ICS 2121 Métodos de Optimización, Sem 2023-2
Prof. Jorge Vera

Interrogación 2. Solución

Pregunta 1 (10 puntos + 3 de bono)

a) (5 pts) Suponga que tenemos una "mochila" en la que tenemos que poner distintas cantidades de n tipos de ítems. El llevar una unidad del ítem j proporciona un beneficio β_j. Sin embargo, existen límites totales, tanto de peso como de volumen, a lo que se puede llevar en la mochila. Específicamente, se pueden cargar a lo más K unidades de peso y L unidades de volumen. Una unidad del ítem j pesa p_j y tiene un volumen v_j. Se quiere determinar cuántas unidades llevar de cada ítem de modo de maximizar el beneficio total, respetando simultáneamente las limitantes de peso y volumen.

Escriba una formulación de Programación Dinámica para abordar este problema. Sea claro en especificar las etapas, el estado del sistema, las acciones (o decisiones), y sea claro al escribir las ecuaciones principales del modelo.

Solución:

- Etapas: $1, \ldots, n$
- Estados: Peso (W) y volumen disponible (V)
- Acciones: Se decide sobre la cantidad a introducir de cada ítem (x_t).

La ecuación de Bellman es

$$G_j(W,V) = \max\{\beta_j x_j + G_{j+1}(W - p_j x_j, V - v_j x_j)\} \quad \forall j = 1, \ldots, n-1$$
$$x_j \in \mathbb{Z}^+, x_j \leq \min\left\{\left\lfloor \frac{W}{p_j} \right\rfloor, \left\lfloor \frac{V}{v_j} \right\rfloor\right\}$$

en donde tenemos como condición terminal

$$G_n(W,V) = \max\{\beta_n x_n\}$$
$$x_n \in \mathbb{Z}^+, x_n \leq \min\left\{\left\lfloor \frac{W}{p_n} \right\rfloor, \left\lfloor \frac{V}{v_n} \right\rfloor\right\}$$

Puntaje:

- **(1 pto)** Por definir bien los estados, etapas y acciones
- **(3 ptos)** Por ecuación de Bellman principal
- **(1 pto)** Por condición de borde
- **(-0.5 ptos)** Si está incorrecto la definición de etapas, acciones o estados
- **(-1.5 ptos)** Por errores en la ecuación de Bellman principal.
- **(-0.5 ptos)** Por errores en condición terminal

b) (5 pts) Considere la siguiente variante al problema de producción que fue planteado en la Tarea 2. Una empresa que fabrica un único producto enfrenta demandas para T períodos de tiempo, dadas por d_t para el periodo $t = 1, \ldots, T$. El producto se fabrica en unidades discretas, es decir, $1, 2, \ldots$ unidades. Sea c_t el costo unitario de producción en t y h_t el costo unitario de inventario por período en t. No hay otros costos adicionales pero sí existe una restricción ambiental ya que cada vez que se fabrica una unidad del producto en el período t, se consumen a_t "unidades de emisión de carbono". La regulación impone un límite total de K unidades de emisión sobre todo el horizonte de los T períodos.

Escriba una formulación de Programación Dinámica para abordar este problema. Sea claro en especificar las etapas, el estado del sistema, las acciones (o decisiones), y sea claro al escribir las ecuaciones principales del modelo.

Solución:

- Etapas: $1, \dots, T$
- Estados: Inventario (S) y emisión de carbono a agotar (C).
- Acciones: Cantidad de unidades producidas en el período t (x_t)

La ecuación de Bellman es

$$G_t(S, C) = \min_{x_t \in D(t)} \{c_t x_t + h_t(S - d_t + x_t) + G_{t+1}(S - d_t + x_t, C - a_t x_t)\} \quad \forall t = 1, \dots, T-1$$

en donde tenemos como condición terminal

$$G_T(S, C) = \min_{x_T \in D(T)} \{c_T x_T\}$$

donde $D(t)$ denota el conjunto factible para la política. Cuando t es una etapa intermedia se debe cumplir demanda y no superar el límite de las unidades de carbono disponibles. $D(t)$ queda definido por

$$x_t \geq d_i - I$$
$$x_t \leq \left\lfloor \frac{C}{a_t} \right\rfloor$$
$$x_t \in \mathbb{Z}$$

