Pontificia Universidad Católica de Chile
Escuela de Ingeniería
Departamento de Ingeniería Industrial y de Sistemas
Profesores: Gustavo Angulo, Raimundo Cuadrado,
Camila Balbontin, Gonzalo Pérez
Ayudante Coordinador de Tareas: Juan Enrique Hurtado

ICS1113 - Optimización
1er semestre del 2024

Tarea 2 ICS1113

Reglas de la tarea:

- La tarea se puede realizar de a pares o de forma individual (las parejas pueden ser de diferentes secciones). En caso de ser de a pares se debe inscribir con su pareja en los grupos de Canvas.
- La tarea debe ser subida **por solo uno de los integrantes** a Canvas.
- La tarea se entrega el día 27/05/2024 a las 23:59 horas.
- Existirá la opción de usar un cupón de atraso solamente una vez entre las tres tareas, dando la posibilidad de entregar hasta 24 horas después de finalizada la entrega. Para poder acceder a este cupón se deben inscribir con el nombre de los integrantes de la tarea en un Google Forms que será habilitado en su debido tiempo y que cierra junto a la fecha oficial de la entrega. En caso de ser de a pares ambos deben tener un cupón disponible y se usará uno por cada uno y tendrán 24 horas extra (no 48).
- Las tareas atrasadas no serán corregidas, así que recuerden entregarlas a tiempo.
- Esta tarea es grupal y el desarrollo y discusión debe ocurrir dentro de cada grupo. No se distribuyan la resolución de las preguntas por separado, hagan realmente un trabajo grupal de desarrollo ya que el no hacerlo va contra la idea de aprendizaje colaborativo y preparación individual de la interrogación a la que está asociada la tarea. Pueden discutir los problemas con los profesores y los ayudantes del curso, pero al final cada grupo debe entregar sus propias soluciones, desarrolladas y escritas por el grupo. La copia o intento de copia a otros grupos será sancionada dependiendo la gravedad (a ser determinada por el equipo docente del curso) con consecuencias que podrían ir desde un 1.0 en la nota de la Tarea hasta la escalación a la Dirección de Docencia de la Escuela, con posible **REPROBACIÓN AUTOMÁTICA** del curso.
- Por entregar en latex se darán 3 décimas para la tarea entregada, las décimas no son transferibles a otras actividades. Para subirlo tendrá que enviar un archivo .pdf y el archivo .tex correspondientes (también se puede incluir el .zip de Overleaf). En caso de no entregar en latex basta con subir un archivo .pdf que contenga las respuestas. Si se hace a mano, debe ser una tarea legible o habrán descuentos a criterio del corrector. En ambos casos los .pdf tienen que contener los nombres de los integrantes.
- Se abrirá un foro en Canvas donde se responderán las dudas que puedan surgir del enunciado.

Pregunta 1 **[20 Puntos]**

Simplex y programción lineal

1. En la siguiente pregunta se le solicita usar el algoritmo simplex. Cuando se le pida graficar debe incluir imágenes muy claras que incluyan explícitamente todos los puntos de interés solicitados. Puede adjuntar imágenes de gráficos hechos a mano o graficados con software (no se descontará en el bonus por látex). Puede usar calculadoras de matrices como Mathematica, pero debe explicitar los pasos que realiza en su desarrollo. Considere el siguiente problema de optimización:

$$\begin{aligned} \text{P) max} \quad & x_1 + x_2 \\ \text{s.a.} \quad & R1) \quad x_1 + 3x_2 \leq 9 \\ & R2) \quad 2x_1 + x_2 \leq 4 \\ & R3) \quad x_1 + x_2 \geq 1 \\ & x_1 \geq 0, x_2 \geq 0 \end{aligned}$$

a) Resuelva el problema de forma gráfica. Considere indicar todos los siguientes elementos en la gráfica y en su descripción:

1. Grafique el dominio del problema indicando cada restricción (incluidas las restricciones de no negatividad).
2. Coloque sobre el gráfico nombres VF_j a todos los vértices j del problema y tabule sus coordenadas en el formato (x_1, x_2) (para bonus, la tabla debe ser en latex). Indique en la tabla qué restricciones se intersectan formando ese vértice.
3. Haga exactamente lo mismo que el punto 2 con todas las n intersecciones de restricciones que no pertenecen al dominio y llámelas VI_n. Agregue en la tabla qué restricción del problema no están cumpliendo esos puntos y qué los torna infactibles.
4. Grafique desde el origen el vector gradiente de la función objetivo.
5. Indique la solución y el valor óptimo del problema.
6. Indique qué restricción(es) se activa(n) en la solución óptima. Indique cuál(es) no se activa(n).

b) Escriba el problema en su forma estándar. Explicite cuáles son variables de holgura y cuáles son variables de exceso.

c) Resuelva la Fase 1 de simplex aplicado a este problema. Durante cada iteración, las variables x_1 y x_2 toman determinados valores. Estos se deben mostrar como un punto en un nuevo gráfico del problema, similar al gráfico del inciso a)1. Debe construirse un nuevo gráfico para cada iteración simplex.

d) En base al resultado de c) resuelva la Fase 2 de simplex aplicado a este problema, partiendo desde la base factible encontrada en c). Continúe el procedimiento de graficar los valores de las variables x_1 y x_2.

e) Por alguna razón, se generó cierta variación δ en el recurso (4) de la restricción R2). Indique mediante un argumento gráfico entre qué valores puede estar δ para mantener la configuración de base óptima actual sin que haya degenerancia. Indique los valores que debe tomar delta para que haya degenerancia en la base óptima actual.

2. Sea $P = \{x \in \mathbb{R}^n : \ Ax = b, \ x \geq 0\}$ un poliedro en forma estándar no vacío definido por m restricciones $a_i^\top x = b_i$ para $i = 1, \ldots, m$ tales que la matriz A tiene filas linealmente dependientes.
   a) Explique por qué, sin pérdida de generalidad, podemos suponer que existen escalares $\lambda_1, \ldots, \lambda_{m-1} \in \mathbb{R}$ tales que $\sum_{i=1}^{m-1} \lambda_i a_i = a_m$.
   b) Explique por qué podemos remover la restricción $a_m^\top x = b_m$ sin que P se vea alterado.
   c) ¿Cómo se relaciona lo anterior con el supuesto de filas linealmente independientes en el método simplex?

Solution: a)

[3 pts] Asignar 0.5 puntos por cada item cumplido. No asignar puntajes intermedios por item.

