figura https://services.math.duke.edu/

Donde el área azul contiene las soluciones básicas factibles (SBF) limitadas por las 2 restricciones **ax** =< **b** y por las condiciones de no-negatividad. La recta roja representa a la función objetivo **z** a optimizar y su valor estará en función de cuàn lejos del origen se encuentre. Los puntos en fucsia representan la soluciones en los vértices del área de SBF.

Solucionar P.L. con Simplex

Lo primero que vamos a hacer para preparar el problema de P.L. al formato simplex es convertir las inecuaciones de las restricciones en ecuaciones. Para eso se agregarán variables ‘dummies’ o ‘slack’, también llamadas variables de holgura y las restricciones pasarán a ser ecuaciones.

$$... \leq b \rightarrow \; .. + \xi = b$$
$$... \geq b \rightarrow \; .. - \xi = b$$

El hecho de tener un problema planteado con un sistema de ecuaciones lineales permite que se puedan aplicar herramientas de álgebra lineal como Gauss Jordan y otras más. Las variables slack pueden incorporarse de 2 maneras:

- Sumandolas a la combinación lineal de **Ax** en el caso de restricciones de menor o igual. El valor que tome Xi en este caso representará la cantidad de recurso no utilizado.
- Restándolas a la combinación lineal de **Ax** en el caso de restricciones de mayor o igual. El valor que tome Xi en este caso representará la cantidad de recurso excedente.

Solucionar P.L. con Simplex

$$\begin{aligned} c_1x_1 + c_2x_2 \quad &= \mathbf{z} \\ a_{11}x_1 + a_{12}x_2 &\leq b_1 \\ a_{21}x_1 + a_{22}x_2 &\leq b_2 \end{aligned}$$

Problema original: función objetivo lineal con restricciones lineales

$$\begin{aligned} c_1x_1 + c_2x_2 + 0x_3 + 0x_4 \quad &= \mathbf{z} \\ a_{11}x_1 + a_{12}x_2 + x_3 + 0x_4 &= b_1 \\ a_{21}x_1 + a_{22}x_2 + 0x_3 + x_4 &= b_2 \end{aligned}$$

Problema adaptado para resolverlo con Simplex: Sistema de ecuaciones lineales (agregando las variables slack x3 y x4).

Notar que se agregan tantas variables slack como cantidad de restricciones y que éstas solo participan en la restricción en cuestión, teniendo coeficiente 0 para la función objetivo y el resto de las restricciones.

Algoritmo del Simplex

$$
\begin{aligned}
c_1x_1 + c_2x_2 + 0x_3 + 0x_4 &= \mathbf{z} \\
a_{11}x_1 + a_{12}x_2 + x_3 + 0x_4 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + 0x_3 + x_4 &= b_2
\end{aligned}
$$

El sistema de ecuaciones con las variables slack será el utilizado para inicializar el algoritmo del simplex. Supongamos que en la iteración t=0 el sistema se encuentra en una solución inicial en el origen x1=0 y x2=0, entonces z = 0. En el momento de inicializar el problema, ¿cuál sería la solución a partir de la cuál se empieza a iterar y optimizar? ¿cómo se interpretaría dicha solución?

Algoritmo del Simplex

$$
\begin{aligned}
(t &= 0) \\
x_1 &= 0 \\
x_2 &= 0 \\
x_3 &= b_1 \\
x_4 &= b_2
\end{aligned}
$$

En la iteración t=0 suponiendo que se comienza desde el origen las variables de decisión x1 = 0 y x2 = 0. Esto se puede interpretar como que no se activó ninguna de las variables de decisión, entonces no se consumió ningún recurso. Por lo tanto en t=0 la solución está gobernada íntegramente por las variables slack que representan la cantidad de recurso no utilizado.

Vamos a decir que todas las variables Xi > 0 serán parte de la solución y serán llamadas '**variables básicas**'.