Para la última etapa asumiremos que no quedará inventario, y la condición será

$$x_t = d_i - I$$
$$x_t \leq \left\lfloor \frac{C}{a_t} \right\rfloor$$
$$x_t \in \mathbb{Z}$$

Puntaje:

- **(1 pto)** Por definir bien los estados, etapas y acciones
- **(3 ptos)** Por ecuación de Bellman principal
- **(1 pto)** Por condición de borde
- **(-0.5 ptos)** Si la definición de etapas, acciones o estados es incorrecta
- **(-1.5 ptos)** Por errores en la ecuación de Bellman principal.
- **(-0.5 ptos)** Por errores en condición terminal

Comentario adicional: Como no se menciona cual debe ser la condición terminal, cualquier otra condición bien justificada y formalmente bien escrita puede estar bien, queda a criterio del ayudante.

c) (3 puntos de bono) Continuando con el problema de la parte b), suponga que las cantidades a producir no fueran discretas sino completamente continuas (es decir, cualquier valor ≥ 0 para x es válido, como sería el caso cuando se fabrican productos a granel). ¿Qué modificaciones debería hacerse a la formulación planteada por usted en b)? Explique con claridad y justifique de acuerdo a los contenidos del curso.

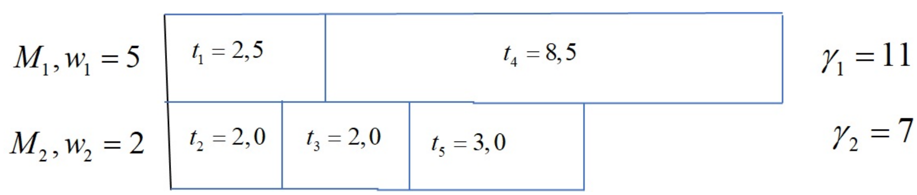
Solución:

Para este inciso, como tenemos unidades continuas, el problema puede interpretarse como un problema de flujo en redes, en donde cada nodo que represente el tiempo hay un equilibrio de flujo teniendo en cuenta que existe una restricción adicional por las emisiones de carbono. En este caso, los arcos a diseñar son las cantidades que se necesitan producir para satisfacer la demanda.

Puntaje:

- **(3 ptos)** Por mencionar y justificar correctamente la relación e interpretación con problema de flujo en redes.3

Pregunta 2 (10 puntos + 3 de bono):
En este problema abordamos la siguiente situación, que se presenta frecuentemente en programación de operaciones en los más diversos ámbitos industriales. Tenemos que realizar n trabajos, los cuales se pueden hacer en cualesquiera de m máquinas. El trabajo j requiere de t_j horas para ser realizado (independientemente de la máquina en que se haga). Si se asignan algunos trabajos a una máquina, esta terminará de trabajar en un tiempo igual a la suma de los tiempos de los trabajos que se asignaron. Ese tiempo, que denotaremos γ_i, para la máquina i, se llama "tardanza" (tardiness, es el término que se usa en Inglés). Existen pesos $w_i > 0$ asociados a cada máquina y queremos encontrar una forma de asignar los trabajos a las distintas máquinas de modo tal que se minimice la "tardanza ponderada total" (total weighted tardiness), dada por $\sum_{i=1}^{m} w_i\gamma_i$.
Adicionalmente, existe el requisito de que cada máquina debe trabajar un tiempo total mínimo de R_{min} horas. Notemos que, una vez asignados algunos trabajos a una máquina, da lo mismo el orden en que se hagan, la tardanza de esa máquina es la misma y la asignación es válida, mientras cumpla el mínimo de horas.
La siguiente figura muestra un ejemplo de 5 trabajos asignados a 2 máquinas (llamadas M1 y M2), con un tiempo mínimo de trabajo $R_{min} = 5$ horas, y con una tardanza ponderada total igual a 69.



