Departamento de Informática
Universidad Carlos III de Madrid

Grado en Informática

Heurística y Optimización

27 de Junio de 2014

Normas generales del examen

① El tiempo para realizar el examen es de **4 horas**

② No se responderá a ninguna pregunta sobre el examen transcurridos los primeros **30 minutos**

③ Cada pregunta debe responderse en páginas separadas en el mismo orden de sus apartados. Si no se responde, se debe entregar una página en blanco

④ Escribe con claridad y en limpio, de forma ordenada y concisa

⑤ Si se sale del aula, no se podrá volver a entrar durante el examen

⑥ No se puede presentar el examen escrito a lápiz

Pregunta 1 (1 puntos)

Una empresa de comunicaciones dispone de varios equipos de transmisión digital de datos (*routers*) en varias ciudades de la geografía nacional. Cada equipo está conectado a otros (pero no necesariamente a todos) y conoce, con absoluta precisión, el tiempo de transmision de datos medido en milisegundos entre dos extremos conectados entre sí. La empresa desea conocer, con absoluta seguridad, el menor y mayor tiempo de transmisión entre dos ciudades cualesquiera suponiendo, para ello, que cada paquete se envía siempre a lo largo de la ruta más rápida.

Se pide:

(a) **(½ puntos)** Modelizar el problema como un problema de grafos indicando claramente el significado de cada una de sus partes.

(b) **(½ puntos)** ¿Qué algoritmo de *Programación Dinámica* puede usarse para resolver este problema? Justifica tu respuesta indicando claramente cómo usarlo en este problema en particular.

Pregunta 2 (1½ puntos)

Se ha decidido que el desarrollo de un proyecto informático seguirá las fases tpicas de un ciclo de vida en cascada que son, en orden: *Análisis*, *Diseño*, *Implementación*, *Pruebas* y *Verificación*. La empresa que dirigirá el desarrollo cuenta para ello con un conjunto de profesionales con competencias en varias de estas fases tal y como se indica en la Tabla siguiente:

Fase	Alejandro	Belén	Carmen	Diego	Elena	Fernando
1:Análisis	✓	✓				✓
2:Diseño	✓	✓		✓	✓	
3:Implementación		✓	✓	✓		
4:Pruebas			✓	✓	✓	✓
5:Verificación			✓		✓	✓

Se desea construir un sistema automático que decida la asignación de una (y sólo una) persona a cada fase aunque la misma persona puede trabajar en diferentes fases si se verifican todas las restricciones.

Obviamente, nadie puede ser asignado a una fase en particular si no tiene competencias para élla. Es preciso tener en cuenta que el ciclo de vida contempla que existirán solapamientos entre fases consecutivas de modo que *no es posible asignar la misma persona a dos fases consecutivas.* Por último, los productos de una fase serán usados siempre en la fase siguiente y, por ello, la empresa desea que las personas asignadas a una fase y la inmediatamente siguiente pertenezcan al mismo Grupo de Trabajo. Los Grupos de Trabajo creados son:

Grupo de trabajo 1 : Alejandro, Belén y Diego

Grupo de trabajo 2 : Carmen y Diego

Grupo de trabajo 3 : Elena y Fernando

Se pide:

(a) (½ **puntos**) Modelar este problema como un *problema de satisfacción de restricciones.* Identifica claramente todas las componentes necesarias y justifica todas las decisiones tomadas.

(b) (½ **puntos**) Utiliza *arco-consistencia* para determinar quién puede, y quién no, llevar a cabo la fase de *Análisis* si a Belén se le asigna la fase de *Diseño.*

(c) (½ **puntos**) Considerando únicamente las fases de *Análisis*, *Implementación* y *Verificación*, utiliza *camino-consistencia* para determinar quién puede, y quién no, llevarlas a cabo.

Pregunta 3 (1½ puntos)

Considérese la fórmula en Formal Normal Conjuntiva $F = \bigwedge_{i=1}^{6} C_i$ que contiene las siguientes cláusulas:

$$\begin{array}{ll} C_1\colon (\overline{x}_1 \vee x_4 \vee \overline{x}_5) & C_4\colon (\overline{x}_2 \vee x_3) \\ C_2\colon (x_2) & C_5\colon (\overline{x}_3) \\ C_3\colon (\overline{x}_4 \vee \overline{x}_6) & C_6\colon (x_1 \vee x_6) \end{array}$$

Se pide responder razonadamente las siguientes cuestiones:

(a) (½ **puntos**) Indica qué literales son puros y resuelve la *resolución* de la fórmula F respecto de esos literales

(b) (½ **puntos**) Obtener un modelo que satisfaga la fórmula resultante del apartado anterior con el algoritmo de Davis-Putnam. *Considera, para ello, las variables en orden ascendente de su subíndice.*

(c) (½ **puntos**) A partir de los resultados del algoritmo de Davis-Putnam, ¿cuántos modelos hay que satisfacen la fórmula del apartado anterior?

Pregunta 4 (3 puntos)

Considerése el siguiente problema de Programación Lineal:

$$\begin{array}{rrrrrrrrrrr} & & & \min z = -2x_1 - x_2 + 3x_3 - 5x_4 & & & & & & & \\ & x_1 & + & 2x_2 & + & 2x_3 & + & 4x_4 & \leq & & 40 \\ - & 2x_1 & + & x_2 & - & x_3 & - & 2x_4 & \geq & - & 8 \\ & 4x_1 & - & 2x_2 & + & x_3 & - & x_4 & \leq & & 10 \\ & & & & \mathbf{x} \geq \mathbf{0} & & & & & & \end{array}$$

Se pide responder razonadamente las siguientes preguntas:

(a) (½ **puntos**) Expresar el problema de Programación Lineal anterior en forma *estándar* de maximización.

