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:

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

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 que 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). 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. Concluya la solución y el valor óptimo del problema original.

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?

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 exiten tres soluciones óptimas diferentes para P), entonces existen infinitas soluciones óptimas.

4. Sea $P = \{x \in \mathbb{R}^n : \ Ax \geq b\} = \{x \in \mathbb{R}^n : \ a_i^\top x \geq b_i \ \forall i = 1, \dots, m\}$ 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 $I \subseteq \{1, \dots, m\}$ 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 $P = \{x \in \mathbb{R}^n : \ 0 \leq x_i \leq 1 \ \forall i = 1, \dots, n\}$ en función de n. Le puede servir identificar que para $n = 2$ el total de caras es 10.

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

5. Sea $P = \{x \in \mathbb{R}^n \ Ax \geq b\} = \{x \in \mathbb{R}^n : \ a_i^\top x \geq b_i \ \forall i = 1, \dots, m\}$. Dado $\hat{x} \in P$, sea $I \subseteq \{1, \dots, m\}$ 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 $\text{mín}\{c^\top x : \ Ax \geq b\}$.
   b) Suponga que $\hat{x}$ es vértice de P. Demuestre que $\hat{x}$ es la única solución óptima de $\text{mín}\{c^\top x : Ax \geq b\}$.

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 la información de la variable X a modo de calendario. 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 y que no esté impresa en la consola. Tiene permitido usar cualquier librería externa.