Simplemente "a ojo" se puede ver que esa asignación se puede mejorar: basta cambiar el trabajo 1 a la máquina 2 y se tiene una nueva asignación que sigue cumpliendo el trabajo mínimo, y la tardanza ponderada disminuye a 61,5.

Podemos pensar este problema de la siguiente manera, en forma general: existen "opciones de asignación" de trabajos, las que identifican los trabajos que se le asignan a una máquina. Estas opciones, para cada máquina, las podemos describir con un coeficiente asociado a la opción k para la máquina i:

$$a_{jk}^i = \begin{cases} 1 & \text{si en la opción } k \text{ de la máquina } i \text{ el trabajo } j \text{ se asigna a la máquina} \\ 0 & \text{si no} \end{cases}$$

Asumiendo que la opción cumpla con el requisito de tiempo mínimo, estos coeficientes describen completamente las posibles opciones válidas de asignar trabajos a las máquina. En el ejemplo, digamos que la opción 1 es la que se está usando en la máquina 1 y, entonces, estará dada por $a_{11}^1 = 1, a_{21}^1 = 0, a_{31}^1 = 0, a_{41}^1 = 1, a_{51}^1 = 0$, indicando que, para la máquina 1, opción 1, los trabajos 1 y 4 se asignan a esa máquina y los otros no. Otras opciones tendrán valores diferentes, pero todas deben cumplir con el tiempo mínimo de trabajo.
De este modo, podemos ahora definir la "tardanza de la opción k para la máquina i", y que denotaremos por γ_{ki}, como $\gamma_{ki} = \sum_{j=1}^{n} a_{jk}^i t_j$.
Notemos también que la cantidad posible de opciones para una máquina particular puede ser muy grande (a lo más, pueden llegar a ser 2^n). Llamaremos r al número total de opciones válidas posibles para una máquina. Como las máquinas son idénticas, hay la misma cantidad de opciones para cada máquina.

a) (3 pts) Considere el siguiente problema de Programación Lineal Entera, donde y_{ki} es una variable binaria que vale 1 si se usa la opción k de la máquina i, y 0 si no:

$$\begin{aligned} \min \quad & \sum_{i=1}^{m}\sum_{k=1}^{r} w_i\gamma_{ki}y_{ki} & & \\ & \sum_{i=1}^{m}\sum_{k=1}^{r} a_{jk}^i y_{ki} = 1, \; j = 1,\dots,n & & (1) \\ & \sum_{k=1}^{r} y_{ki} = 1, \; i = 1,\dots,m & & (2) \\ & y_{ki} \in \{0,1\}, \quad i = 1,\dots,m; k = 1,\dots,r & & \end{aligned}$$

Argumente que este problema permite decidir qué opción de asignación usar para cada máquina de modo tal de minimizar la tardanza ponderada total y garantizar que todos los trabajos son realizados. ¿Qué condición deben cumplir los coeficientes a_{jk}^i para representar una opción válida?

Solución:

Comenzamos describiendo el problema de optimización dado. En primer lugar, la función objetivo minimiza la tardanza ponderada total. A continuación, la restricción (1) asegura que cada trabajo sea realizado por una única máquina. Finalmente, la restricción (2) asegura que en cada máquina se utilice una única opción. De esta manera, el problema minimiza la tardanza ponderada total garantizando que todos los trabajos sean realizados.

Por otro lado, para que una opción k sea válida, se debe cumplir que todas las máquinas bajo la opción k cumplan con el tiempo de trabajo total mínimo R_{min}. Para ello, los coeficientes a^i_{jk} deben cumplir con la siguiente condición:

$$\gamma_{ki} = \sum_{j=1}^{n} a^i_{jk} t_j \geq R_{min}, \quad \forall i = 1, \ldots, m; \ \forall k = 1, \ldots, r$$

Puntaje:

- **(0,5 pto)** Por describir correctamente la función objetivo
- **(0,5 pto)** Por describir correctamente la restricción (1)
- **(0,5 pto)** Por describir correctamente la restricción (2)
- **(1,5 pto)** Por indicar correctamente la condición que deben cumplir los coeficientes a^i_{jk}

b) (7 pts) Proponga la forma de abordar el problema anterior mediante Generación de Columnas. Sea muy claro en su formulación de problema maestro y del satélite y describa con precisión la forma en que interactúan. ¿Qué tipo de problema es el satélite? ¿Es fácil o difícil de resolver, en su opinión? Justifique sobre la base de los contenidos del curso.