La tabla para los vértices factibles

Nombre vértice factible	Coordenadas	Restricciones activas
VF1	(1,0)	R3 y NV x_2
VF2	(0,1)	R3 y NV x_1
VF3	(2,0)	R2 y NV x_2
VF4	(0,3)	R1 y NV x_1
VF5	$(\frac{3}{5}, \frac{14}{5})$	R1 y R2

La tabla para los vertices infactibles:

Nombre intersección	Coordenadas	R activas	R que genera infactibilidad
VI1	(0,0)	NV x_1 y NV x_2	R3
VI2	(9,0)	R1 y NV x_2	R2
VI3	(0,4)	R2 y NV x_1	R1
VI4	(-3,4)	R3 y R1	R2
VI5	(3,-2)	R3 y R2	R1

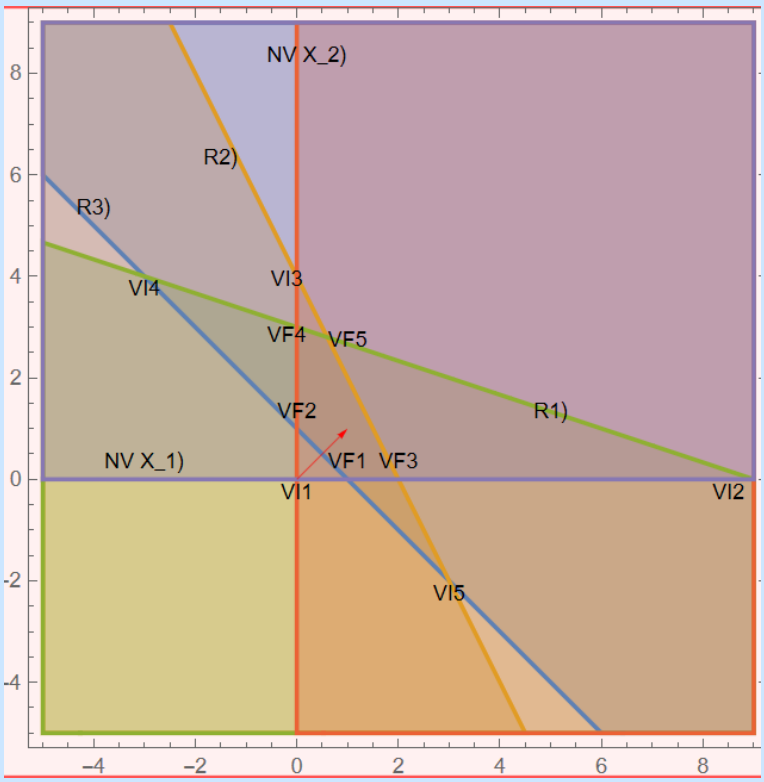


Figura 1: Grafico a)

La solución óptima del problema es $(x_1, x_2) = (\frac{3}{5}, \frac{14}{5})$ y el valor óptimo es $\frac{17}{5}$ unidades. Las restricciones activas en el óptimo son R1 y R2. Por otro lado, las restricciones no activas son R3 y la NV de x_1 y la de x_2.

b) [1 pto] Debe estar perfectamente escrito, sin ningún error.

El problema en su forma estándar es:

$$
\begin{aligned}
\text{P) min} \quad & -x_1 - x_2 \\
\text{s.a.} \quad R1) \quad & x_1 + 3x_2 + h_1 = 9 \\
R2) \quad & 2x_1 + x_2 + h_2 = 4 \\
R3) \quad & x_1 + x_2 - e_1 = 1 \\
& x_1, x_2, h_1, h_2, e_1 \geq 0
\end{aligned}
$$

Las variables de holgura son: h_1, h_2
La variable de exceso es: e_1

c)

[3ptos] Por la correcta respuesta del inciso c) de esta pregunta.
Debe incluir los elementos calculados en la iteración, tales como la base a usar durante la iteración, $\bar{b}$

y los costos reducidos. Debe observarse explícitamnte el criterio de salida y de entrada. Debe incluir el gráfico pedido. Al finalizar cada iteración debe concluir con la base que servitá para la próxima iteración.
Si es que llega a argumentar claramente que no es necesaria la iteración 2, no hay probelma y se debe asignar el puntaje completo.

Procedemos a aplicar simplex Fase I.

Existe un problema y es que no hay una base factible trivial para comenzar a iterar Fase II. Por eso se debe añadir una variable auxiliar que se intenta minimizar.

$$\begin{aligned} \text{P) min} \quad & t \\ \text{s.a.} \quad & R1) \quad x_1 + 3x_2 + h_1 = 9 \\ & R2) \quad 2x_1 + x_2 + h_2 = 4 \\ & R3) \quad x_1 + x_2 - e_1 + t = 1 \\ & x_1, x_2, h_1, h_2, e_1, t \geq 0 \end{aligned}$$

$$A = \begin{bmatrix} 1 & 3 & 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & -1 & 1 \end{bmatrix}, \; c = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \; b = \begin{bmatrix} 9 \\ 4 \\ 1 \end{bmatrix}$$

Usaremos la siguiente notación: B los elementos de la base, N los elementos no básicos, A_B columnas de los elementos de la base, A_N columnas de los elementos no básicos y $\bar{b} = A_B^{-1} b$.

Iteración 1 Fase I

Los elementos de la base y los no básicos son:

$$B = \{h_1, h_2, t\} \;\; N = \{x_1, x_2, e_1\}$$

$$A_B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \;\; A_N = \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & -1 \end{bmatrix}$$

$$\bar{b} = \begin{bmatrix} 9 \\ 4 \\ 1 \end{bmatrix}$$

Luego, los costos reducidos son:

$$c_R^T = c_N^T - c_B^T A_B^{-1} A_N = (-1, \; -1, \; 1,)$$

Entra x_1 a la base. Aplicamos el criterio de salida:
$\frac{\bar{b}}{R_{x_1}} = min\left\{\frac{9}{1}, \frac{4}{2}, \frac{1}{1}\right\} = 1$
Sale t de la base por el criterio de salida. Con esto, la nueva base sería B= (x_1, h_1, h_2).

Durante la iteración, los valores que se calcularon fueron $(x_1, x_2) = (0,0)$ y se muestran con un punto verde en el siguiente gráfico.