(b) (½ **puntos**) Responde razonadamente la siguiente cuestión ¿es posible que este problema *no tenga solución*? Si o no y por qué.

(c) **(1 punto)** Resolver el problema de Programación Lineal obtenido en el primer apartado con el algoritmo SIMPLEX haciendo los preparativos que sean precisos para comenzar con la matriz identidad.

Es imprenscindible indicar claramente, en cada iteración: las variables escogidas en la base, su valor, y el valor de la función objetivo

(d) **($1/2$ puntos)** Interpretar el resultado y explicar qué conclusiones pueden extraerse de él.

(e) **($1/2$ puntos)** Calcula la contribución por unidad de recurso del problema de Programación Lineal obtenido en el primer apartado al valor óptimo de la función objetivo.

Preguntas 5 (3 puntos)

Una empresa de comunicaciones dispone de varios equipos de transmisión digital de datos (*routers*) en varias ciudades de la geografía nacional. Cada equipo está conectado a otros (pero no necesariamente a todos) y conoce, con absoluta precisión, el tiempo de transmision de datos medido en milisegundos entre dos extremos conectados entre sí.

Dada una posición final y una posición final, la empresa desea minimizar, inicialmente, el número de equipos intermedios que se deben usar para llegar hasta el destino. Se pide:

(a) **($1/2$ puntos)** Representar el problema como un *espacio de estados*.

(b) **($1/2$ puntos)** ¿Cuál es la profundidad máxima del árbol de búsqueda desarrollado por un algoritmo de búsqueda de fuerza bruta? ¿Cuál es el factor de ramificación máximo?

(c) **($1/2$ puntos)** Sugerir razonadamente el algoritmo de búsqueda de fuerza bruta que mejor se adapta a este problema. Considerar, en particular, el caso de un número alto de equipos de transmisión digital de datos.

(d) **(1 punto)** Después de haber hecho varios experimentos con la configuración anterior, la empresa está interesada ahora en que los datos alcancen su destino *en el menor tiempo posible.*

Se pide diseñar una función heurística $h(n)$ que sea *admisible* y que esté bien informada usando, para ello, la técnica de *relajación de restricciones.*

(e) **($1/2$ puntos)** Sugerir razonadamente el algoritmo de búsqueda heurística que mejor se adapta a este problema.

Soluciones del examen de Heurística y Optimización Junio 2014

Problema 1

1. El enunciado del problema sugiere claramente la consideración de un grafo $G(V, E)$ donde existe un vértice $v_i \in V$ por cada ciudad donde hay un equipo de transmisión digital de datos. Además, existirá un arco $(v_i, v_j) \in E$ si y sólo si existe una conexión directa entre las ciudades representadas simbólicamente por los vértices v_i y v_j.

   Por último, cada arco (v_i, v_j) está etiquetado con el coste de alcanzar el segundo vértice desde el primero que, en el caso de este problema, consiste en el tiempo medido en milisegundos en la transmisión de datos entre las ciudades representadas por esos vértices.

2. El algoritmo de *Programación Dinámica* de Floyd-Warshall calcula, precisamente, el coste del camino más corto entre dos pares cualesquiera de vértices v_i y v_j. Por lo tanto, el resultado de su ejecución consiste en una matriz M cuya componente M_{ij} contiene el tiempo mínimo requerido para llegar al segundo vértice desde el primero conmutando tantas veces como sea necesario a través de la ruta más rápida.

   Por lo tanto, el menor y mayor tiempo solicitado por la empresa son, respectivamente:

$$\min_{i,j=1,|V|} \{M_{ij}\}$$
$$\max_{i,j=1,|V|} \{M_{ij}\}$$

   donde $|V|$ es el número de vértices del grafo o, equivalentemente, el número de ciudades equipadas con un *router*.

Problema 2

1. Un problema de satisfacción de restricciones se define como una terna (X, D, C) donde $X = \{x_i\}_{i=1}^n$ es el conjunto de variables; $D = \{D_i\}_{i=1}^n$ representa los dominios de cada variable respectivamente y $C = \{C_i\}_{i=1}^m$ es el conjunto de restricciones del problema.

   Aparentemente, existen dos formulaciones válidas —como en casi todos los problemas de satisfacción de restricciones. Ambas se discuten separadamente a continuación.

   Opción 1 La primera opción consiste en modelizar quién llevará a cabo cada fase. Puesto que hay 5 fases se definen hasta 5 variables, $X = \{x_i\}_{i=1}^5$:

x_1	$\rightleftharpoons$	Persona que llevará a cabo la fase 1: Análisis
x_2	$\rightleftharpoons$	Persona que llevará a cabo la fase 2: Diseño
x_3	$\rightleftharpoons$	Persona que llevará a cabo la fase 3: Implementación
x_4	$\rightleftharpoons$	Persona que llevará a cabo la fase 4: Pruebas
x_5	$\rightleftharpoons$	Persona que llevará a cabo la fase 5: Verificación

   de donde resulta obvio, entonces, que el dominio de cada variable x_i, D_i, son las personas capacitadas para llevar a cabo el trabajo de la fase i-ésima:

D_1	{Alejandro, Belén, Fernando}
D_2	{Alejandro, Belén, Diego, Elena}
D_3	{Belén, Carmen, Diego}
D_4	{Carmen, Diego, Elena, Fernando}
D_5	{Carmen, Elena, Fernando}

   Las restricciones se representarán como pares R_{ij} que representan los valores simultaneamente factibles para la realización de las fases i y j. A propósito de su formulación es preciso tener en cuenta:

- *El ciclo de vida contempla que existirán solapamientos entre fases consecutivas de modo que no es posible asignar la misma persona a dos fases consecutivas.*
  En otras palabras, $(N, N) \notin R_{i,i+1}$ donde N es cualquiera de las personas consideradas para trabajar en el proyecto. De hecho, este caso no se muestra en el conjunto final de restricciones mostrado en el siguiente punto.
- *La empresa desea que las personas asignadas a una fase y la inmediatamente siguiente pertenezcan al mismo Grupo de Trabajo.*
  Por lo tanto,

R_{12}	(Alejandro, Belén), (Alejandro, Diego), (Belén, Alejandro), (Belén, Diego), (Fernando, Elena)
R_{13}	(Alejandro, Belén), (Alejandro, Diego), (Belén, Belén), (Belén, Diego)
R_{14}	(Alejandro, Diego), (Belén, Diego), (Fernando, Elena)
R_{15}	(Fernando, Elena), (Fernando, Fernando)
R_{23}	(Alejandro, Belén), (Alejandro, Diego), (Belén, Diego), (Diego, Belén), (Diego, Carmen)
R_{24}	(Alejandro, Diego), (Belén, Diego) (Diego, Carmen), (Diego, Diego)
R_{25}	(Diego, Carmen), (Elena, Elena), (Elena, Fernando)
R_{34}	(Belén, Diego), (Carmen, Diego), (Diego, Carmen)
R_{35}	(Carmen, Carmen), (Diego, Carmen)
R_{45}	(Diego, Carmen), (Elena, Fernando), (Fernando, Elena)

  Obviamente, se han omitido las restricciones del tipo $R_{ji}, j > i$ puesto que $R_{ij} = R_{ji}$.

Como puede verse, la primera tarea (Análisis), debe ser llevada a cabo necesariamente por Fernando. Asimismo, la última tarea, Verificación, debe asignarse a Carmen. El motivo es que todas las tuplas de R_{15} empiezan con Fernando y todas las tuplas de R_{35} acaban con Carmen de modo que cualesquiera otras asignaciones son incompatibles con estas restricciones.

Opción 2 La segunda opción consiste en determinar qué tareas realiza cada individuo. Por lo tanto, se definen hasta 6 variables de decisión: $X = \{x_A, x_B, x_C, x_D, x_E, x_F\}$ que representan la tarea que realizarán Alejandro, Belén, Carmen, Diego, Elena y Fernando respectivamente[1].

Sin embargo, esta opción no es viable, al menos con los contenidos estudiados en la asignatura. El motivo es que, como advertía el enunciado, la misma persona puede llevar a cabo más de una fase y, por lo tanto, estas variables de decisión serían *multivaluadas* —por ejemplo, $x_C = \{3, 5\}$ se verifica en las restricciones mostradas anteriormente.

Por lo tanto, la opción preferida es la primera.

2. La arco consistencia entre dos variables x_i y x_j sirve para determinar si para cada valor $a_i \in D_i$ existe otro $a_j \in D_j$ que satisfaga la restricción que relaciona las variables i y j, R_{ij}. Si no fuera así, entonces a_i puede eliminarse del dominio de la primera variable, D_i.

   Las variables involucradas en la pregunta son x_1 y x_2, de modo que la arco consistencia se resolverá atendiendo a las tuplas que hay en R_{12} y que se muestran en la tabla desarollada en el apartado anterior. El único valor legal para x_1 si $x_2 =$ Belén es Alejandro puesto que (Alejandro, Belén) $\in R_{12}$. El resto de valores en el dominio de x_1, (Belén y Fernando), se pueden eliminar puesto que no existen en la restricción R_{12}.

3. La camino consistencia (de longitud 2) entre dos variables x_i y x_j respecto de una tercera x_k, consiste en determinar si para cada asignación de valores $a_i \in D_i$ y $a_j \in D_j$ a las variables x_i y x_j respectivamente compatible con la restricción que las une, R_{ij}, existe un valor $a_k \in D_k$ que sea consistente con las relaciones R_{ik} y R_{jk}. Si no fuera así, la asignación (a_i, a_j) puede eliminarse de la relación R_{ij}

   En este caso, las restricciones involucradas son R_{13}, R_{15} y R_{35} que se muestran de nuevo a continuación por comodidad:

[1] La selección de los nombres no era casual y se hizo para que las variables, en esta opción, tuvieran nombres nemotécnicos.

R_{13}	(Alejandro, Belén), (Alejandro, Diego), (Belén, Belén), (Belén, Diego)
R_{15}	(Fernando, Elena), (Fernando, Fernando)
R_{35}	(Carmen, Carmen), (Diego, Carmen)

donde conviene recordar nuevamente que $R_{ij} = R_{ji}$.

En realidad, puesto que el segundo elemento de todas las tuplas de R_{35} es Carmen, la camino consistencia elimina todas las tuplas de las otras restricciones, R_{13} y R_{15} puesto que ninguna la contiene.

Por lo tanto, el problema es infactible.