Algoritmo del Simplex

Función Objetivo	$Z_0 = 0$	$Z_1 = \#\# > Z_0$	$Z_2 = \#\# > Z_1$	$Z_3 = \#\# > Z_2$
Solución básica	$Xi = \#$ $Xj = \#$	$Xi = \#$ $Xk = \#$	$Xe = \#$ $Xk = \#$	$Xe = \#$ $Xh = \#$
Iteración	T = 0	T = 1	T = 2	T = 3

Para aproximar a la solución básica factible óptima (SBFO) el SIMPLEX en cada iteración ingresará una variable no-básica a la solución básica y quitará una variable básica para transformarla en no-básica con el fin de optimizar la función objetivo iterativamente.

La política que toma el SIMPLEX para ingresar una variable a la base es agotando en cada iteración el recurso/restricción que cuyo agotamiento genere el mayor incremento del funcional **Z**. Esto quiere decir que el SIMPLEX siempre considerará soluciones en los **límites** de la región factible de soluciones, concretamente al agotar de a 1 recurso a la vez el SIMPLEX considerará posibles soluciones en los vértices de la región. El simplex iterará hasta que el funcional Z no se modifique mas.

Algoritmo del Simplex

Del sistema de ecuaciones podemos construir la tabla inicial del simplex. Esta tabla contempla los coeficientes que son valores conocidos, la solución actual y el valor de Z para la iteración en cuestión.

T = 0			c1	c2	c3	c4
Coef. en Z de var. básica	Variable Básica	**Bi**	**x1**	**x2**	**x3**	**x4**
C3 = 0	x3	b1	a11	a12	a13 = 1	a14 = 0
C4 = 0	x4	b2	a21	a22	a23 = 0	a24 = 1
Z = 0	Zj - Cj		-x1*c1	-x2*c2	0	0

En t=0 la solución básica factible es x3 = b1 y x4 = b2 con Z = 0. Notar que el recuadro violeta indica con la **matriz de identidad** las variables que están en la base.

Algoritmo del Simplex

Ingreso de una variable no-básica a la base: En cada iteración para determinar que variable debe entrar a la solución básica se observará el Zj-Cj de las variables no-básicas. Zj-Cj representa el costo de oportunidad que tiene de incrementar el funcional Z una variable no-basica al ingresar a la solución básica.

T = 0			c1	c2	c3	c4
Coef. en Z de var. básica	Variable Básica	**Bi**	**x1**	**x2**	**x3**	**x4**
C3 = 0	x3	b1	a11	a12	1	0
C4 = 0	x4	b2	a21	a22	0	1
Z = 0	Zj - Cj		-x1*c1	-x2*c2	0	0

Es decir que se definirá el ingreso de la variable no-básica que mayor valor absoluto de Zj - Cj tenga. Supongamos que entra a al base la variable X1 por que x1*c1 tiene el mayor valor absoluto.

Algoritmo del Simplex

Egreso de una variable básica de la base: En cada iteración para determinar que variable debe salir de la solución básica se calculará para cada variable básica su respectivo Bi/Aik siendo k la variable que decidimos que entra en la base (ver paso anterior en el ejemplo seria k = x1).

T = 0			c1	c2	c3	c4
Coef. en Z de var. básica	Variable Básica	**Bi**	**x1**	**x2**	**x3**	**x4**
C3 = 0	x3	b1	a11	a12	1	0
C4 = 0	x4	b2	a21	a22	0	1
Z = 0	Zj - Cj		-x1*c1	-x2*c2	0	0

Calcular el [Bi/Aik] para cáda variable básica, saldrá de la solución la que menor [Bi/Aik] reporte. Esto se podría interpretar como que saldrá de la solución la variable básica que más rápido se agote. El SIMPLEX busca agotar recursos para ir encontrando la solución. Supongamos que X4 reporta el menor [Bi/Aik] y sale.

Algoritmo del Simplex

Una vez que se decide que variable entra y sale de la base debemos actualizar la tabla del simplex para la iteracion t=1 por Gauss-Jordan. El elemento pivote será la posición de intersección entre la fila de la variable básica que sale y la columna de la variable no-básica que entra.