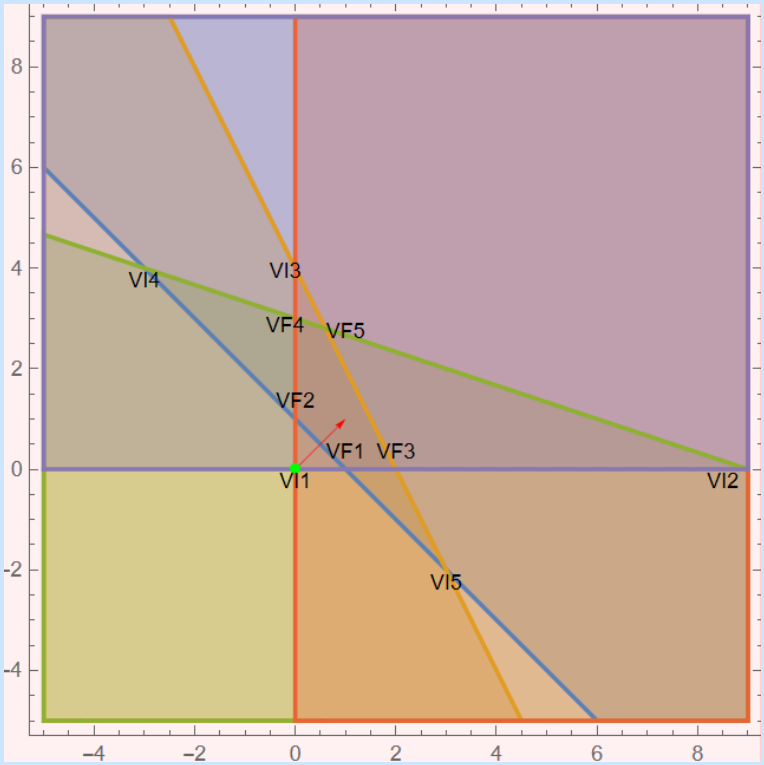


Figura 2: Gráfico iteración 1 Fase I

Opción 1: Mencionar que ya se llegó a una base óptima puesto que t no está en la base (y por tanto es cero) y la función objetivo está acotada inferirmente por cero.

Opción 2: Demostrar que esa base es óptima de Fase I mediante una segunda iteración simplex.

Iteración 2 Fase I

(Es opcional esta iteración)

Los elementos de la base y los no básicos son:

$$B = \{x_1, h_1, h_2\} \quad N = \{x_2, e_1, t\}$$

$$A_B = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad A_N = \begin{bmatrix} 3 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & -1 & 1 \end{bmatrix}$$

$$\bar{b} = \begin{bmatrix} 1 \\ 8 \\ 2 \end{bmatrix}$$

Luego, los costos reducidos son:

$$c_R^T = c_N^T - c_B^T A_B^{-1} A_N = (0,\ 0,\ 1)$$

Como todos los costos reducidos son no negativos, se puede concluir que se llegó al óptimo de fase I y que una base factible para el problema es B= (x_1, h_1, h_2).

Durante la iteración, los valores que se calcularon fueron (x_1, x_2) = (1,0) y se muestran con un punto verde en el siguiente gráfico.

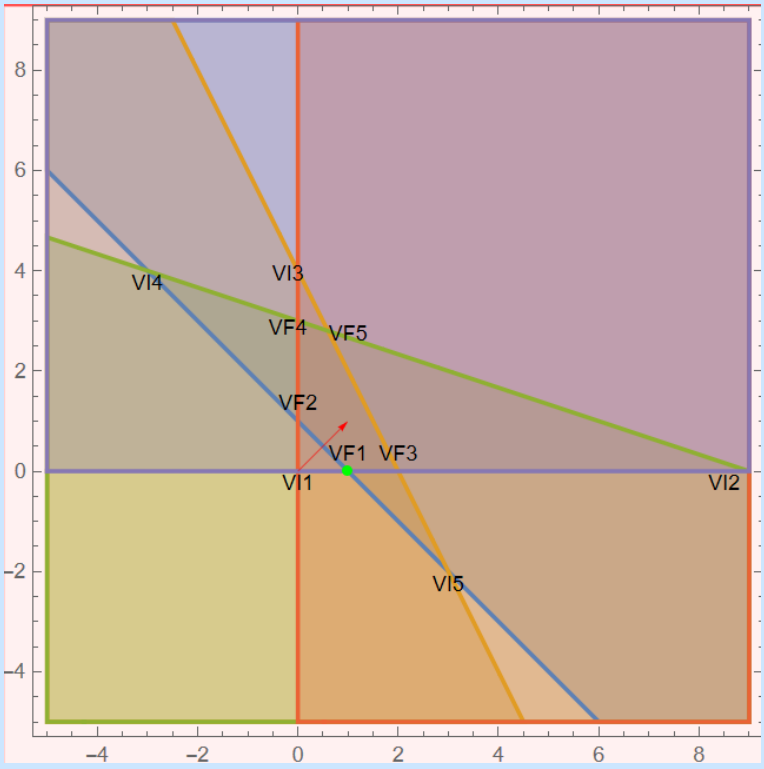


Figura 3: Gráfico iteración 2 Fase I

d)

[4 ptos] Por responder la totalidad de la pregunta.

[3 pts] por iteración y [1 pto] por conclusión del valor óptimo y solución óptima del problema original $= \frac{17}{5}$.

Debe incluir los elementos calculados en la iteración, tales como la base a usar durante la iteración, $\bar{b}$ y los costos reducidos. Debe observarse explícitamnte el criterio de salida y de entrada. Debe incluir el gráfico pedido. Al finalizar cada iteración debe concluir con la base que servirá para la próxima iteración.

