

Pauta del control de Optimización Combinatorial.

Pregunta 1.-

~~Def~~ (1.1).

Note que encontrar un grafo acíclico es decir que queremos encontrar un forest, para esto proponemos el siguiente algoritmo:

① ordenemos $E = \{e_1, e_2, \dots, e_m\}$ de modo que $c(e_1) \geq c(e_2) \geq \dots \geq c(e_m)$

② Pongamos $F := (V(G), \emptyset)$

③ Para $i := 1$ hasta m hacemos

$\Theta(n) \rightarrow$ Si $T + e_i$ no contiene un circuito entonces
| $T := T + e_i$

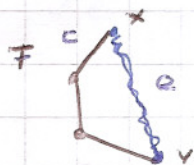
m iteraciones de $\Theta(m) \Rightarrow \Theta(mn)$

(Notemos q' este es el algoritmo de Kruskal un poco modificado)

Es claro q' el algoritmo construye una foresta de G en $\Theta(mn)$

Para ver su validez supongamos que el algoritmo nos entrega una Foresta ~~F~~ F

notemos que lo que cumplen los arcos en F debido al algoritmo es:
 $\forall e = (x, y) \in E(G) \setminus E(F)$ ningún arco en el camino $x-y$ en F tiene un costo menor que e .



$$\forall e \in E(G) \setminus E(F) \quad \forall f \in \mathcal{L}_x^y(F) \quad c(f) \geq c(e)$$

Supongamos que F no tiene peso máximo, entonces existe otro F' tq $c(F') > c(F)$ luego ~~existe una arista~~

~~$a \in E(F')$ tq $a \notin E(F)$ sea $a = (x, y)$ y tomemos~~
~~en F $\mathcal{L}_x^y$ obtenemos que $f \in E(\mathcal{L}_x^y(F))$ en F $c(f) \geq c(a)$~~

Ahora por cada $e \in F \setminus F'$ podemos hacer el análisis:
 si existe $c \in \mathcal{B}_y^*$ en F tq $c(c) > c(e)$ está listo
 pues podemos ver que

$$C(F') = \sum_{e \in F'} c(e) < \sum_{c \in F' \setminus F} c(c) + \sum_{e \in F \cap F'} c(e)$$

luego tenemos que para todo $c \in \mathcal{B}_y^*$ en F $c(c) = c(e)$,
 aplicando este argumento $(n-1)$ veces se obtiene que

$$C(F') = C(F) \text{ por tanto } F \text{ es de costo máximo}$$

(1.2)

Simplemente tendríamos que buscar en la salida que entrega
 el algoritmo un ciclo ($\mathcal{O}(m)$) y si lo encontramos:

1) el ciclo tiene costo negativo $\Rightarrow$ el digrafo no tiene
 costos conservativos

2) la solución no es óptima

si no encontramos un ciclo $\Rightarrow$ la solución es óptima.

(1.3) ~~Tomamos~~ Llamemos s al nodo que ~~toma~~ el algoritmo
 de Dijkstra como inicio de los caminos, lo que significa
 es que para todo nodo $v \in V(G)$ siempre existe un
 camino de longitud (# de arcos) menor o igual a 2.
 $\forall v \in V(G) \exists \mathcal{B}_s^v$ tq $|\mathcal{B}_s^v| \leq 2$.

$$\text{donde } |\mathcal{B}| = |E(\mathcal{B})|.$$

(1.4)

Sea 1 el primer nodo que toma el algoritmo, significa
 que para todo nodo v ($v \neq 1$) existe siempre un
 circuito (de costo negativo) de la forma $\forall v, 1 \in \mathcal{B}_v^v$



21 February

Note que el requerimiento de Beaucorff se puede poner como: Mas equipos

Power curve:

MS

rodinba

ОКРУЖ

2

5000

3)

July

5

A gas to

9

Septembre

2

Para ~~res~~ modelar ~~los problemas que tiene la empresa~~
este problema ~~vous faire~~ nous faire ~~un~~ un grafo, suponiendo
que la empresa no ~~va a~~ ~~dejar~~ ~~mas~~ ~~que~~ un ~~e~~ puede
coger de cumplir con Beacheft con ~~un~~ ~~de~~ un equipo
mi ~~no~~ tener instalado ~~un~~ ~~de~~ un equipo en demanda.
(Estar ~~de~~ ~~financiar~~ ~~supuestos~~ ~~solo~~ ~~para~~ ~~que~~ ~~el~~ ~~ejercicio~~ ~~sea~~
mas ~~cor to~~)