T = 0			c1	c2	c3	c4
Coef. en Z de var. básica	Variable Básica	**Bi**	**x1**	**x2**	**x3**	**x4**
C3 = 0	x3	Actualizar por G.J.	0	Actualizar por G.J.	1	Actualizar por G.J.
C4 = 0	x4	#/a21	a21 = 1	#/a21	0	#/a21
Z = #	Zj - Cj		0	Actualizar por G.J.	Actualizar por G.J.	Actualizar por G.J.

- Las columnas correspondientes a las variables que permanecen en la base se mantienen igual en t = t+1
- Todos los elementos de la columna del pivot se llevan a 0 y el pivot a 1 en t=t+1.
- El renglón del pivot se divide por el valor del pivot
- El resto de las celdas se deben actualizar por Gauss-Jordan utilizando el pivot en t=0.

Algoritmo del Simplex

Para actualizar las celdas de la próxima iteración utilizaremos la regla de gauss-jordan de la siguiente manera:

Valor nuevo de la celda = valor actual de la celda - [("valor en el renglón del pivot" * "valor en la columna del pivot")/pivot]

K	Pc	
Pf	**P**	

$$K' = K - [(Pf * Pc)/P]$$

- Las columnas correspondientes a las variables que permanecen en la base se mantienen igual en t = t+1
- Todos los elementos de la columna del pivot se llevan a 0 y el pivot a 1 en t=t+1.
- El renglón del pivot se divide por el valor del pivot
- El resto de las celdas se deben actualizar por Gauss-Jordan utilizando el pivot en t=0.

Algoritmo del Simplex

T = 1			c1	c2	c3	c4
Coef. en Z de var. básica	Variable Básica	**Bi**	**x1**	**x2**	**x3**	**x4**
C3 = 0	x3	##	0	##	1	##
C1	x1	##	1	##	0	##
Z = ##	Zj - Cj		0	##	##	##

Luego de la primer iteración la solución básica ahora tiene una nueva variable adentro y otra que salió. En este ejemplo genérico salió x4 y entró x1. Se puede observar que en el cuadro de la derecha para estas 2 variables se forma la matriz de identidad. El valor del funcional Z debería haber incrementado su valor. Siempre que queden Zj-Cj < 0 quiere decir que hay costo de oportunidad para mejorar, es decir que hay lugar para una nueva iteración del SIMPLEX.

Ejemplo

Optimización con SIMPLEX

Programación Lineal: toy example

$$\max_x \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned}
10x_1 + 5x_2 &\leq 200 \\
2_x 1 + 3x_2 &\leq 60 \\
x_1 &\leq 34 \\
x_2 &\leq 14 \\
x_1, x_2 &\geq 0
\end{aligned}$$