Se procede a realizar las iteraciones simplex pedidas para fase II. Se parte desde la base encontrada en el inciso anterior B= (x_1, h_1, h_2). Además, se utiliza el problema en forma estándar:

$$
\begin{aligned}
\text{P) min} \quad & -x_1 - x_2 \\
\text{s.a.} \quad R1) \quad & x_1 + 3x_2 + h_1 = 9 \\
R2) \quad & 2x_1 + x_2 + h_2 = 4 \\
R3) \quad & x_1 + x_2 - e_1 = 1 \\
& x_1, x_2, h_1, h_2, e_1 \geq 0
\end{aligned}
$$

$$
A = \begin{bmatrix} 1 & 3 & 1 & 0 & 0 \\ 2 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & -1 \end{bmatrix}, \; c = \begin{bmatrix} -1 \\ -1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \; b = \begin{bmatrix} 9 \\ 4 \\ 1 \end{bmatrix}
$$

Iteración 1 Fase II

Los elementos de la base y los no básicos son:

$$B = \{x_1, h_1, h_2\} \;\; N = \{x_2, e_1\}$$

$$
A_B = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \;\; A_N = \begin{bmatrix} 3 & 0 \\ 1 & 0 \\ 1 & -1 \end{bmatrix}
$$

$$
\bar{b} = \begin{bmatrix} 1 \\ 8 \\ 2 \end{bmatrix}
$$

Luego, los costos reducidos son:

$$c_R^T = c_N^T - c_B^T A_B^{-1} A_N = (0, \; -1)$$

Entra e_1 a la base porque su costo reducido es el único negativo. Aplicamos el criterio de salida:

$\frac{\bar{b}}{R_{e_1}} = min\left\{\frac{1}{-1}, \frac{8}{1}, \frac{2}{2}\right\} = min\left\{\bullet, \frac{8}{1}, \frac{2}{2}\right\} = 1$

Sale h_2 de la base por el criterio de salida. Con esto, la nueva base sería B= (x_1, h_1, e_2).

Durante la iteración, los valores que se calcularon fueron (x_1, x_2) = (1,0) y se muestran con un punto verde en el siguiente gráfico.

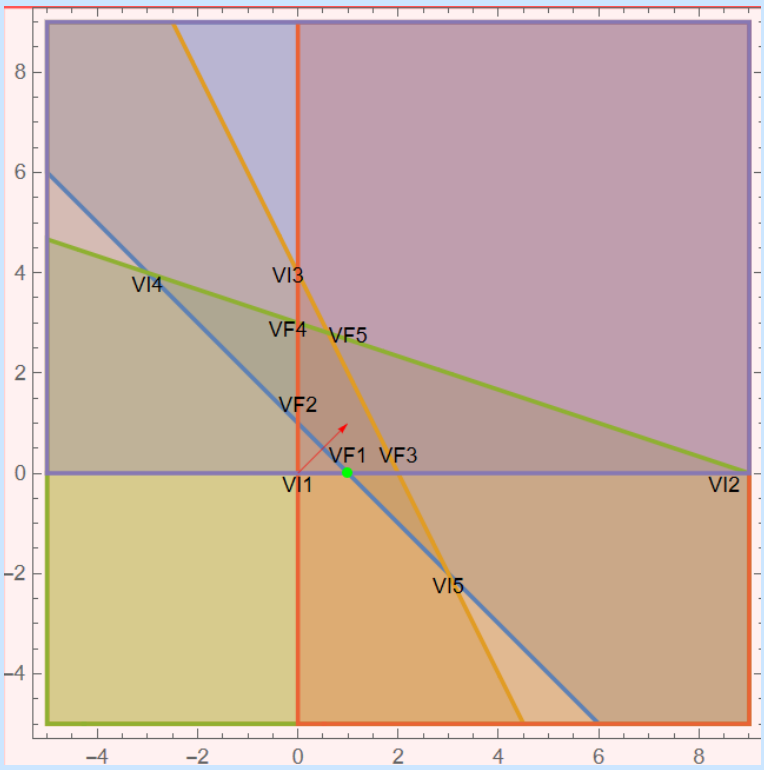


Figura 4: Gráfico iteración 1 Fase II

Iteración 2 Fase II

Los elementos de la base y los no básicos son:

$$B = \{x_1, h_1, e_2\} \quad N = \{x_2, h_2\}$$

$$A_B = \begin{bmatrix} 1 & 1 & 0 \\ 2 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 3 & 0 \\ 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\bar{b} = \begin{bmatrix} 2 \\ 7 \\ 1 \end{bmatrix}$$

Luego, los costos reducidos son:

$$c_R^T = c_N^T - c_B^T A_B^{-1} A_N = (\frac{-1}{2}, \ \frac{1}{2})$$

Entra x_2 a la base porque su costo reducido es el único negativo. Aplicamos el criterio de salida:
$\frac{\bar{b}}{R_{e_1}} = min\left\{\frac{2}{\frac{1}{2}}, \frac{7}{\frac{5}{2}}, \frac{1}{\frac{-1}{2}}\right\} = min\left\{4, \frac{14}{5}, \bullet\right\} = 1$
Sale h_1 de la base por el criterio de salida. Con esto, la nueva base sería B= (x_1, x_2, e_2).

Durante la iteración, los valores que se calcularon fueron $(x_1, x_2) = (2,0)$ y se muestran con un punto verde en el siguiente gráfico.

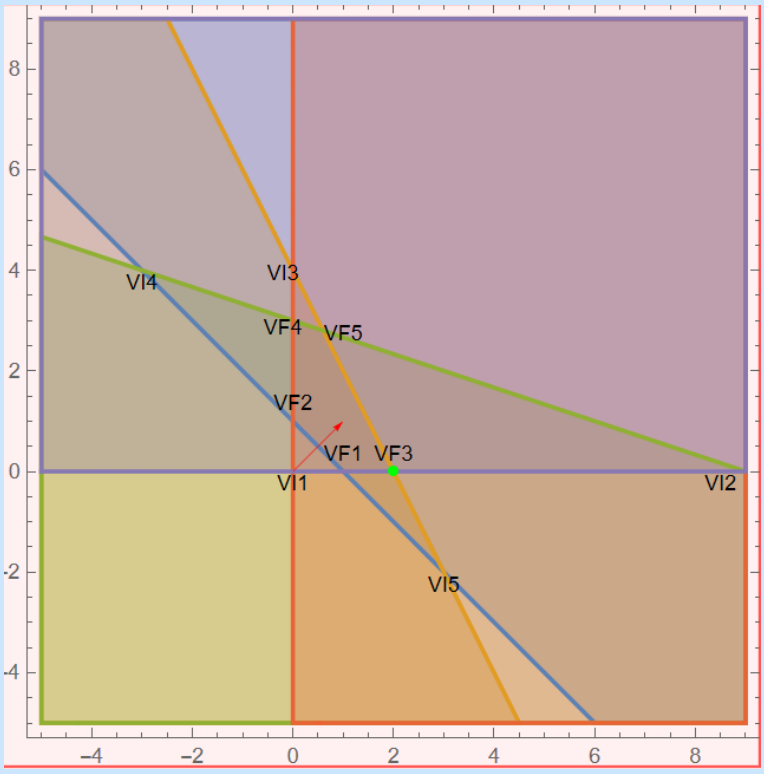


Figura 5: Gráfico iteración 2 Fase II

Iteración 3 Fase II

Los elementos de la base y los no básicos son:

$$B = \{x_1, x_2, e_2\} \quad N = \{h_1, h_2\}$$

$$A_B = \begin{bmatrix} 1 & 3 & 0 \\ 2 & 1 & 0 \\ 1 & 1 & -1 \end{bmatrix}, \quad A_N = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}$$

$$\bar{b} = \begin{bmatrix} \frac{3}{5} \\ \frac{14}{5} \\ \frac{12}{5} \end{bmatrix}$$