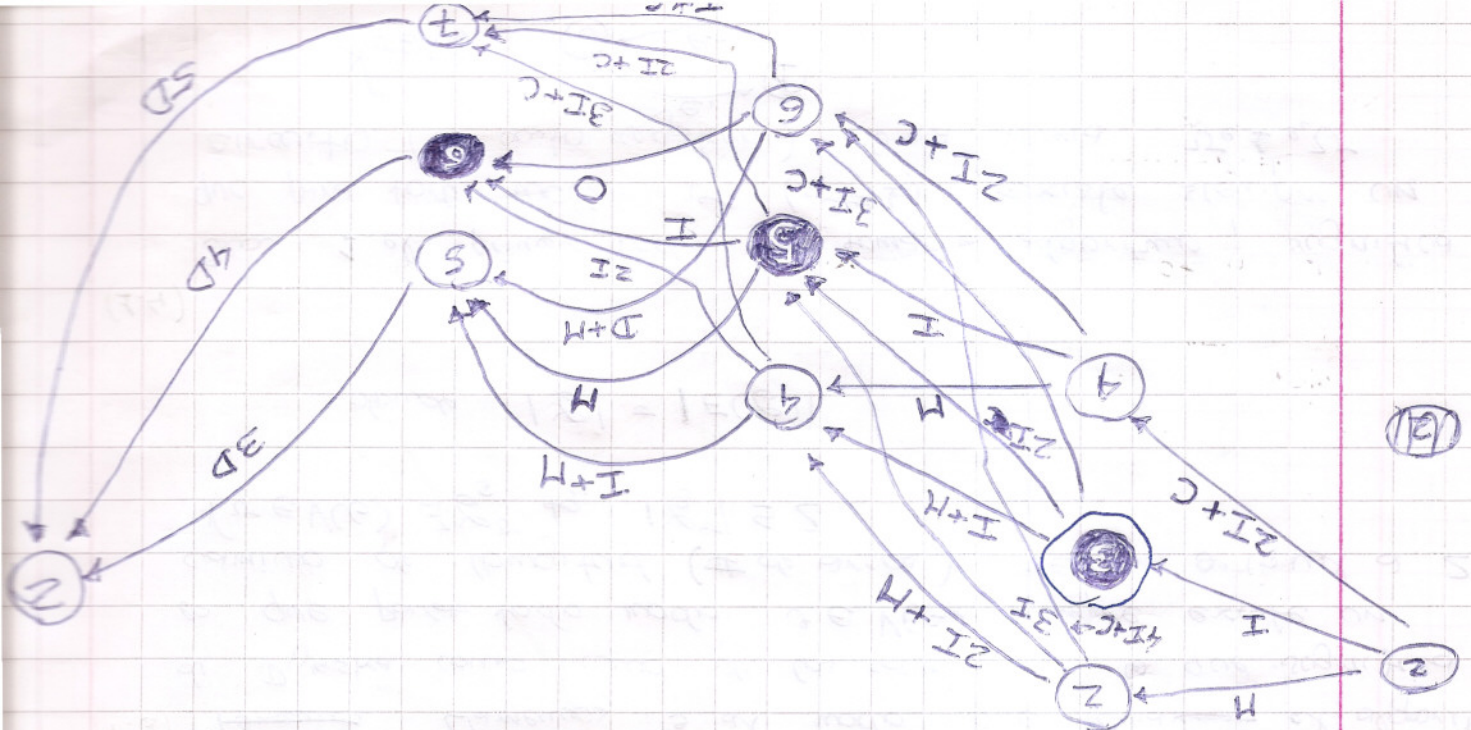
ОУО

Q. No 2

01709

AGOST.