$$\boldsymbol{maxZ} = cX$$

$$Ax \leq b$$

$$x \geq 0$$

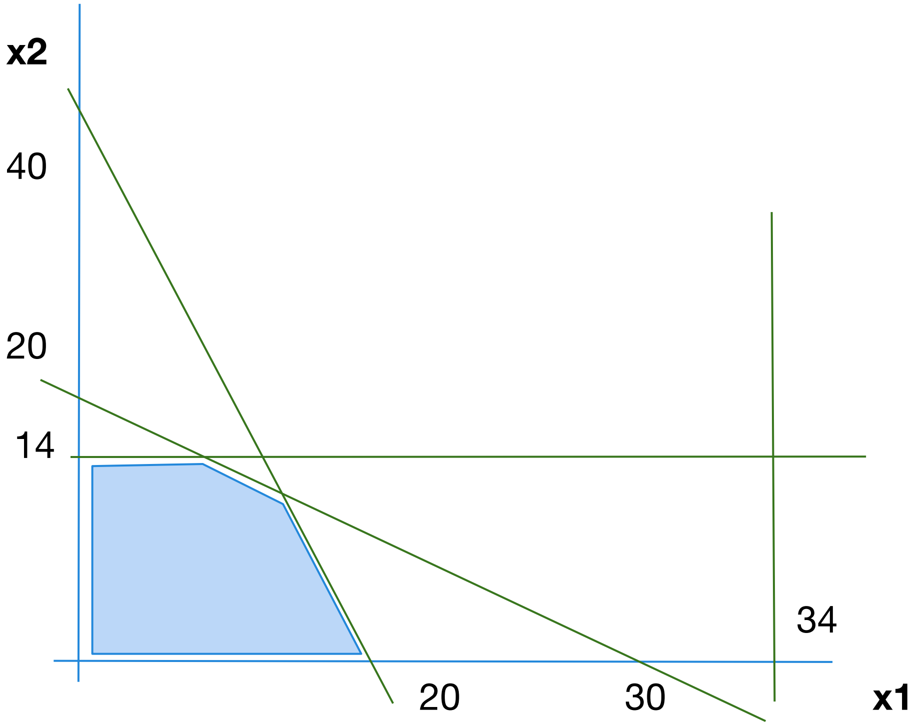
$$x = (x_1, x_2) \quad c = (1000, 1200)$$

$$A = \begin{bmatrix} 10 & 5 \\ 2 & 3 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \quad b = (200, 60, 34, 14)$$

Programación Lineal: toy example

$$\max_{x} \qquad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned}
10x_1 + 5x_2 &\leq 200 \\
2_x 1 + 3x_2 &\leq 60 \\
x_1 &\leq 34 \\
x_2 &\leq 14 \\
x_1, x_2 &\geq 0
\end{aligned}$$



Programación Lineal: variables slack

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned} 10x_1 + 5x_2 &\leq 200 \\ 2_x1 + 3x_2 &\leq 60 \\ x_1 &\leq 34 \\ x_2 &\leq 14 \\ x_1, x_2 &\geq 0 \end{aligned}$$

$$\rightarrow$$

$$\boldsymbol{maxZ} = 1000\boldsymbol{x_1} + 1200\boldsymbol{x_2}$$

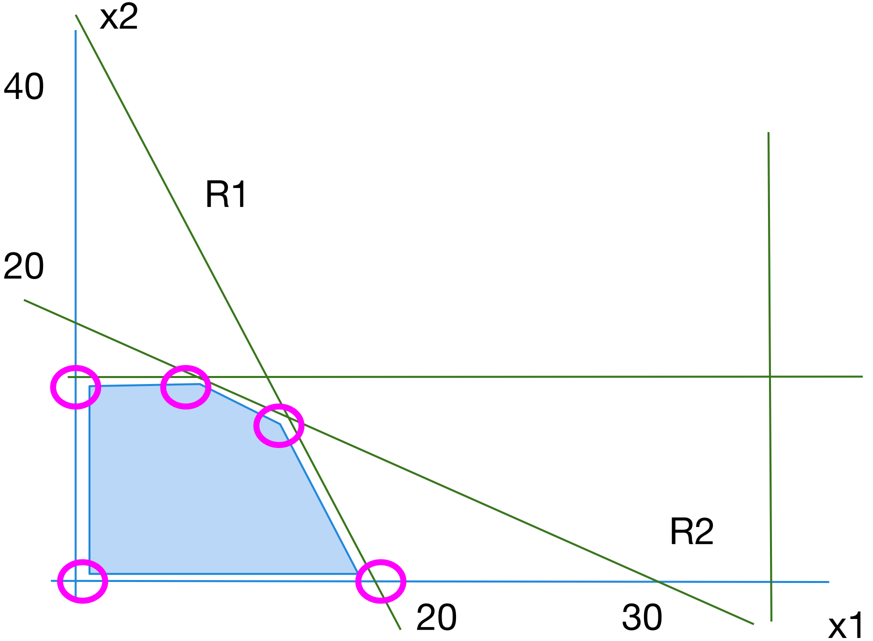
$$\begin{aligned} 10x_1 + 5x_2 + x_3 + 0x_4 + 0x_5 + 0x_6 &= 200 \\ 2x_1 + 3x_2 + 0x_3 + x_4 + 0_x5 + 0_x6 &= 60 \\ x_1 + 0x_3 + 0x_4 + x_5 + 0x_6 &= 34 \\ x_2 + 0x_3 + 0x_4 + 0x_5 + x_6 &= 14 \end{aligned}$$