Luego, los costos reducidos son:

$$c_R^T = c_N^T - c_B^T A_B^{-1} A_N = (\frac{1}{5}, \ \frac{2}{5})$$

Se ha llegado al óptimo del probblema ya que todos los costos reducidos son no negativos.

Durante la iteración, los valores óptimos que se calcularon fueron $(x_1, x_2) = (\frac{3}{5}, \frac{14}{5})$ y se muestran con un punto verde en el siguiente gráfico. Además, el valor óptimo se calcula evaluando la solución óptima en la función objetivo del problema original, resultando $\frac{17}{5}$.

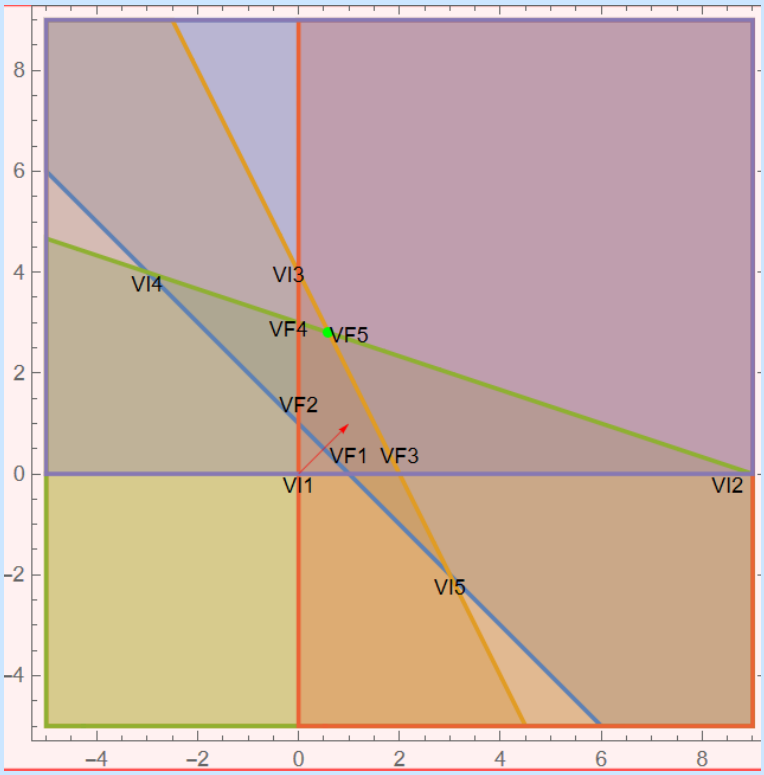


Figura 6: Gráfico iteración 3 Fase II

e)

[3 pts] En total.

[1pto] por mencionar los valores de δ para los cuales hay degenerancia. [2pts] por mencionar el rango de valores de δ para los cuales no hay degenerancia. No es necesario graficar, simplemente un argumento gráfico de traslación de restricción.

Análisis gráfico:

Es claro que al cambiar el recurso de la segunda restricción en δ unidades la restricción R2 se traslada. Si se observa el gráfico del inciso a), se puede observar que en esas traslaciones el vértice VF5 se deslizaría a través de la restrición R1 acercándose al VF4 o al VI2 cuando se cambia el recurso de R2. Lo anterior nos puede dar el razonamiento de las situaciones que deben pasar para que el vértice VF5 tenga una base degenerada. Si la R2 se traslada hasta pasar por el vértice VF4, se tendría una situación de degenerancia en la base óptima actual. Para eso el valor de δ = -1. Por otro lado, si R2 se traslada hasta pasar por VI2 se tendría denuevo una situación de degenerancia. Para que eso pase, el valor de δ

$= 14$. Se concluye por análisis gráfico que un intervalo para δ sin que se produzca degenerancia es (-1,14).

2) Son [6pts] en total para esta pregunta.
a) [2pts]

Se puede suponer por la definición de dependencia lineal de las filas de A. En caso de que la m-ésima fila no sea posible de construir con ponderaciones lineales de las otras, entonces se puede reordenar la matriz. Sin pérdida de generalidad se pueden reordenar las filas tal que en la m-ésima posición cumplir con el supuestoo del enunciado.

b) [2pts]

Porque es una restriccion construible a partir de las otras con ponderaciones λ_i que cumplan $\sum_{i=1}^{m-1} \lambda_i a_i = a_m$ y $\sum_{i=1}^{m-1} \lambda_i b_i = b_m$. notar que si es que no se cumpliera la segunda ecuación, el poliedro sería vacío, pero como eso contradice el enunciado, se puede asumir que no ocurre. Luego la última restricción no impone ninguna condición nueva para P que las otras restricciones no hayan impuesto, y por lo tanto es removible sin consecuencias.

c) [2pts]

Están estrechamente relacionadas. Simplex impone la condición de filas linealmente independientes para poder introducir el uso de matrices base invertibles. Para utilizar esas matrices base invertibles en el algoritmo, no puede ocurrir que sus filas sean linealmente dependientes. Luego, es necesario remover cuantas restricciones sean necesarias hasta que sólo queden restricciones linealmente independientes.

Pregunta 2 **[20 Puntos]**

Geometría Poliedral

1. Demuestre que un poliedro $P = \{x \in \mathbb{R}^n : Ax \leq b\}$ es un conjunto convexo. Considere $A \in \mathbb{R}^{m \times n}$ y $b \in \mathbb{R}^m$.

2. Demuestre que un conjunto $P = \left\{ \begin{pmatrix} x \\ z \end{pmatrix} : \quad x \in \mathbb{R}^n, z \in \{0,1\}, A \begin{pmatrix} x \\ z \end{pmatrix} \leq b \right\}$ no es necesariamente un conjunto convexo. Considere $A \in \mathbb{R}^{m \times (n+1)}$ y $b \in \mathbb{R}^m$.