Solución:

Problema maestro:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{m} \sum_{k \in \bar{R}_i} w_i \gamma_{ki} y_{ki} \\
& \sum_{i=1}^{m} \sum_{k \in \bar{R}_i} a^i_{jk} y_{ki} = 1, \ j = 1, \ldots, n \qquad (\pi_j) \\
& \sum_{k \in \bar{R}_i} y_{ki} = 1, \ i = 1, \ldots, m \qquad (\nu_i) \\
& 0 \leq y_{ki} \leq 1, \quad i = 1, \ldots, m; \ k \in \bar{R}_i
\end{aligned}$$

Donde $\bar{R}_i \subseteq \{1, \ldots, r\}$ para todo $i \in \{1, \ldots, m\}$ (lo anterior es equivalente a tomar subconjuntos $i' = \{1, \ldots, m'\}$ y $k' = \{1, \ldots, r'\}$). A partir del problema maestro se obtienen los costos reducidos, y con estos se construye el problema satélite:

$$\begin{aligned}
\bar{C}_{ki} &= w_i \gamma_{ki} - \sum_{j=1}^{n} \pi_j a^i_{jk} - \nu_i \\
&= \sum_{j=1}^{n} w_i a^i_{jk} t_j - \sum_{j=1}^{n} \pi_j a^i_{jk} - \nu_i
\end{aligned}$$

Problema satélite:

$$\begin{aligned}
\eta_i = \min \quad & \sum_{j=1}^{n} w_i u_j t_j - \sum_{j=1}^{n} \pi_j u_j - \nu_i \\
& \sum_{j=1}^{n} u_j t_j \geq R_{min} \\
& u_j \in \{0, 1\}, \quad j = 1, \ldots, n
\end{aligned}$$

La interacción entre el problema maestro y el problema satélite se da de la siguiente manera: se resuelve el problema maestro, el cual envía información (π, ν) al problema satélite. Sea $\bar{\eta} = \min_{i=1,\ldots,m}\{\eta_i\}$ el problema satélite con el menor costo reducido dado (π, ν), entonces:

si $\bar{\eta} \geq 0 \rightarrow$ se ha llegado al óptimo
si $\bar{\eta} < 0 \rightarrow$ la solución del problema satélite entrega una nueva columna para agregar al problema maestro

Si el problema maestro recibe una nueva columna, entonces este se vuelve a resolver y así sucesivamente. Finalmente, observando la forma del problema satélite, se concluye que es un problema de la mochila, el cual es medianamente fácil de resolver con programación dinámica (tal como se vio en clases).

Puntaje:

- **(2 pto)** Por formular correctamente el problema maestro
  - **(-1 pto)** Por formular el problema maestro sin la relajación lineal
  - **(-1 pto)** Por no identificar los nuevos subconjuntos $\bar{R}_i$
- **(3 pto)** Por formular correctamente el problema satélite
  - **(-1 pto)** Por formular un único problema satélite η en lugar de un problema satélite η_i para cada $i = 1, \ldots, m$
- **(1 pto)** Por describir correctamente la forma en que interactúan el problema maestro y el problema satélite
- **(0,5 pto)** Por identificar que el problema satélite es un problema de la mochila
- **(0,5 pto)** Por describir la dificultad de resolución del problema satélite justificadamente

c) (3 pts de bono) En la implementación de Generación de Columnas para este problema, hay una forma natural de hacer "Multiple Pricing". Describa cuál es. Sea preciso en su justificación (hay sólo una respuesta posible a esto, así que piense bien, no basta con que escriba la descripción general de lo que es "Multiple Pricing").