$$x_1, x_2, x_3, x_4, x_5, x_6 \geq 0$$

Programación lineal: vértices del polígono cerrado

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned} 10x_1 + 5x_2 &\leq 200 \\ 2_x1 + 3x_2 &\leq 60 \\ x_1 &\leq 34 \\ x_2 &\leq 14 \\ x_1, x_2 &\geq 0 \end{aligned}$$



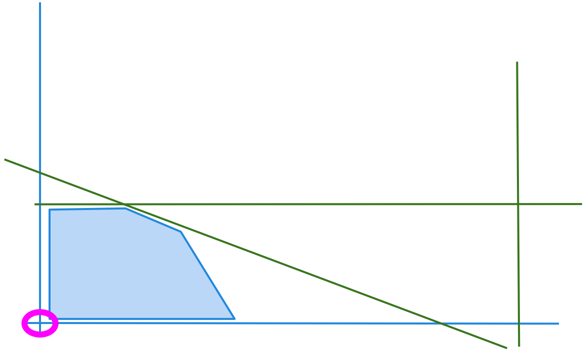
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	**Var. Basica**	**Bk**	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**
0	x3	200	10	5	1	0	0	0
0	x4	60	2	3	0	1	0	0
0	x5	34	1	0	0	0	1	0
0	x6	14	0	1	0	0	0	1
0	zj - cj		-1000	-1200	0	0	0	0

Tabla01: tabla inicial del simplex

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned}
10x_1 + 5x_2 &\leq 200 \\
2_x1 + 3x_2 &\leq 60 \\
x_1 &\leq 34 \\
x_2 &\leq 14 \\
x_1, x_2 &\geq 0
\end{aligned}$$



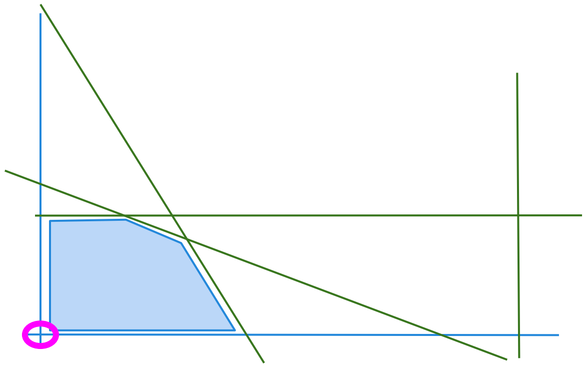
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	**Var. Basica**	**Bk**	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**
0	x3	200	10	5	1	0	0	0
0	x4	60	2	3	0	1	0	0
0	x5	34	1	0	0	0	1	0
0	**x6**	**14**	**0**	**1 (pivot)**	**0**	**0**	**0**	**1**
0	**zj - cj**		-1000	-1200	0	0	0	0

Tabla02: se observa que la variable x2 presenta el menor zj - cj (variable que entra a la base) y la variable hasta ahora básica x6 presenta el menor bK / Aik = 14 / 1 = 14 (variable que sale de la base). La celda que se encuentra en la intersección se la denomina "Pivot".

