ICS 2121 Métodos de Optimización, Sem 2020-2
Prof. Jorge Vera

Tarea 3.

Instrucciones: Esta tarea debe realizarse en los grupos constituidos en el curso según las instrucciones dadas. La tarea debe ser respondida en forma ordenada y escrita en computador, usando LaTeX u otro procesador de texto. **No se aceptarán tareas manuscritas. De ocurrir, no serán corregidas. Debe ser subida a la plataforma de tareas a más tardar el viernes 30 de octubre de 2020 a las 23:59 horas, como un archivo PDF.** Si usan LaTeX, les daremes 2 décima de bono en la tarea. También deberá subir el código del problema computacional, se le indicará más adelante cómo hacerlo. Durante su trabajo en la tarea deben respetar las normas éticas correspondientes, cualquier colaboración indebida entre los grupos sera sancionada según establecen los reglamentos de la Escuela y Universidad, pudiendo incluso implicar la reprobación del curso. Todos los temas de esta tarea pueden ser evaluados en las interrogaciones.

Los problemas 1, 2 y 3 requieren de desarrollos computacionales.

Pregunta 1 (6 pts):
El siguiente problema se refiere a la planificación de operaciones que debe realizar un aserradero en la industria forestal. Para introducir el contexto, digamos que un aserradero debe procurar abastecimientos de troncos de diversas características, los que son cortados según un cierto patrón de corte (recuerde que los troncos tienen perfil más o menos circular mientras que las tablas son rectangulares). Esto da origen a los distintos productos, que son tablas de distintas medidas. El aserradero debe, entonces, decidir cuántos troncos de distintos tipos comprar y cómo cortarlos, de modo de cumplir con la demanda, y todo esto al mínimo costo. Esto último incluye costos de procesamiento, de inventario de productos finales, y costos fijos de setup cada vez que se procesan troncos de ciertos tipos, además del costo de compra de los troncos. A continuación se presenta un modelo simplificado de la situación.

Useremos los siguientes conjuntos, parámetros y variables:
Conjuntos:

M**:** Conjunto de los productos finales, tipos de tablas.

K**:** Conjunto de tipos de troncos (materia prima).

T**:** Conjunto de periodos de tiempo.

Variables:

s_{kt}**:** Volumen de troncos tipo $k \in K$ a comprar y procesar en el periodo $t \in T$ (m^3).

w_{mt}**:** Volumen de tablas tipo $m \in M$ en inventario en el periodo $t \in T$ (m^3).

r_{mt}**:** Producción de tablas tipo $m \in M$ en el periodo $t \in T$ (m^3).

z_{kt}**:** variable binaria que vale uno si se procesan troncos tipo $k \in K$ en el periodo $t \in T$.

Parámetros:

CT_{kt}**:** Costo de compra de troncos tipo $k \in K$ en $t \in T$ ($\$/m^3$).

CA_{kt}**:** Costo de procesamiento de troncos tipo $k \in K$ en $t \in T$ ($\$/m^3$).

FC_{kt}**:** Costo fijo de setup en que se incurre al procesar troncos tipo $k \in K$ en $t \in T$.

CB_{mt}: Costo de inventario de tablas tipo $m \in M$ en $t \in T$ ($\$/m^3$).

R_{km}: Rendimiento de procesamiento de troncos tipo $k \in K$ como volumen de tablas tipo $m \in M$ por unidad de troncos (m^3 tabloas/ m^3 ltroncos).

PA_t: Capacidad de procesamiento en $t \in T$ (m^3).

PB_t: Capacidad de inventario en $t \in T$ (m^3).

D_{mt}: Demanda estimada por tablas tipo $m \in M$ en $t \in T$ (m^3).

M: Un valor grande que está por arriba de cualquier valor posible para s_{kt}.

El Modelo:

$$
\begin{aligned}
\min z = & \sum_{t \in T} \left[\sum_{k \in K} (CT_{kt} + CA_{kt}) s_{kt} + \sum_{k \in K} FC_{kt} z_{kt} + \sum_{m \in M} CB_{mt} w_{mt} \right] && && (1) \\
s.t. \; & \sum_{k \in K} s_{kt} \le PA_t && \forall t \in T && (2) \\
& \sum_{m \in M} w_{mt} \le PB_t && \forall t \in T && (3) \\
& w_{mt-1} + \sum_{k \in K} R_{km} s_{kt} - w_{mt} = D_{mt} && \forall m \in M, \forall t \in T && (4) \\
& s_{kt} \le M z_{kt} && \forall k \in K, \forall t \in T && (5) \\
& w_{mt}, s_{kt} \ge 0 && \forall m \in M, \forall k \in K, \forall t \in T && (6)
\end{aligned}
$$

Junto con esta tarea, se entrega un código Python llamado `Tactical.py`. Este código implementa, en Python y Gurobi, el modelo anterior para un caso particular de pequeño tamaño. Corra el código y registre el valor óptimo alcanzado para el problema, el tiempo de ejecución y, también haga una tabla ordenada con la solución del problema, indicando mes a mes y para cada tipo de tronco y producto los valores de la solución óptima. Indique también en qué periodos hubo producción (las variables z).