3. Sea P) un problema de programación lineal dado por $P) : \{\text{máx}\, c^T x \quad s.a. \quad x \in \mathbb{R}^n, Ax \leq b\}$. Considere $A \in \mathbb{R}^{m \times n}$, $c \in \mathbb{R}^n$ y $b \in \mathbb{R}^m$. Demuestre que si existen tres soluciones óptimas diferentes para P), entonces existen infinitas soluciones óptimas.

4. Sea $P = \{x \in \mathbb{R}^n : \ Ax \geq b\} = \{x \in \mathbb{R}^n : \ a_i^\top x \geq b_i \ \forall i = 1, \ldots, m\}$ un poliedro. Considere $A \in \mathbb{R}^{m \times n}$ y $b \in \mathbb{R}^m$. Se define una cara como un subconjunto de P obtenido al hacer activas algunas restricciones. Formalmente, si C es una cara de P, entonces existe $I \subseteq \{1, \ldots, m\}$ tal que $C = \{x \in P : \ a_i^\top x = b_i \ \forall i \in I\}$. La cara vacía es trivial y se define como cara para todo poliedro por convención, al igual que lo es el poliedro completo P (que corresponde a la cara definida cuando I es el conjunto vacío).
   a) Encuentre y explique una expresión para el número total de caras de $P = \{x \in \mathbb{R}^n : \ 0 \leq x_i \leq 1 \ \forall i = 1, \ldots, n\}$ en función de n. Le puede servir identificar que para $n = 2$ el total de caras es 10.

   b) Sea $P = \{x \in \mathbb{R}^n : \ x_i \geq 0 \quad i = 1, \ldots, n\}$. Demuestre que el total de caras de P es $2^n + 1$.

5. Sea $P = \{x \in \mathbb{R}^n \ Ax \geq b\} = \{x \in \mathbb{R}^n : \ a_i^\top x \geq b_i \ \forall i = 1, \ldots, m\}$. Dado $\hat{x} \in P$, sea $I \subseteq \{1, \ldots, m\}$ el conjunto de restricciones activas en $\hat{x}$ y definamos $c = \sum_{i \in I} a_i$.
   a) Demuestre que $\hat{x}$ es solución óptima de mín$\{c^\top x : \ Ax \geq b\}$.
   b) Suponga que $\hat{x}$ es vértice de P. Demuestre que $\hat{x}$ es la única solución óptima de mín$\{c^\top x : Ax \geq b\}$.

Solution:
1) [2 pts]

Sea x_1, $x_2 \in P$. Sea $x_3 = \lambda \cdot x_1 + (1 - \lambda) \cdot x_2, \lambda \in [0,1]$ la combinación convexa de esos dos puntos. Para demostrar que P es convexo, es necesario demostrar que x_3 pertenece a P para todo x_1 y x_2 partenecientes al conjunto.

$$Ax_3 = A \cdot (\lambda \cdot x_1 + (1 - \lambda) \cdot x_2) = \lambda \cdot Ax_1 + (1 - \lambda) \cdot Ax_2 \leq \lambda \cdot b + (1 - \lambda) \cdot b = b$$

Por lo anterior, queda demostrado por definición que P es un conjunto convexo.

2) [3 pts]

No es necesariamente convexo, puesto que el sistema $A \begin{pmatrix} x \\ z \end{pmatrix} \leq b$ podría llegar a permitir, con sus restricciones, que la variable z tome tanto el valor 0 como el 1 en el dominio. En ese caso, basta tomar

$y_1 = \begin{pmatrix} x_1 \\ 0 \end{pmatrix} \in P$ e $y_2 = \begin{pmatrix} x_2 \\ 1 \end{pmatrix} \in P$, observar que la combinación convexa de esos elementos del dominio no pertenece al dominio por violar la naturaleza binaria de la última variable.

3) [3 pts]

Para este probleam se puede identificar que si existen 3 soluciones óptimas, entonces en particular también existen 2. Sean x_1 y x_2 soluciones óptimas. Sea $x_3 = \lambda \cdot x_1 + (1-\lambda) \cdot x_2, \lambda \in [0,1]$
la combinación convexa que, como se demostró en a), pertence al dominio. Si v^* es el valor, entonces $v^* = c^T x_1 = c^T x_2$. Por último el desarrollo $c^T x_3 = c^T(\lambda \cdot x_1 + (1-\lambda) \cdot x_2) = v^*$ demuestra que x_3 es solución óptima para todo $\lambda \in [0,1]$ por lo que hay infinitas soluciones óptimas para P).

4)

a)

[3pts] Asignar puntaje según logro de la solución. No necesariamente será igual a la pauta.

Alternativa 1

Primero que nada es conveniente comprender la lógica del problema para un caso de n = 2. Está la cara trivial vacía, cara trivial de todo el cuadrado, 4 caras vértices y cuatro aristas (dim = 1).
Las caras que son vértices se obtienen activando 2 restricciones que se intersecten. Se obtienen tomando 2 de 2 variables y multiplicando por 2^2 opciones de diferentes valores para esas 2 variables (dependiendo de qué restricción se escoja activar). De este proceso resultan las caras: $(0,0), (0,1), (1,0), (1,1)$
Las caras que son aristas se obtienen activando 1 sola restricción. Se obtienen tomando 1 de 2 variables y multiplicando por 2^1 opciones de diferentes valores para esa variable, quedando la otra libre en el dominio. De este proceso resultan las caras: $(0, x_2), (x_1, 1), (1, x_2), (x_1, 1)$
La cara trivial de todo el poliedro se obtiene no forzando a estar activa ninguna restricción. Se obtiene tomando 0 de 2 variables y multiplicando por 2^0 opciones de diferentes valores para esa variable. De este proceso resulta la cara: (x_1, x_2)
Considerando además la cara vacía de dimensión -1 se encuentran las 10 caras de ese poliedro.
Cabe destacar que el procedimeinto para encontrar la cantidad de caras en las que k variables estuvieran fijas (se activara su restricción inferior o superior) en un dominio de esta forma se puede generalizar. Se puede generalizar a n variables. La expresión sería:

$$\binom{n}{k} \cdot 2^k \tag{1}$$

Ahora, considerando que la única cara de dimensión k=n es una de las dos caras triviales (la que es el poliedro completo), se puede hacer una suma para encontrar todas las caras (y agregarle de manera extra la cara trivial nula). De esta manera, se generaliza una expresión para encontrar caras en un poliedro de las características del enunciado.

$$\textbf{Cantidad de caras} : 1 + \sum_{k=0}^{k=n} \binom{n}{k} \cdot 2^k \tag{2}$$

Alternativa 2

Para $P = \{x \in \mathbb{R}^n : 0 \leq x_i \leq 1 \; \forall i = 1, \ldots, n\}$, que está definido como un cuadrado en $\mathbb{R}^2$, un cubo en $\mathbb{R}^3$, y sus análogos en mayores dimensiones.