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned} 10x_1 + 5x_2 &\leq 200 \\ 2_x1 + 3x_2 &\leq 60 \\ x_1 &\leq 34 \\ x_2 &\leq 14 \\ x_1, x_2 &\geq 0 \end{aligned}$$



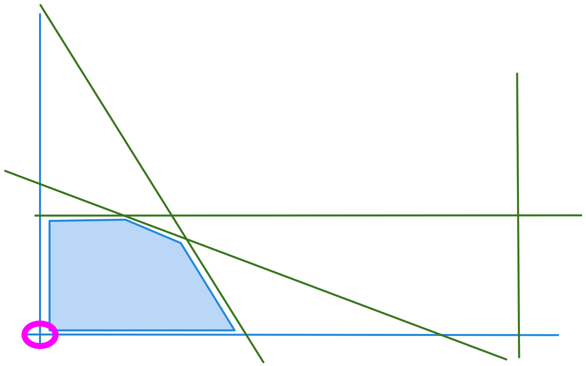
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	Var. Basica	Bk	x1	x2	x3	x4	x5	x6
0	x3	200	10	5	1	0	0	0
0	x4	60	2	3	0	1	0	0
0	x5	34	1	0	0	0	1	0
0	x6	14	0	1 (pivot)	0	0	0	1
0	zj - cj		-1000	-1200	0	0	0	0

Tabla 03 **Valor nuevo de la celda = valor actual de la celda - [(“valor en el renglón del pivot” * “valor en la columna del pivot”)/pivot]**

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned} 10x_1 + 5x_2 &\leq 200 \\ 2_x1 + 3x_2 &\leq 60 \\ x_1 &\leq 34 \\ x_2 &\leq 14 \\ x_1, x_2 &\geq 0 \end{aligned}$$



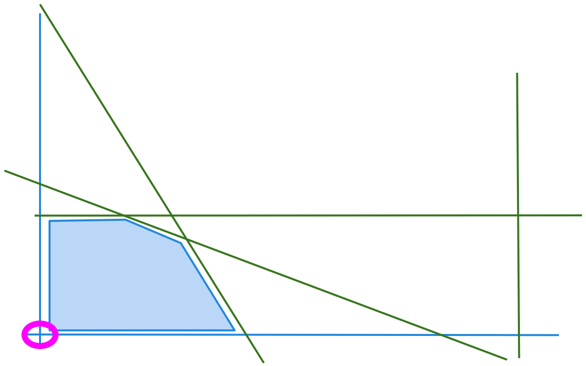
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	**Var. Basica**	**Bk**	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**
0	x3	***calcular por g.j.***	***calcular por g.j.***	5	1	0	0	***calcular por g.j.***
0	x4	***calcular por g.j.***	***calcular por g.j.***	3	0	1	0	***calcular por g.j.***
0	x5	***calcular por g.j.***	***calcular por g.j.***	0	0	0	1	***calcular por g.j.***
0	**x6**	**14**	**0**	**1 (pivot)**	**0**	**0**	**0**	**1**
0	**zj - cj**		*calcular por g.j*	0	*calcular por g.j.*	*calcular por g.j.*	*calcular por g.j.*	*calcular por g.j.*

Tabla04: se detallan las celdas que deben ser calculadas por gauss jordan.

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned} 10x_1 + 5x_2 &\leq 200 \\ 2_x 1 + 3x_2 &\leq 60 \\ x_1 &\leq 34 \\ x_2 &\leq 14 \\ x_1, x_2 &\geq 0 \end{aligned}$$



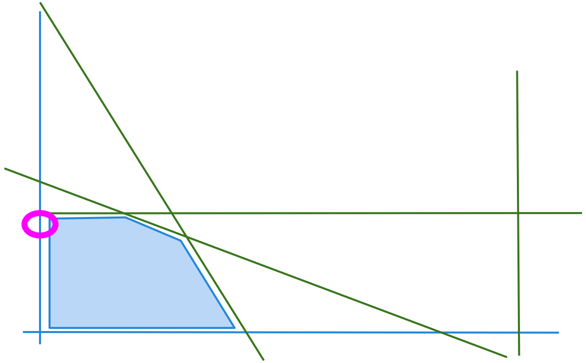
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	**Var. Basica**	**Bk**	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**
0	x3	130	10	0	1	0	0	-5
0	x4	18	2	0	0	1	0	-3
0	x5	34	1	0	0	0	1	0
1200	x2	14	0	1	0	0	0	1
16800	**zj - cj**		-1000	0	0	0	0	1200

Tabla05: es la tabla del simplex luego de la primera iteración, al inicio de la segunda.