Problema 3

1. Un literal se define como la asociación de una variable (por ejemplo, x_1) y su signo. Por lo tanto, x_1 y $\overline{x}_1$ son literales distintos aunque se refieren a la misma variable. Negando cualquiera de ellos se obtiene el otro.

   Un literal ℓ es puro si y sólo si no aparece negado en ninguna cláusula. Todas las variables de la fórmula conjuntiva F aparecen negadas y afirmadas menos x_5 que aparece negada únicamente en la cláusula C_1.

   Por otra parte, la resolución de una fórmula F respecto de un literal puro ℓ es $F \setminus \ell$ que consiste en las mismas cláusulas originales de F menos aquellas que contiene el literal puro ℓ. El resultado de la resolución es razonable puesto que buscando modelos que satisfagan expresiones lógicas en Forma Normal Conjuntiva, las cláusulas que contengan un literal puro se satisfacen automáticamente en cuanto la variable correspondiente toma el valor que se corresponde con el signo del literal. En este caso, $x_5 = \perp$.

   Ahora bien, la fórmula resultante entonces es:

$$\begin{array}{ll} & C_4\colon (\overline{x}_2 \vee x_3) \\ C_2\colon (x_2) & C_5\colon (\overline{x}_3) \\ C_3\colon (\overline{x}_4 \vee \overline{x}_6) & C_6\colon (x_1 \vee x_6) \end{array}$$

   donde se observa, nuevamente, que x_1 y $\overline{x}_4$ son a su vez literales puros puesto que aparecen únicamente en las cláusulas C_6 y C_3 respectivamente. De modo que aplicando nuevamente la resolución respecto de estas clausulas resulta ahora la fórmula F formada únicamente por las siguientes cláusulas:

$$\begin{array}{ll} & C_4\colon (\overline{x}_2 \vee x_3) \\ C_2\colon (x_2) & C_5\colon (\overline{x}_3) \end{array}$$

2. El enunciado advertía, intencionadamente, que se aplicara la resolución considerando las variables en orden ascendente de su subíndice. Esto es, primero x_2 y luego x_3. Por lo tanto, a continuación se aplica el algoritmo con este orden de variables a la fórmula conjuntiva F' que resulta de eliminar las cláusulas que contenías literales puros:

$$F' = C_2 \wedge C_4 \wedge C_5$$

   A continuación se detallan todos los pasos del algoritmo de Davis-Putnam. Primero se aplica el procedimiento de resolución *hacia delante*. Si y sólo si resultara el conjunto vacío, entonces se aplicará la búsqueda del modelo hacia atrás. En otro caso, si resulta la cláusula vacía $\{\varnothing\}$, entonces la fórmula es insatisfacible y se concluye el proceso.

   Paso 0 $G_0 = \{C_2, C_4, C_5\}$

   Tal y como adverta el enunciado, primero se elige la variable x_2 y se almacena la selección en un vector dedicado `varSelect`. Además, el vector `layerSeq` contendrá las cláusulas involucradas en cada paso:

varSelect[0] $= x_2$ *varSelect almacena la seleccion de variables por paso, y layerSeq las clausulas la resolucion crea C_7*

layerSeq[0] $= G_0 \setminus \text{Res}(G_0, x_2) =$
$G_0 \setminus \{C_5, C_7 :\ (x_3)\} =$
$\{C_2, C_4\}$

La ventaja de calcular las cláusulas involucradas en cada paso como una diferencia de conjuntos es que, al mismo tiempo se obtiene el conjunto de cláusulas que deben considerarse en el siguiente paso:

Paso 1 $G_1 = \{C_5, C_7\}$

A continuación se considera la siguiente variable en el orden indicado, x_3:

varSelect[1] $= x_3$
layerSeq[1] $= G_1 \setminus \text{Res}(G_1, x_3) =$
$G_1 \setminus \{\varnothing\}$

Como puede verse, en el último paso se ha generado la cláusula vacía (y, por lo tanto, se ha detectado una contradicción). Por lo tanto, la fórmula conjuntiva F' es insatisfacible y, por ello, también la fórmula conjuntiva original F.

3. Ninguno puesto que la fórmula no es satisfacible.

Problema 4

Este problema resulta de una pequeña variación del problema 2 de la serie de problemas 3.3a (página 82) de Hamdy A. Taha. Investigación de Operaciones. Sexta Edición, Prentice-Hall. 1998.

1. En el primer apartado se pedía expresar el problema de Programación Lineal del enunciado en *forma estándar* (de maximización), precisamente en preparación a su resolución con el algoritmo SIMPLEX.

   Un problema de programación lineal está en forma *estándar* si todas las restricciones son de igualdad, las variables de decisión son no negativas y, por último, el vector de constantes o recursos **b** no contiene términos negativos. Estará, adems, en forma de maximización si la función objetivo maximiza y de minimización en otro caso. El problema, tal y como estaba enunciado, no verifica estas condiciones porque: uno, la función objetivo es de minimización; dos, la segunda restricción tiene un recurso negativo; por último, ninguna restricción es de la forma de igualdad.

   La siguiente formulación resuelve los dos primeros problemas:

   - Para cambiar el sentido de la función objetivo basta con multiplicarla por -1
   - Asimismo, para cambiar el signo del segundo miembro de alguna de las restricciones basta con multiplicarla por -1, cambiando entonces el signo de la desigualdad, si la hubiera.

