Cuarta semana

- Seguimos aprendiendo a linealizar restricciones que no cumplen alguno de los supuestos básicos de la Programación Lineal
- Empezamos a trabajar con problemas combinatorios:
  - ⌘Problema del viajante

    Un problema combinatorio clásico

El costo de mantenimiento

- Se agrega un costo de mantenimiento de la máquina que se viene utilizando en la producción.

$[2 X1 + 3 X2 + 2 X3 + X4 + 3 X5 + 4 X6 \leq 150 \text{ (h/sem)}]$

- ✔Si la máquina funciona entre 1 y 30 horas, se pagará un costo de mantenimiento de $1 por hora.
- ✔Si funciona más de 30 horas pero menos de 60, se pagará un costo de mantenimiento de $0,70 la hora.
- ✔Si funciona 60 horas o más, el costo de mantenimiento será de $0,50 la hora.

Costo diferencial por intervalo

Guerra de precios

- La recesión sigue haciendo estragos: tenemos que aumentar los precios para que nos siga conviniendo vender producto 2.
- El nuevo esquema de precios es así: a las unidades de producto 2 que vendamos hasta 10 unidades, las venderemos al precio actual de $18/un, las vendidas por encima de 10 y hasta 17 las cobraremos a $20/un y las que vendamos por encima de 17 unidades las cobraremos $24/un. (Por ejemplo, si vendemos 25 unidades cobraremos 10 a $18, 7 a $20 y 8 a $24)

Función cóncava seccionalmente lineal

Problema del viajante - TSP

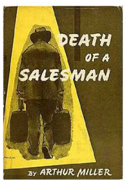
Robert Bosch: Mona Lisa con 100000 ciudades

Problema del viajante - TSP

- Un viajante tiene que partir de su casa y visitar una serie de clientes antes de retornar finalmente a su casa
- No puede dejar de visitar ningún cliente
- Se conocen las distancias entre cada par de clientes y entre cada cliente y la casa del viajante (llamamos *cij* a la constante que indica la distancia entre un lugar i y un lugar j)

Problema del viajante – TSP

- Fue definido en el siglo XIX por el matemático irlandés W.R. Hamilton (foto) y por el matemático británico Thomas Kirkman que lo llamaron problema del mensajero (o cartero)



- En el siglo XX el matemático norteamericano Hassler Whitney le dio el nombre de Traveling Salesman Problem (problema del viajante o agente viajero)

Tipos de problema

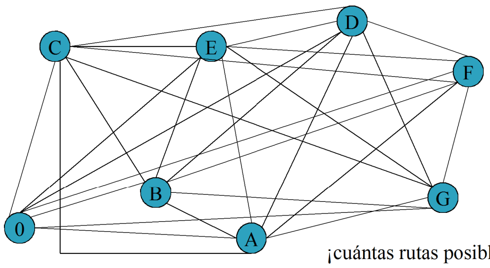
- Dependiendo de si la dirección en la cual se atraviesa un eje importa o no:
- Viajante simétrico: No importa la dirección
  $(C_{ij} = C_{ji})$
- Viajante asimétrico: Importa la dirección

Formulación

- Variables: Yij : { vale 1 si el tour va directamente de la ciudad i a la ciudad j o vale cero en el caso que no va de i a j}
- Exactamente una ciudad debe ser visitada después de la ciudad i
- Exactamente una ciudad debe ser visitada antes de la ciudad j

Un ejemplo de viajante

- Supongamos que el viajante tiene que recorrer siete ciudades (A, B, C, D, E, F, G)



¡cuántas rutas posibles!

Un problema conocido

- Una formulación equivalente en términos de la teoría de grafos es la de encontrar en un grafo completamente conexo y con arcos ponderados el ciclo hamiltoniano de menor costo.
- En esta formulación cada vértice del grafo representa una ciudad, cada arco representa una ruta y el peso asociado a cada arco representa la longitud de la ruta.

Formulación matemática

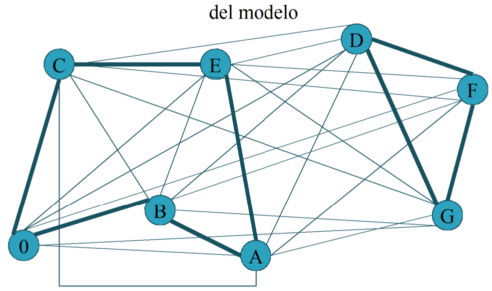
$$\sum_{\substack{j=0 \\ i \neq j}}^{n} Yij = 1 \qquad \forall i = 0,1,2,\ldots\ldots n$$

$$\sum_{\substack{i=0 \\ i \neq j}}^{n} Yij = 1 \qquad \forall j = 0,1,2,\ldots\ldots n$$

$$Z = \sum^{n} C\,ij\,Yij \quad \text{(MINIMO)}$$

Una solución podría ser

(vemos resaltadas las rutas que hará el viajante, según el resultado del modelo



Subtours

- Lo que sucedió es que se formaron dos tours distintos (llamados subtours) cuando debería haber un tour único.
- Podemos ver el archivo "VIAJANTE SIN UI.txt" donde muestra un viajante de 4 ciudades (más la ciudad cero) donde toman valor = 1 las siguientes variables: y01, y12, y20, y34 e y43. Es decir, el tour va de 0 a 1, de 1 a 2 y vuelve a 0. Por otra parte va de 3 a 4 y de 4 a 3.

Debemos evitar la posibilidad de "Subtours"

Para ello agregaremos las siguientes variables y vínculos:

Ui: Número de secuencia en la cual la ciudad i es visitada, $\forall$ i = 1,2,…….,n

$$Ui - Uj + n\ Yij \leq n-1$$

$\forall$ i = 1,2,…….,n
$\forall$ j = 1,2,…….,n
$\forall$ i $\neq$ j

¿Cómo funcionan las Ui?