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned}
10x_1 + 5x_2 &\leq 200\\
2_x1 + 3x_2 &\leq 60\\
x_1 &\leq 34\\
x_2 &\leq 14\\
x_1, x_2 &\geq 0
\end{aligned}$$



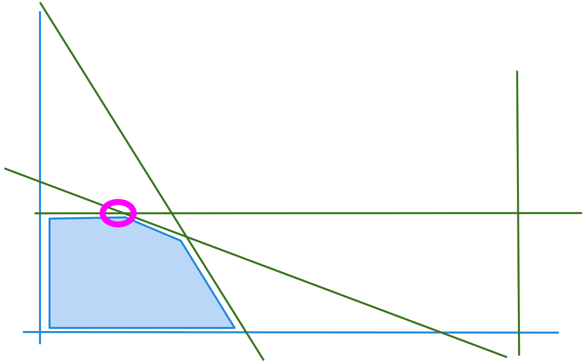
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	Var. Basica	Bk	x1	x2	x3	x4	x5	x6
0	x3	40	0	0	1	-5	0	10
1000	x1	9	1	0	0	1/2	0	-3/2
0	x5	25	0	0	0	-1/2	1	3/2
1200	x2	14	0	1	0	0	0	1
25800	zj - cj		0	0	0	500	0	-300

Tabla06: tabla al inicio de la tercer iteración.

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned}
10x_1 + 5x_2 &\leq 200 \\
2_x1 + 3x_2 &\leq 60 \\
x_1 &\leq 34 \\
x_2 &\leq 14 \\
x_1, x_2 &\geq 0
\end{aligned}$$



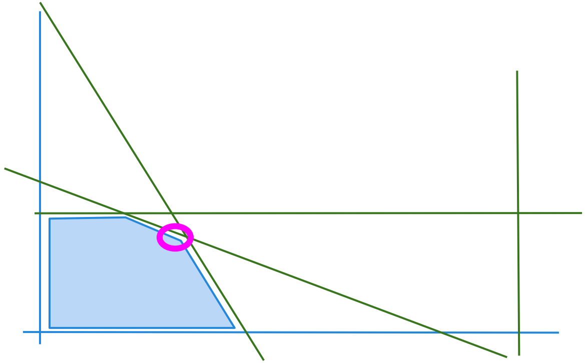
Programación lineal: resolución con SIMPLEX

coef en Z (cj) ->			1000	1200	0	0	0	0
coef en Z de var. basica	**Var. Basica**	**Bk**	**x1**	**x2**	**x3**	**x4**	**x5**	**x6**
0	x6	4	0	0	1/10	-1/2	0	1
1000	x1	15	1	0	3/20	-1/4	0	0
0	x5	19	0	0	-3/20	1/4	1	0
1200	x2	10	0	1	-1/10	1/2	0	0
27000	zj - cj		0	0	30	350	0	0

Tabla07: tabla final con la optimización resuelta.

$$\max_{x} \quad Z = 1000x_1 + 1200x_2$$

$$\begin{aligned} 10x_1 + 5x_2 &\leq 200 \\ 2_x1 + 3x_2 &\leq 60 \\ x_1 &\leq 34 \\ x_2 &\leq 14 \\ x_1, x_2 &\geq 0 \end{aligned}$$



Algoritmo del Simplex

Conclusión

En cada iteración del simplex entrará una variable no-básica a la solución y saldrá una de la base. Esta iteración generará que el funcional se incremente hasta cierto punto donde hacer entrar una variable no-basica a la base no optimizará el funcional, es decir cuando el $Z_j - C_j >= 0$ (en el caso de maximización).

Todas la variables 'activadas' estarán en la base, tengan o no coeficiente en el funcional. Si una variable 'slack' finaliza >0 en la base al terminar el SIMPLEX entonces podemos concluir que ese recurso sobra en la solución óptima.