SEP 7



$$M = \text{Multa} = 250$$

$$C = \text{Costo de Mantenimiento} = 200$$

$$I = \text{Instalación} = 100$$

$$D = \text{Desinstalación} = 160$$

Para resolverlo utilizaremos Dijkstra (Algoritmo de Camino mínimo) donde el nodo 3 es el nodo

②-Mayo y el de llegada es ②-septiembre, y vemos que

iteración ①

$$R = \{ \text{②-mayo} \}$$

Junio

$$\times l(2) = M = 250$$

$$\times l(3) = I = 100$$

$$\times l(4) = 2I + C = 400$$

iteración ②

$$R = \{ \text{②-mayo}, \text{③-junio} \}$$

JULIO

$$l(4) = 100 + I + M = 450$$

$$l(5) = 100 + 2I = 300$$

$$l(6) = 100 + 3I + C = 600$$

iteración ③

$$R = \{ \text{②-mayo}, \text{③-junio}, \text{②-junio} \}$$

JULIO

$$\times l(4) = \begin{array}{r} 250 + 2I + M = 400 \\ 450 \end{array} \quad 450$$

$$\times l(5) = \begin{array}{r} 250 + 3I = 550 \\ 300 \end{array} \quad 300$$

$$\times l(6) = \begin{array}{r} 250 + 4I + C = 850 \\ 600 \end{array} \quad 600$$

Iteración 4

$$P = \{ \textcircled{2} \text{ - mayo}, \textcircled{3} \text{ - junio}, \textcircled{5} \text{ - julio} \}$$