Pregunta 2 (10 pts):
Continuando con el problema de la Pregunta 1, aseguramos que este problema tiene una estructura adecuada para ser abordado mediante descomposición de Benders, considerando las variables s_{kt} y z_{kt} como variables complicantes.

a) (4 pts) Escriba el modelo en forma reordenada de modo tal que se evidencie explícitamente la estructura de bloques que corresponde a la Descomposición de Benders.

b) (6 pts) Escriba los elementos principales de la Descomposición de Benders, de la forma que resultan para este problema, específicamente escriba en forma explícita el problema maestro y el satélite.

Pregunta 3 (10 pts):
Ahora le pedimos que implemente, usando Python y Gurobi, la descomposición de Benders para el problema anterior. Para esto le recomendamos que tome como base el código del ejemplo de Benders que se ha usado en clases, junto con el código Python que codifica el problema, el del archivo `Tactical.py`. De este modo, puede directamente escribir los elementos necesarios para implementar el maestro y el satélite. Ejecute su código de Benders y haga pruebas con número de iteraciones, criterio de parada, etc., lo que usted considere conveniente para tratar de llegar al óptimo que entrega la descomposición. Haga un gráfico de la evolución de las iteraciones y vea si acaso está llegando a la misma solución que obtuvo en la Pregunta 1.

Pregunta 4 (10 pts):

Adjunto a esta tarea está el artículo de J. Janacek, M. Kohami, M. Koniorczyk y P. Marton, "Optimization of Periodic Crew Schedules with application of Column Generation method", publicado en Transportation Research C, vol 83, 2017. Deben leer este artículo y preparar uun resumen de un máximo de 3 planas, destacando el modelo y la forma en que este resulta adecuado al esquema de Generación de Columnas, y por qué. Deben también comentar respecto a la implementación del algoritmo, los problemas maestros y satélites, y cómo resultó la experiencia comparativa, en experimentos reales.

Pregunta 5 (8 pts):
Una empresa está efectuando el proceso de planificación de su parque de equipos. Según los estudios realizados existen los siguientes costos:

$C(e)$**:** costo de operación anual de un equipo de edad e.

I**:** inversión de un equipo nuevo.

$V(e)$**:** valor residual o de venta de un equipo de edad e.

Además se han realizado estudios sobre productividad, determinando que la productividad de un equipo de edad e es $P(e)$ toneladas/hora. Asuma que el período de planificación es N años, y que al final de éste se venden todos los equipos. Los requerimientos anuales están dados por los valores R_t toneladas en el año t. El máximo de gastos totales por año que se permite es igual a K_t (incluyendo eventual inversión en nuevo equipo y los costos de operación). (Nota: Un año productivo posee 250 días y cada día 16 horas.) Formule un modelo de programación dinámica que permita resolver el problema en el horizonte de planificación indicado.

Pregunta 6 (8 pts):
Considere el siguiente problema de optimización no lineal en variables enteras:

$$\begin{array}{ll} \max & (x_1)^2(x_2)^2x_3 \\ s.a. & \\ & x_1 + 2x_2 + 3x_3 \leq 10 \\ & x_1, x_2, x_3 \geq 0 \text{ y enteros} \end{array}$$

Resuelva mediante Programación Dinámica (Esto sí se puede hacer, pese a que la función objetivo no es lineal).

Pregunta 7 (10 pts):
Considere el siguiente problema de asignación de tareas. Un total de n trabajos deben asignarse a m máquinas de modo que cada trabajo se asigne a una sola máquina. Asumamos además que $n > m$ y de este modo, algunas máquinas deberán procesar más de un trabajo. Sea t_{ij} el tiempo de procesamiento del trabajo j en la máquina i. Interesa determinar una asignación de trabajos a máquinas de modo tal que todo se termine en el menor tiempo posible. Sea x_{ij} igual a 1 si el trabajo j se asigna en la máquina i y 0 en caso contrario. Un modelo de programación entera 0-1 que asigna los trabajos a las máquinas de modo tal que el tiempo de la última máquina en terminar ("makespan" se llama en operaciones) sea mínimo es el siguiente:

$$P) \quad \begin{array}{rll} z^* = \min & \gamma & \\ s.a. & \sum_{j=1}^{n} t_{ij}x_{ij} \leq \gamma \ , \ i = 1, \ldots, m & (1) \\ & \sum_{i=1}^{m} x_{ij} = 1 \ , \ j = 1, \ldots, n & (2) \\ & x_{ij} \in \{0,1\}, i = 1, \ldots, m; j = 1, \ldots, n & (3) \end{array}$$

Aquí, γ es una variables auxiliar que sirve para minimizar el máximo tiempo usado por las máquinas. Es como si la función objetivo fuera $\min_x \max_{i=1,\ldots,m} \sum_{j=1}^{n} t_{ij}x_{ij}$.

Planteamos que este problema puede ser abordado mediente descomposición de Dantzig Wolfe, aplicando la descomposición a la relajación lineal del problema, si se consederan las restricciones (1) como complicantes.

a) (4 pts) Reescriba el problema, en forma ordenada, destacando la estructura en bloques. Hágalo en forma análoga a como se escribió el problema de Flujo Multiproducto en clases.

b) (6 pts) Escriba los pasos de la Descomposición de Dantzig-Wolfe, destacando el problema maestro y el satélite. Use la notación del problema, sea ordenado en toda la presentación.