Solución:

Para hacer "Multiple Pricing" en la implementación de Generación de Columnas en este problema, se debe resolver el problema maestro y tomar la información (π, ν) enviada, la cual debe ser evaluada en todos los problemas satélites $i = 1, \ldots, m$. Una vez evaluados todos los problemas satélites, se deben tomar todos aquellos en que $\eta_i < 0$, obteniendo así todas las columnas generadas con costos reducidos negativos, para posteriormente agregarlas al problema maestro y volver a resolverlo, repitiendo el proceso sucesivamente.

Puntaje:

- **(3 pto)** Por describir correctamente la forma de hacer "Multiple Pricing" en este problema

Pregunta 3 (10 puntos):

a) (5 pts) Considere el siguiente problema de Programación Lineal en 2 variables:

$$
\begin{array}{rc}
\max & -4x_1 + \alpha x_2 \\
s.a. & x_1 + x_2 \geq 1 \\
& x_2 \leq 4 \\
& x_1, x_2 \geq 0
\end{array}
$$

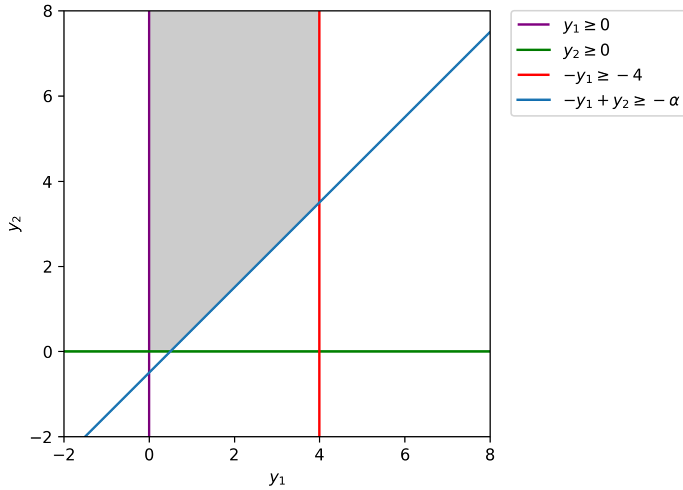
Determine si existe algún valor de α que haga el problema no acotado. Justifique su argumento (ya sea sí o no) en función de los resultados del curso.

Solución:

Sabemos que si el dual es infactible, entonces el primal es no acotado. Para este problema, el dual viene dado por

$$
\begin{array}{rc}
\min & -y_1 + 4y_2 \\
s.a. & -y_1 \geq -4 \\
& -y_1 + y_2 \geq \alpha \\
& y_1, y_2 \geq 0
\end{array}
$$

Esto resulta en el siguiente gráfico (tomamos un $\alpha > 0$ para ilustrar):



De lo anterior, podemos notar que no existe un $\alpha \in \mathbb{R}$ tal que el dual sea infactible, por lo que el primal siempre es acotado.

Otra manera de ver este problema es a través del cono de recesión del sistema

$$\begin{aligned} -x_1 - x_2 &\leq -1 \\ x_2 &\leq 4 \\ -x_1, -x_2 &\leq 0 \end{aligned}$$

Lo que resulta en el cono de la forma

$$\begin{aligned} -x_1 - x_2 &\leq 0 \\ x_2 &\leq 0 \\ -x_1, -x_2 &\leq 0 \end{aligned}$$

que tiene como único rayo el vector $c = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, con $\nabla f(x) = \begin{bmatrix} -4 \\ \alpha \end{bmatrix}$. Luego, tenemos que $c^T \nabla f(x) = -4 < 0$ para todo $\alpha \in \mathbb{R}$. Es decir, como el problema es de minimización, el problema es acotado.