AGOSTO

$$\times 550 = X(5) = 500 + 50 = 550$$

$$\times 400 = X(6) = 300 + 100 = 400$$

$$\times 700 = X(7) = 500 + 200 = 700$$

Iteración 5

$$P = P \cup \{ \textcircled{6} \text{ - agosto} \}$$

Iteración 6

$$P = P \cup \{ \textcircled{4} \text{ - junio} \}$$

JULIO

$$X(4) = 400 + 50 = 450$$

$$X(5) = 400 + 100 = 500$$

$$X(6) = 400 + 200 = 600$$

Iteración 7

$$P = P \cup \{ \textcircled{4} \text{ - julio} \}$$

Iteración 8

$$P = P \cup \{ \textcircled{5} \text{ - agosto} \}$$

SEPTIEMBRE

$$X(7) = 550 + 150 = 700$$

$$X(8) = 700 + 200 = 900$$

$$X(9) = 1030$$

it ⑨

$$Z = Z \cup \{\text{julio} - ⑥\}$$

Agosto

$$l(7) = 600 + 100 + 0 = 700$$

it ⑩

$$Z = Z \cup \{\text{agosto} - ⑦\}$$

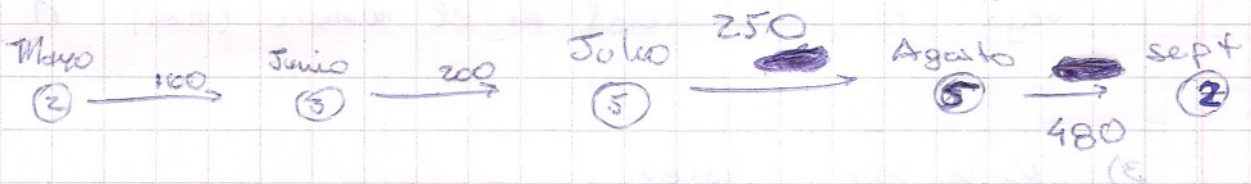
Septiembre

$$l(2) = 700 + 240 = 940$$

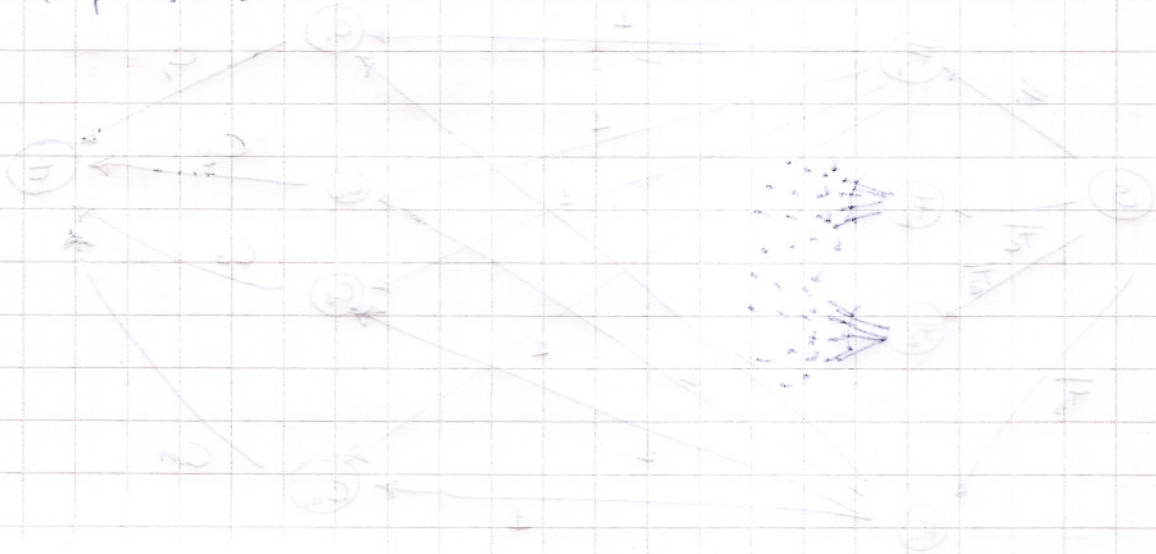
it ⑪

$$Z = Z \cup \{\text{septiembre} - ⑧\}$$

⇒ el costo total es ~~840~~ 1030 con la ruta



La elección del algoritmo Dijkstra es porque Dijkstra es lo más rápido, siempre q' se pueda aplicar, y en este caso se puede, ~~siendo~~ debido a q' todos los costos son positivos.



P3]

1) Input: Un vector con n componentes enteras
 Un vector con m componentes enteras

Output: Una matriz V de $n \times m$ (n filas, m columnas)

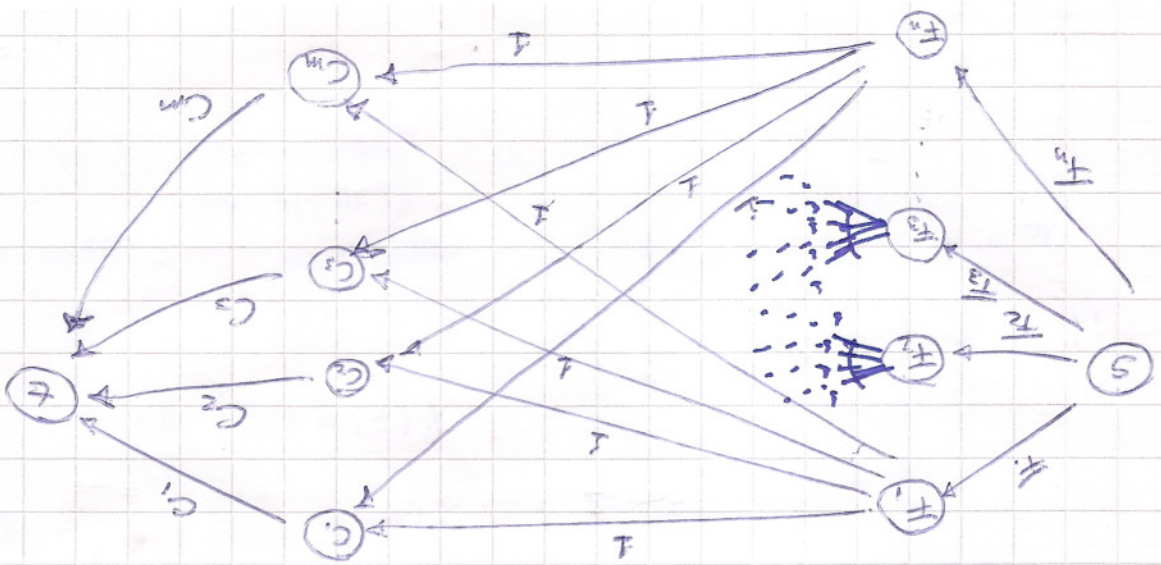
de booleanos (0's o 1's) tal que para $i=1, \dots, n$
 $\sum_{j=1}^m H(i,j) = F_i$, y para $j=1, \dots, m$ $\sum_{i=1}^n H(i,j) = C_j$

si tal matriz existe.

2) Notamos que para ~~formar~~ una matriz con 0's o 1's hay $2^{n \times m}$ formas, por tanto observamos probar (con fuerza bruta) $\%$ de ellos.
 luego es exponencial en el numero (max) n

3) No lo voy a hacer.

4) Hagamos un grafo que ~~establezca~~ las restricciones del problema.



Si resolvemos el problema de flujo máximo en la red anterior
tenemos que si es posible enviar un flujo que
sature las capacidades F_i y C_j se puede obtener
la matriz $(M(i,j))_{i=1, \dots, n}^{j=1, \dots, m}$ y de esta manera los arcos

el flujo en los arcos
 e_{ij} determinan si la matriz tiene un 0 o 1 en
 M_{ij} .