   Por lo tanto, el problema de Programación Lineal queda de momento como sigue:

$$\begin{array}{ccccccccc} \multicolumn{9}{c}{\max z = 2x_1 + x_2 - 3x_3 + 5x_4} \\ x_1 & + & 2x_2 & + & 2x_3 & + & 4x_4 & \leq & 40 \\ 2x_1 & - & x_2 & + & x_3 & + & 2x_4 & \leq & 8 \\ 4x_1 & - & 2x_2 & + & x_3 & - & x_4 & \leq & 10 \\ \multicolumn{9}{c}{\mathbf{x} \geq \mathbf{0}} \end{array}$$

   Para resolver la segunda dificultad, conviene recordar aquí lo siguiente:

   - Una restricción de la forma $\leq$ está acotada superiormente. Puesto que ninguna variable de decisión puede tomar valores negativos, es preciso *sumar* una *variable de holgura* para forzar la igualdad.

- Análogamente, las restricciones de la forma $\geq$ estn acotadas inferiormente de modo que, con variables de decisión que no pueden tomar valores negativos, es preciso *restar* una *variable de holgura* para forzar la igualdad.

Por lo tanto, el problema de Programación Lineal queda finalmente como sigue en forma *estándar* de maximización:

$$\max z = 2x_1 + x_2 - 3x_3 + 5x_4$$
$$\begin{array}{ccccccccccccccc} x_1 & + & 2x_2 & + & 2x_3 & + & 4x_4 & + & x_5 & & & & & = & 40 \\ 2x_1 & - & x_2 & + & x_3 & + & 2x_4 & & & + & x_6 & & & = & 8 \\ 4x_1 & - & 2x_2 & + & x_3 & - & x_4 & & & & & + & x_7 & = & 10 \end{array}$$
$$\mathbf{x} \geq \mathbf{0}$$

donde x_5, x_6 y x_7 son variables de holgura usadas para convertir las desigualdades en igualdades.

2. El algoritmo del SIMPLEX consiste en la aplicación iterativa de tres pasos: cálculo de las variables básicas, selección de la variable de entrada y selección de la variable de salida hasta que se detecte alguna de las siguientes condiciones:

   - El problema puede mejorar el valor de la función objetivo indefinidamente. Se dice entonces que el problema está *no acotado*. Este caso se detecta cuando todas las componentes y_i de la variable de decisión x_i elegida para entrar en la base son todos negativos o nulos.
   - El problema es irresoluble. Esto ocurre cuando en el segundo paso, todos los costes reducidos son positivos y el primer paso asignó un valor no negativo a alguna variable artificial.
   - Se alcanza una solución factible y puede demostrarse que no es posible mejorarla. Esta condición se detecta como en el segundo caso pero cuando las variables artificiales (si las hubiera) tienen valores nulos.

   En el caso de este problema en particular es imposible que el problema no sea resoluble puesto que no hay variables artificiales. En otras palabras, la ausencia de variables artificiales garantiza la solubilidad del problema original.

   De hecho, una solución factible consiste en hacer nulas (*no básicas*) todas las variables $x_i, 1 \leq i \leq 4$, y asignar a cada variable de holgura $x_i, 5 \leq i \leq 7$, el recurso de la restricción donde aparecen. Esto equivale a iniciar el algoritmo SIMPLEX con una base igual a la matriz identidad y así será, precisamente, como se iniciará el algoritmo en el siguiente apartado.

3. A continuación se aplica el algoritmo SIMPLEX siguiendo los pasos indicados anteriormente:

   Paso 0 Cálculo de una solución factible inicial

   a) Cálculo de las variables básicas
   Tal y como se ha indicado anteriormente (y como se requería explícitamente en el enunciado), la primera iteración se inicia con una base igual a la matriz identidad de dimensión 3 puesto que hay hasta tres restricciones. Por lo tanto, son variables básicas en el primer paso $\{x_5, x_6, x_7\}$.

   $$B_0 = I_3 \qquad\qquad B_0^{-1} = I_3$$
   $$x_0^* = B_0^{-1}b = b = \begin{pmatrix} 40 \\ 8 \\ 10 \end{pmatrix} \qquad z_0^* = c_{B_0}^T x_0^* = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 40 \\ 8 \\ 10 \end{pmatrix} = 0$$

   b) Selección de la variable de entrada
   Como en la primera iteración $y_i = a_i$, se omite el cálculo de los vectores columna y_i. El cálculo de los costes reducidos es, entonces:

$$z_1 - c_1 = c_{B_0} a_1 - c_1 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 4 \end{pmatrix} - 2 = -2$$

$$z_2 - c_2 = c_{B_0} a_2 - c_2 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ -1 \\ -2 \end{pmatrix} - 1 = -1$$

$$z_3 - c_3 = c_{B_0} a_3 - c_3 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + 3 = +3$$

$$z_4 - c_4 = c_{B_0} a_4 - c_4 = \begin{pmatrix} 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 2 \\ -1 \end{pmatrix} - 5 = -5$$

Por lo tanto, entra la variable con el valor más negativo (puesto que es la que provocará el mayor crecimiento neto de la función objetivo), x_4.

c) Selección de la variable de salida
La regla de salida establece que debe salir aquella variable con el menor cociente x_i/y_{ij} donde x_i es la variable elegida en el paso anterior para añadirse a la base y $0 \leq j < 3$ puesto que la base tiene dimensión 3:

$$\min \left\{ \frac{40}{4}, \frac{8}{2}, \cancel{\frac{10}{-1}} \right\}$$

y sale la variable x_6 que es la que se corresponde con la segunda fracción, precisamente la que tiene el menor valor. Nótese que se ha eliminado el tercer cociente, puesto que tiene un denominador negativo.