- **(3 ptos)** Por plantear y graficar el problema dual correctamente, o bien por plantear correctamente el cono de recesión y su rayo.
- **(2 ptos)** Por argumentar que el problema siempre es acotado.

b) (5 pts) Considere el siguiente problema de programación lineal:

$$P) \quad \begin{array}{ll} \min & c^T x \\ s.a. & \alpha_1^T x \geq b_1 \\ & \alpha_2^T x \geq b_2 \\ & \vdots \quad \vdots \\ & \alpha_m^T x \geq b_m \\ & x \geq 0 \end{array}$$

donde $c \in \mathbb{R}^n$, $\alpha_i \in \mathbb{R}^n$ y los b_i son escalares, $i = 1, \ldots, m$. Una restricción se dice "redundante" si al quitarla de la formulación, la región factible del problema sigue siendo exactamente la misma.

i) (2 pts) Supongamos que la restricción 1 (uno) es redundante. Argumente que, entonces, el problema

$$P_r) \quad \begin{array}{ll} \min & c^T x \\ s.a. & \alpha_1^T x \leq b_1 - \epsilon \\ & \alpha_2^T x \geq b_2 \\ & \vdots \quad \vdots \\ & \alpha_m^T x \geq b_m \\ & x \geq 0 \end{array}$$

es infactible, con $\epsilon > 0$ chico.

Solución:

Notemos que si se cumple $\alpha_1^T x \geq b_1$, entonces, para $\varepsilon > 0$, necesariamente tenemos $\alpha_1^T x > b_1 - \varepsilon$, lo que quiere decir que es infactible que $\alpha_1^T x \leq b_1 - \varepsilon$.

Por conveniencia, definamos

$$\bar{A} = \begin{bmatrix} \alpha_2^T \\ \vdots \\ \alpha_m^T \end{bmatrix}, \ \bar{b} = \begin{bmatrix} b_2 \\ \vdots \\ b_m \end{bmatrix}$$

Luego, podemos afirmar que si se cumple que $\bar{A}x \geq \bar{b}$, entonces también se cumple que $\alpha_1^T x \geq b_1$. Con todo esto, obtenemos que para P_r siempre se cumple que

$$\forall x \in \mathbb{R}^n \ : \ \bar{A}x \geq \bar{b} \implies \alpha_1^T x \geq b_1 \implies \alpha_1^T x > b_1 - \varepsilon$$

por lo que el sistema es infactible.

- **(2 ptos)** Por argumentar y concluir correctamente que el problema es infactible.

ii) (3 pts) Sobre la base del resultado anterior, construya un sistema de desigualdades y/o igualdades tal que si ese sistema es factible, eso permite certificar que la restricción 1 es redundante (En forma aproximada, sobre la base de ese sistema alternativo, es que software como Gurobi preprocesan un modelo y lo "limpian" de restricciones redundantes, lo cual es muy relevante para la eficiencia computacional).

Solución:

Una de las variantes del Lema de Farkas dice que solo uno de los siguientes sistemas tiene solución:

$$\exists x \in \mathbb{R}^n \ : \ x \geq 0, \ Ax \geq b \tag{1}$$

$$\exists y \in \mathbb{R}^m \ : y \geq 0, \ y^T A \leq 0, \ y^T b > 0 \tag{2}$$

Usando el resultado de **i)**, podemos definir los sistemas

$$\exists x \in \mathbb{R}^n \ : \ x \geq 0, \ \begin{bmatrix} \alpha_1^T \\ -\bar{A} \end{bmatrix} x \leq \begin{bmatrix} b_1 - \varepsilon \\ -\bar{b} \end{bmatrix} \tag{1}$$

$$\exists y \in \mathbb{R}^m \ : y \geq 0, \ y^T \begin{bmatrix} \alpha_1^T \\ -\bar{A} \end{bmatrix} \leq 0, \ y^T \begin{bmatrix} b_1 - \varepsilon \\ -\bar{b} \end{bmatrix} > 0 \tag{2}$$

por lo que si el sistema (2) es factible, sabemos que el sistema (1) es infactible, lo que nos dice que la restricción es redundante.

- **(2 ptos)** Por plantear los sistemas para utilizar el Lema de Farkas.
- **(1 pto)** Por argumentar por qué si el sistema correspondiente es factible, entonces se puede certificar que la restricción es redundante.