- Las Ui son variables enteras aunque no se las defina como enteras
- Secuencia:

  Ui – Uj + N * Yij <= N – 1

  si Yij = 0 $\Rightarrow$ Ui – Uj <= N – 1 , obliga que la diferencia de secuencia de visitas entre 2 ciudades que no son visitadas directamente sea menor igual que N –1.

  si Yij = 1 $\Rightarrow$ Ui – Uj <= N – 1 – N = – 1

  $\Rightarrow$ Uj >= Ui + 1 , obliga que la secuencia de visita de la ciudad j que es visitada inmediatamente después que la ciudad i, debe ser por lo menos mayor o igual que 1

¿Cómo funcionan las Ui?

- Correlación Unitaria: El significado de las variables Ui es el número de secuencia en el cual es visitada la ciudad i.

  O sea si Yij = 1 $\Rightarrow$ Uj = Ui + 1.

No puedo ir de la ciudad orden N a ninguna otra excepto la 0 $\Rightarrow$ en particular Yij (para i con Orden N y j con Orden 1) es igual a 0.

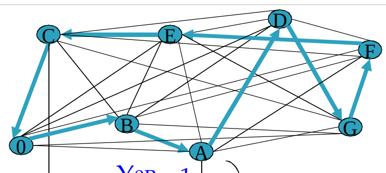
O sea

Ui – Uj + N * Yij <= N –1 ……. N – 1 <= N –1

la única forma para que dé así es que se incremente la secuencia de a 1(ya que se deben incrementar por algo >= que 1, Secuencia).

¿Cómo funcionan las Ui?

- Evitar subtours: Por las ecuaciones de la transparencia anterior, no se puede volver a una ciudad que tiene una Ui con valor superior al valor de la Ui de la ciudad en la cual estamos (a la única que se puede volver es a la cero porque no tiene Ui)
- En el archivo "VIAJANTE CON UI.txt" se ve el mismo problema de las cuatro ciudades que vimos antes, pero ahora arma un único tour, gracias al uso de las Ui



$U_A = 2$
$U_B = 1$
$U_C = 7$
$U_D = 3$
$U_E = 6$
$U_F = 5$
$U_G = 4$

$$\left.\begin{array}{l} Y_{0B} = 1 \\ Y_{BA} = 1 \\ Y_{AD} = 1 \\ Y_{DG} = 1 \\ Y_{GF} = 1 \\ Y_{FE} = 1 \\ Y_{EC} = 1 \\ Y_{C0} = 1 \end{array}\right\}$$

Cumplen con los vínculos del tipo:

$$Ui - Uj + n\ Yij \leq n-1$$

Viajante – Modelo Matemático

$$\sum_{\substack{j=0\\ i\neq j}}^{n} Yij = 1 \qquad \forall i = 0,1,2,\ldots\ldots n$$

$$\sum_{\substack{i=0\\ i\neq j}}^{n} Yij = 1 \qquad \forall j = 0,1,2,\ldots\ldots n$$

$$Ui - Uj + n\,Yij \leq n - 1 \qquad \forall i, \forall j, i\neq$$

$$Z = \sum^{n} \sum^{n} C\,ij\,Yij \quad (\text{MINIMO})$$

Dimensión de la formulación

(n+1)n	Yij variables	
n	Ui variables	
n+1	Vínculos del tipo	$\sum_{j=0;\, i\neq j}^{n} Yij = 1 \quad \forall\, i = 0,1\ldots n$
n+1	Vínculos del tipo	$\sum_{i=0;\, i\neq j}^{n} Yij = 1 \quad \forall\, j = 0,1\ldots n$
n (n-1)	Vínculos del tipo	$Ui - Uj + n\,Yij \leq n-1$

Otra forma de formularlo sin las Ui

$$\sum_{\substack{j=0\\ i\neq j}}^{n} Xij = 1 \qquad \forall i = 0,1,2,\ldots\ldots n$$

$$\sum_{\substack{i=0\\ i\neq j}}^{n} Xij = 1 \qquad \forall j = 0,1,2,\ldots\ldots n$$

$$\sum_{\substack{i,j\in M\\ i\neq j}} x_{ij} \leq |M| - 1 \qquad \forall\, M \subset N \text{ such that } \{1\} \notin M, |$$

$$\sum^{n} \sum^{n}$$

Dimensión de esta nueva formulación

(n-1)n	variables Xij
$2^n + 2n - 2$	restricciones

El número exponencial de restricciones hace impráctico resolverlo directamente.
Una opción es agregar únicamente las restricciones para evitar subtours en los casos en los cuales arma subtour y no en todos los casos

Características del problema

Una situación se modela como problema del viajante cuando:

- En el problema se plantean actividades o cosas de las cuales no se conoce el orden en el cual se realizan
- La función objetivo (directa o indirectamente) depende del orden que indique el modelo para esas actividades (a distinto orden, distinto resultado)

Variaciones al problema del Viajante

- Variables para recorrido: Yij
  - Ahora hay dos medios de transporte para ir de cada ciudad i a cada ciudad j (tren y camión). Cada medio tiene un cto. distinto para ir de i a j.
    Se debe usar por lo menos dos veces cada medio de transporte en el camino recorrido.
- Variables para orden: Ui
  - No se puede visitar al cliente de la ciudad D si antes no se visitó al de la ciudad G.
  - No se puede visitar al cliente de la ciudad F si antes no se visitó al de la ciudad E o al de la ciudad B

No olvidar...

- Leer el material adicional a esta clase en la página web (problemas de particionamiento, cobertura de conjuntos y packing)