Paso 1 Mejora de la solución actual (iteración #1)

a) Cálculo de las variables básicas
Puesto que en el paso anterior se eligió la variable x_6 para abandonar la base y, en su lugar, se decidió introducir x_4, la nueva base estará formada por las variables básicas $\{x_4, x_5, x_7\}$.

$$B_1 = \begin{pmatrix} 4 & 1 & 0 \\ 2 & 0 & 0 \\ -1 & 0 & 1 \end{pmatrix} \qquad B_1^{-1} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ 1 & -2 & 0 \\ 0 & \frac{1}{2} & 1 \end{pmatrix}$$

$$x_1^* = B_1^{-1} b = b = \begin{pmatrix} 4 \\ 24 \\ 14 \end{pmatrix} \quad z_1^* = c_{B_1}^T x_1^* = \begin{pmatrix} 5 & 0 & 0 \end{pmatrix} \begin{pmatrix} 4 \\ 24 \\ 14 \end{pmatrix} = 20$$

y, efectivamente, el valor de la función objetivo ha ascendido como corresponde en un problema de maximización.

b) Selección de la variable de entrada
Como la base actual ya no es la matriz identidad, se procede primero al cálculo de los vectores y_i de todas las variables no básicas:

$$y_1 = B_1^{-1}a_1 = \begin{pmatrix} 1 \\ -3 \\ 5 \end{pmatrix} \quad y_2 = B_1^{-1}a_2 = \begin{pmatrix} -\frac{1}{2} \\ 4 \\ -\frac{5}{2} \end{pmatrix}$$

$$y_3 = B_1^{-1}a_3 = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \frac{3}{2} \end{pmatrix} \quad y_6 = B_1^{-1}a_6 = \begin{pmatrix} \frac{1}{2} \\ -2 \\ \frac{1}{2} \end{pmatrix}$$

A continuación se muestra el cálculo de los costes reducidos:

$$z_1 - c_1 = c_{B_1}y_1 - c_1 = \begin{pmatrix} 5 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ -3 \\ 5 \end{pmatrix} - 2 = +3$$

$$z_2 - c_2 = c_{B_1}y_2 - c_2 = \begin{pmatrix} 5 & 0 & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{2} \\ 4 \\ -\frac{5}{2} \end{pmatrix} - 1 = -\frac{7}{2}$$

$$z_3 - c_3 = c_{B_1}y_3 - c_3 = \begin{pmatrix} 5 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ 0 \\ \frac{3}{2} \end{pmatrix} + 3 = \frac{11}{2}$$

$$z_6 - c_6 = c_{B_1}y_6 - c_6 = \begin{pmatrix} 5 & 0 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{2} \\ -2 \\ \frac{1}{2} \end{pmatrix} - 0 = \frac{5}{2}$$

Por lo tanto, la variable x_2 entrará en la siguiente base.

c) Selección de la variable de salida
A continuación se calcula la variable que será reemplazada por x_2. Para ello, se calcula el mínimo de los cocientes x_1^*/y_2:

$$\text{mín}\left\{\cancel{\frac{4}{-\frac{1}{2}}}, \frac{24}{4}, \cancel{\frac{14}{-\frac{5}{2}}}\right\}$$

de modo que sale la segunda variable de la base, x_5 que, de hecho, es la única que no se deshecha por tener un denominador negativo.

Paso 2 Mejora de la solución actual (iteración #2)

a) Cálculo de las variables básicas
En el tercer paso del algoritmo SIMPLEX la base B_2 está formada por las variables $\{x_2, x_4, x_7\}$ de modo que:

$$B_2 = \begin{pmatrix} 2 & 4 & 0 \\ -1 & 2 & 0 \\ -2 & -1 & 1 \end{pmatrix} \qquad B_2^{-1} = \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} & 0 \\ \frac{1}{8} & \frac{1}{4} & 0 \\ \frac{5}{8} & -\frac{3}{4} & 1 \end{pmatrix}$$

$$x_2^* = B_2^{-1}b = b = \begin{pmatrix} 6 \\ 7 \\ 29 \end{pmatrix} \quad z_2^* = c_{B_2}^T x_2^* = \begin{pmatrix} 1 & 5 & 0 \end{pmatrix} \begin{pmatrix} 6 \\ 7 \\ 29 \end{pmatrix} = 41$$

y, efectivamente, el valor de la función objetivo vuelve a ascender nuevamente.

b) Selección de la variable de entrada
Nuevamente, se procede primero al cálculo de los vectores y_i para todas las variables no básicas (es decir, aquellas susceptibles de formar parte de la nueva base, si la hubiera):

$$y_1 = B_2^{-1}a_1 = \begin{pmatrix} -\frac{3}{4} \\ \frac{5}{8} \\ \frac{25}{8} \end{pmatrix} \quad y_3 = B_2^{-1}a_3 = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{3}{2} \end{pmatrix}$$
$$y_5 = B_2^{-1}a_5 = \begin{pmatrix} \frac{1}{4} \\ \frac{1}{8} \\ \frac{5}{8} \end{pmatrix} \quad y_6 = B_2^{-1}a_6 = \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{4} \\ -\frac{3}{4} \end{pmatrix}$$