Se ordenarán las restricciones del problema en $\mathbb{R}^n$ de la siguiente forma para poder definir correctamente un conjunto I para cada cara.

$$\begin{pmatrix} 0 \leq x_1 \; (1) \\ \vdots \\ 0 \leq x_n \; (n) \\ x_1 \leq 1 \; (n+1) \\ \vdots \\ x_n \leq 1 \; (2n) \end{pmatrix}$$

Encontrar la cantidad de caras de un poliedro se puede hacer contando cúantos conjuntos $I \subseteq \{1, \ldots, m\}$ se pueden formar, tal que C_I es no vacío, y sumarle el conjunto vacío. Para conocer esa cantidad se genera un vector G con información equivalente a I.

Cada variable $x_i \in x$ puede ser parte de una cara de 3 formas, 1: $\{x_i = 0\}$, 2: $\{x_i = 1\}$ y 3: $\{0 \leq x_i \leq 1\}$ (siendo así una variable libre en la cara). El tercer tipo de cara mencionado anteriormente, asociado a un conjunto I vacío, tiene todas las varibles libres (dentro del poliedro).

Para cada conjunto I, se genera un vector $G \in \mathbb{R}^n$ con filas $g_i \in \{1, 2, 3\}$ que describe el tipo de comportamiento de x_i en C_I, según las formas de pertenencia a una cara descritas en el párrafo anterior. Cada cara, por consiguiente queda descrita por un vector G, y cada vector G describe una cara real del poliedro.

Cabe destacar que en $\mathbb{R}^n$, hay 3^n combinaciones de vectores G, por lo que, sumando el conjunto vacío, en el poliedro P en $\mathbb{R}^n$ hay $3^n + 1$ caras.

b) [3pts] Asignar puntaje según logro de la solución. No necesariamente será igual a la pauta.

Alternativa 1

La idea de esta demostración es llegar a la conclusión: para construir una cara en que estén activas k restricciones de un total de n se deben simplemente seleccionar por medio de una combinatoria. El resultado sería:

$$\sum_{k=0}^{k=n} \binom{n}{n-k} = 2^n \tag{3}$$

Como se debe agregar la cara trivial nula el resultado final es $2^n + 1$ caras en total.

$$\textbf{Cantidad de caras} : 1 + \sum_{k=0}^{k=n} \binom{n}{n-k} = 2^n + 1 \tag{4}$$

Alternativa 2

El problema $P = \{x \in \mathbb{R}^n : x_i \geq 0 \text{ para } i = 1, \ldots, n\}$ es similar al anterior, con un razonamiento parecido, podemos generar el vector G con elemengtos $g_i \in \{1, 2\}$, por las 2 posibilidades $(x_i = 0,\ x_i \geq 0)$ que tiene cada variable x_i en una cara. Por el mismo razonamiento de antes, las combinaciones de G son 2^n , y caras (sumando la cara vacía) $2^n + 1$.

5)

a) [3pts] Asignar puntaje según logro de la solución. No necesariamente será igual a la pauta.

Alternativa 1

Por cómo están construidas las restricciones del problema, $\hat{x}$ es óptimo del problema si el gradiente de la función objetivo en $\hat{x}$ es igual a una combinación cónica de los gradientes de las restricciones activas en $\hat{x}$. Las restricciones activas son las del conjunto I y el gradiente de cada una es su respectivo a_i. Por otro lado, el gradiente de la función objetivo es:

$$\nabla f = \sum_{i \in I} a_i = c$$

Por lo que se puede afirmar que sí existen ponderadores $\lambda_i \geq 0$ que en una combinación cónica de los gradientes de las restricciones activas son capaces de construir el gradiente de la función objetivo. Es claro ver que existen y que son $\lambda_i = 1 \quad \forall i \in I$. Por lo anterior $\hat{x}$ es óptimo del problema.

Alternativa 2

Todo $x \in P$ satisface $a_i^\top x \geq b_i$ para $i \in I$, por lo que $c^\top x \geq \sum_{i \in I} b_i$. Por otro lado, $\hat{x}$ satisface $a_i^\top \hat{x} = b_i$ para $i \in I$, por lo que $c^\top \hat{x} = \sum_{i \in I} b_i$. Luego, $c^\top x \geq c^\top \hat{x}$ para todo $x \in P$, por lo que $\hat{x}$ es solución óptima.

b) [3pts] Asignar puntaje según logro de la solución. No necesariamente será igual a la pauta.

Alternativa 1

Si $\hat{x}$ es un vértice de P, entonces en él hay al menos n restricciones activas linealmente independientes. Esto implica que el conjunto I contiene al menos n restricciones linealmente independientes. En la parte anterior se demostró que $\hat{x}$ es óptimo. Recordemos que el gradiente de la función objetivo c se obtiene sumando los gradientes de las restricciones activas en I, donde cada restricción se pondera con 1.

Ahora, supongamos que existe otra solución óptima x_1. Esta solución activa menos de n restricciones linealmente independientes de I, de lo contrario sería exactamente igual a $\hat{x}$. Sin embargo, dado que activa menos de n restricciones linealmente independientes, la combinación convexa de los gradientes de sus restricciones nunca podrá sumar c. Esto contradice el hecho de que I contiene al menos n restricciones linealmente independientes que, ponderadas todas por 1, generan c.

En resumen, la existencia de otra solución óptima aparte de $\hat{x}$ resultaría en una contradicción, ya que la combinación convexa de sus gradientes de restricción no podría igualar el gradiente de la función objetivo c. Por lo tanto, concluimos que $\hat{x}$ es la única solución óptima.