Y ahora ya es posible calcular los costes reducidos de todas las variables no básicas como se indica a continuación:

$$z_1 - c_1 = c_{B_2}y_1 - c_1 = \begin{pmatrix} 1 & 5 & 0 \end{pmatrix} \begin{pmatrix} -\frac{3}{4} \\ \frac{5}{8} \\ \frac{25}{8} \end{pmatrix} - 2 = \frac{3}{8}$$
$$z_3 - c_3 = c_{B_2}y_3 - c_3 = \begin{pmatrix} 1 & 5 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{3}{2} \end{pmatrix} + 3 = \frac{11}{2}$$
$$z_5 - c_5 = c_{B_2}y_5 - c_5 = \begin{pmatrix} 1 & 5 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{4} \\ \frac{1}{8} \\ \frac{5}{8} \end{pmatrix} - 0 = \frac{7}{8}$$
$$z_6 - c_6 = c_{B_2}y_6 - c_6 = \begin{pmatrix} 1 & 5 & 0 \end{pmatrix} \begin{pmatrix} -\frac{1}{2} \\ \frac{1}{4} \\ -\frac{3}{4} \end{pmatrix} - 0 = \frac{3}{4}$$

De modo que no entra ninguna variable puesto que todos los costes reducidos son no nulos.

Por lo tanto, el algoritmo SIMPLEX ha concluido con la solución:

$$x^* = \begin{pmatrix} 0 \\ 6 \\ 0 \\ 7 \\ 0 \\ 0 \\ 29 \end{pmatrix}$$

para el que la función objetivo tiene un valor $z^* = 41$

4. La interpretación de un problema incluye varias consideraciones como son estudiar: si el problema es o no satisfacible, si la solución es única o hay varias soluciones o si está o no acotado. Además, debe estudiarse el uso de recursos: si sobra o no alguno y cual es su contribución al crecimiento de la función objetivo.

Interpretación de la solución De la solución se puede advertir lo siguiente:

- El problema es factible (como ya se había anticipado en la respuesta del segundo apartado) puesto que la solución no contiene valores positivos para alguna variable artificial. En particular, de hecho ni siquiera hay variables artificiales en el problema.

- La solución es única porque los costes reducidos son todos estrictamente positivos. Eso significa que cualquier cambio en la base implicaría un decremento neto en el valor de la función objetivo mientras que el caso de *soluciones múltiples* ocurre cuando una cantidad infinita de soluciones tiene el mismo valor de la función objetivo.
- El valor de la función objetivo está acotado porque siempre se pudo aplicar la regla de salida con, al menos, alguna variable básica.

Interpretación de los recursos De los recursos se advierte:

- En la solución óptima sobran hasta 29 unidades del tercer recurso como lo atestigua el valor *óptimo* de la variable de holgura x_7. Es decir, se trata de un *recurso sobrante*.
- Por último, es muy conveniente estudiar la contribución por cada unidad de recurso al crecimiento de la función objetivo. Esto, en particular, se hace con el análisis del problema dual en el siguiente apartado.

5. Para la resolución del último apartado, basta con recordar la *interpretacíń económica* de las soluciones de un problema dual que advierte que:

> La variable dual $x_i'^*$ indica la contribución por unidad del recurso i-ésimo b_i a la variación el en valor óptimo z^* actual del objetivo

Puesto que el enunciado requiere la contribución unitaria de todos los recursos, entonces es preciso calcular la solución completa del problema dual.

Para ello, es posible iniciar la aplicación de otro SIMPLEX. Sin embargo, en su lugar es preferible hacer uso del siguiente resultado teórico:

> Si el problema de programación lineal en forma simétrica tiene una solución óptima correspondiente a una base $\mathbf{B}$, entonces $\mathbf{x'}^T = \mathbf{c}_\mathbf{B}^T\mathbf{B}^{-1}$ es una solución óptima para el problema dual

En este teorema, los términos usados para el cálculo de la solución óptima del problema dual se refieren al problema primal, salvo que se indique explcitamente lo contrario. Por lo tanto c_B^T es el vector de costes de las variables básicas en la solución del problema primal y B la base usada para el cálculo de la misma solución —que se mostró en el último paso de aplicación del SIMPLEX. Por el contrario, x'^T es la solución del problema dual.

En particular:

$$x'^* = c_B^T B^{-1} = (1 \quad 5 \quad 0) \begin{pmatrix} \frac{1}{4} & -\frac{1}{2} & 0 \\ \frac{1}{8} & \frac{1}{4} & 0 \\ \frac{5}{8} & -\frac{3}{4} & 1 \end{pmatrix} = \begin{pmatrix} \frac{7}{8} & \frac{3}{4} & 0 \end{pmatrix}$$

de donde resulta que la contribución del primer recurso al crecimiento de la función objetivo es $\frac{7}{8}$, mientras que el del segundo recurso es de $\frac{3}{4}$ por unidad de recurso. No es una casualidad que la contribución del tercer recurso al crecimiento de la función objetivo sea nulo puesto que, como ya se advirtió en el apartado anterior, se trata de un recurso sobrante.

Problema 5

1. La representación de un problema como un *espacio de estados* requiere identificar dos componentes diferentes:

Estados con la descripción de configuraciones factibles del problema.

Operadores con la descripción de funciones que permiten transitar desde un estado a otro.

Los primeros se asocian al conjunto V de vértices de un grafo G y los segundos al conjunto de arcos E de modo que existe un arco $(v_i, v_j) \in E$ si y sólo existe un operador del problema que permite transitar desde el estado representado por el vértice v_i al estado representado por el vértice v_j. Por lo tanto, el espacio de estados se representa de forma natural mediante un *grafo de búsqueda* $G(V, E)$.