Alternativa 2

De la parte anterior, tenemos que toda solución óptima necesariamente satisface $a_i^\top x = b_i$ para $i \in I$. En particular, como $\hat{x}$ es vértice, existen n de ellas linealmente independientes. Sea $A'x = b'$ el subsistema de $Ax = b$ dado por dichas restricciones. Tenemos que A' es invertible, por lo que existe solución única. Como $\hat{x}$ es solución, no puede haber otra.

Pregunta 3 **[20 Puntos]**

1. Implementación computacional

1.1. Descripción del problema

Considere el problema de la pregunta 1 de la I1. Lo que se pide a grandes rasgos es la implementación computacional, usando la interfaz Python-Gurobi, del modelo del problema. Debe utilizar el modelo de la pauta que está subido en Canvas. A continuación se indicarán los elementos técnicos a considerar en esta pregunta.

1.2. Base de datos:

La implementación computacional de este problema debe poder recibir cualquier instancia de parámetros, independiente de su tamaño y valores de los datos. No puede asumir valores ni cardinalidades de ningún parámetro o conjunto. Tanto los cuadrantes como los tipos de semillas serán trabajados de manera abstracta, sólo considerándolos por sus índices. Toda la información de las instancias de datos vendrá en 7 archivos con extensión .csv. Una instancia de estos archivos se puede descargar desde Canvas, junto al enunciado de la tarea. Es fuertemente recomendado testear la robustez del código con estas instancias, ya que para la evaluación se utilizarán unos de idéntico formato pero con diferentes valores y dimensiones. Cabe aclarar que los tamaños de los archivos csv indicarán los tamaños de los conjuntos K, J y T de la pauta. Puede asumir que todos los valores dentro de los archivos son valores enteros. Deben saber que en la pauta todos esos conjuntos de índices parten en 1 y terminan en la cardinalidad del conjunto. A continuación se detalla el formato de cada uno de estos archivos.

1.2.1. capacidad_por_saco.csv

Este archivo csv tiene solo 1 columna y $|J|$ filas. Contiene la información del parámetro α_j de cada semilla $j \in J$. En la primera fila, está el valor de α_1, en la segunda está el valor de α_2, y así sucesivamente.

1.2.2. costo_saco.csv

Este archivo csv tiene una matriz de $|J|$ filas y $|T|$ columnas. Contiene la información del parámetro c_{jt} de cada semilla $j \in J$ para cada mes $t \in T$. En la primera fila, en la primera columna está el valor de $c_{1,1}$ y están ordenados de manera creciente los datos.

1.2.3. tiempo_demora.csv

Este archivo csv tiene solo 1 columna y $|J|$ filas. Contiene la información del parámetro θ_j de cada semilla $j \in J$. En la primera fila, está el valor de θ_1 y están ordenados de manera creciente.

1.2.4. kilos_fruta.csv

Este archivo csv tiene solo 1 columna y $|J|$ filas. Contiene la información del parámetro λ_j de cada semilla $j \in J$. En la primera fila, está el valor de λ_1 y están ordenados de manera creciente.

1.2.5. precio_venta.csv

Este archivo csv tiene una matriz de $|J|$ filas y $|T|$ columnas. Contiene la información del parámetro β_{jt} de cada semilla $j \in J$ para cada mes $t \in T$. En la primera fila, en la primera columna está el valor de $\beta_{1,1}$ y están ordenados de manera creciente los datos.

1.2.6. capital_inicial.csv

Este archivo csv tiene solo 1 columna y 1 fila. Contiene la información del parámetro γ. Es sólo un valor.

1.2.7. cantidad_cuadrantes.csv

Este archivo cvs tiene solo 1 columna y 1 fila. Contiene la información de la cantidad de cuadrantes. Es sólo un valor.

1.3. Código main.py

Se debe elaborar y entregar, junto con el desarrollo de la tarea, un código llamado main.py. Ese código debe cumplir una serie de requisitos:

1) Que al ser ejecutado en una carpeta que contenga los 7 archivos de la base de datos, logre abrir los archivos y extraer su información. Puede asumir que los archivos seguirán el formato mencionado de manera exacta.
2) Se debe resolver el modelo utilizando la librería Gurobi. Cada parte del código debe estar comentada de manera brevísima y clara. Además, debe asumir que la solución óptima existe.
3) El código debe imprimir en consola de manera clara el valor óptimo y su unidad de medida. Además, debe imprimir en consola de manera clara la cantidad de veces que se plantó (no importa qué semilla) en cada terreno $k \in K$ en el marco de T meses.
4) En su código debe elaborar una tabla que permita visualizar en consola la información de la variable X. En cada fila debe ir cada terreno y en las columnas los meses. El interior de la tabla se debe rellenar con un cero cuando no se plante nada, y con un valor entero (representando el tipo de semilla $j \in J$) cuando se plante. Debe ser una tabla perfectamente entendible. Tiene permitido usar cualquier librería externa.

Solution: Si al correr el código, se muestra exactamente lo esperado, asignar [20 pts] a la pregunta. Si no ocurre lo anterior, entonces proceder a la asiganción de puntaje:

La asignación de puntaje en esta pregunta se separa en 2 ramas: elaboración del código [6 pts] y correcto funcionamiento del código [14 pts].

Elaboración del código [6pts]

Se otorga puntaje si están completas las siguientes estructuras en el código main.py entregado.

1) Apertura de todos los archivos.csv [1pto]
2) Definición de todas las variables del problema. [1 pto]
3) Definición de todas las restricciones (todas presentes) [2 pts]
4) Estructura para imprimir en consola lo pedido en el inciso 3 de la pregunta. [1pto]
5) Estructura de código para generar la solución gráfica a modo de calendario. (Puede ser interfaz gráfica, creación de archivo de texto ordenado, creción de csv o excel,.. etc, pero NO imprimir tabla en consola) [1pto]

Funcionamiento del código [14 pts]

Se otorga puntaje si se cumple cada uno de los puntos: 1) El código corre sin arrojar error y se imprimen valor y solución en consola. No es la solución correcta necesariamente. [4pts]
2) El código genera una solución de manera gráfica a modo de calendario (Puede ser interfaz gráfica, creación de archivo de texto ordenado, creción de csv o excel,.. etc, pero NO imprimir tabla en consola). No necesariamente con los valores correctos. [2pts]
3) El código imprimer la solución óptima correcta del problema. Recordar que el orden de los terrenos no es relevante porque son terrenos iguales. [5 pts]
4) El código genera una tabla con las instrucciones correctas para plantar. [3 pts]