El problema propuesto consiste en encontrar el camino más corto desde un estado inicial s a otro final t. Por lo tanto, cada estado está identificado únicamente con la localización actual del paquete de datos.

En este problema existe un único operador que consiste en *dirigir* el paquete de datos desde la posición actual indicada en un estado a alguno de los que son inmediatamente adyacentes.

Una consideración de suma importancia en la definición de espacios de estados es el coste de los operadores. Puesto que en este caso sólo se desea minimizar el número de equipos intermedios que se deben usar para llegar hasta el destino, el coste de todos los arcos es el mismo e igual a la unidad —igual, por lo tanto, al número de equipos que deben usarse en cada transición.

2. Obviamente, la ruta óptima en un grafo entre dos vértices no puede repetir ningún vértice, de modo que la profundidad máxima de cualquier árbol de búsqueda en un problema del *camino más corto* es $d_{\text{máx}} = |V| - 1$ donde $|V|$ es el número de vértices o, en el caso de este problema, el número de equipos de transmisión de datos.

   Por el mismo motivo, cada equipo de transmisión digital de datos podría llegar a estar conectado a todos los demás equipos menos él mismo. Por lo tanto, el mayor factor de ramificación es igualmente $b_{\text{máx}} = |V| - 1$.

3. Puesto que la empresa está interesada en encontrar una solución óptima se descarta el algoritmo del *primero en profundidad* puesto que es no admisible. Además, puesto que los operadores tienen todos el mismo coste y las soluciones no tienen por qué estar a la misma profundidad, se descarta también el algoritmo de *ramificación y acotación en profundidad*.

   Por lo tanto, son selecciones factibles las siguientes:

   Algoritmo de el primero en amplitud Se trata de un algoritmo de búsqueda no informada *completo* (esto es, que garantiza que encontrará una solución si existe alguna) y *admisible* —y que, por lo tanto, garantiza que la solución encontrada será óptima.

   Nótese que mientras su completitud no depende de la distribución de costes de los operadores, su admisibilidad sí lo hace.

   Este algoritmo nunca re-expande ningún nodo pero tiene un consumo de memoria que crece exponencialmente con la profundidad del árbol de búsqueda.

   Algoritmo de el primero en profundización iterativa Se trata de un algoritmo de búsqueda completo y admisible si incrementa el parámetro de profundidad máxima en una unidad entre iteraciones.

   Este algoritmo re-expande el mismo nodo un número arbitrario de veces pero, a cambio, tiene un consumo de memoria lineal en la profundidad del árbol de búsqueda.

   Como en el apartado anterior se había calculado que la profundidad podría llegar a ser tan alta como $|V| - 1$ y el enunciado advierte, explícitamente, que debe considerarse el caso de conjuntos de vértices con una alta cardinalidad, el mejor algoritmo será el segundo.

4. El operador *dirigir* identificado en el primer apartado está constreñido únicamente por la relación de adyacencia entre ciudades. Por lo tanto, se sugiere violar esta restricción de modo que el problema *relajado* resultante sea óptimamente resoluble y produzca, además, información útil para la resolución de los problemas de este tipo con algoritmos de búsqueda heurística.

En particular, podría relajarse el problema asumiendo simplemente que es posible moverse desde una ciudad hasta cualquier otra. Esto es, asumir que todos los equipos de transmisión digital están conectados entre ellos. De esta forma, la distancia a recorrer en la transmisión de datos entre dos ciudades arbitrarias localizadas en las coordenadas (x_i, y_i) y (x_j, y_j) sería igual a la *distancia aérea* entre ellas:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

Ahora bien, el problema no pide explícitamente minimizar la distancia recorrida entre dos ciudades, sino el tiempo para la transmisión de datos desde una ciudad hasta otra. Por lo tanto, la función heurística elegida debe producir datos en estas mismas unidades. Para hacerlo, basta con asumir que la transmisión se hará a la mayor velocidad observada y que será proporcional a la distancia entre ambas ciudades.

La velocidad de transmisión más rápida, que se denota como c^*, se calcula como sigue:

$$c^* = \min_{\forall u,v} \left\{ \frac{t_{uv}}{d_{uv}} \right\}$$

donde t_{uv} son los tiempos conocidos por la empresa para la transmisión de datos entre ciudades conectadas entre sí. Por lo tanto, la función heurística que estima el tiempo para que los datos lleguen hasta la ciudad j-ésima desde la ciudad i-ésima pero que no excede el tiempo real es:

$$h(i, j) = d_{ij} \times c^*$$

5. La selección de un algoritmo de búsqueda informada para este caso depende, enteramente, de la dificultad del problema:

A* El algoritmo A* es admisible y garantiza, por lo tanto, que encontrará soluciones óptimas si la función heurística que lo guía también es admisible. Además, no reexpande nodos. Sin embargo, tiene un consumo de memoria que crece exponencialmente con la profundidad a la que se encuentra la solución óptima.

IDA* El algoritmo IDA* reexpande nodos pero no necesita ordenarlos (con el consiguiente ahorro de tiempo de CPU) y, mucho más importante aún, tiene un consumo de memoria lineal en la profundidad de la solución (que en nuestro caso está acotada por $|V| - 1$). Además, también es un algoritmo de búsqueda admisible.

Por lo tanto, para la resolución de instancias sencillas se podría sugerir el primer algoritmo pero, para las instancias más complicadas se recomienda el segundo y, en general, el segundo es el más apropiado si el problema que se pretende resolver no tiene muchas transposiciones.