IN47B/IN4704: Ing. de Operaciones/Gestión de Op. II
Profs: C. Bravo, M. Olivares, F. Ordoñez
Auxs: A. Echeverría, F. Lagos, F. Solari

Auxiliar Examen

Martes 6 de Septiembre de 2011

Pregunta 1

Considere un problema de un vendedor viajero que debe visitar ciudades 1,...,n, partiendo de la ciudad 0. El viaje entre la ciudad i y j tiene un costo de c_{ij} y demora un tiempo $t_{ij} > 0$. Suponga además que debe visitar cada ciudad $i \in \{1, ..., n\}$ antes de una fecha límite l_i.

1. Formule el problema de vendedor viajero que minimice el costo de viaje y satisfaga todas las restricciones de tiempo.
2. Muestre por qué se pueden eliminar subtours en este problema sin utilizar un número exponencial de restricciones.
3. Explique cómo se debe modificar el algoritmo de 2-Opt para el problema del TSP con restricciones de tiempo. Para esto, asuma que tiene el orden en que se visitan las ciudades en un arreglo `pos[i]` con $i = 1, ..., n$, donde `pos[i]` es la i-ésima ciudad visitada en el tour y que para dos ciudades i, j tiene los costos c_{ij} y tiempos t_{ij} entre i y j. (Asuma por simplicidad que ambos son simétricos i.e. $c_{ij} = c_{ji}$ y $t_{ij} = t_{ji}$).

Solución

1.

$$\begin{aligned}
\min \ & \sum_{ij} c_{ij} x_{ij} & \\
s.t. \ & \sum_{j=0}^{n} x_{ij} = 1 & i = 0, ..., n \\
& \sum_{j=0}^{n} x_{ij} = \sum_{k=0}^{n} x_{ki} & i = 0, ..., n \\
& w_i + t_{ij} - w_j \leq M(1 - x_{ij}) & i, j = 0, ..., n \\
& w_i \leq l_i & i = 1, ..., n
\end{aligned}$$

2. Suponga que tiene un subtour que visita los nodos $i_1, i_2, ..., i_n$, entonces el tiempo en que visitamos cada uno de estos nodos satisface

$$w_{i_n} \geq w_{i_{n-1}} + t_{i_{n-1} i_n} \geq w_{i_{n-2}} + t_{i_{n-2} i_{n-1}} + t_{i_{n-1} i_n} \geq w_{i_1} + \sum_{j=1}^{n-1} t_{i_j i_{j+1}}$$

   Como luego va de i_n a i_1 también satisface $w_{i_1} \geq w_{i_n} + t_{i_n i_1}$, estas dos desigualdades con el hecho que $t_{ij} > 0$ nos muestra que $w_{i_n} > w_{i_n}$, una contradicción.

3. En el algoritmo 2-Opt el paso que se ve afectado es determinar qué rutas pertenecen a la vecindad de una ruta dada. Para eso, además de recostruir la ruta, se debe verificar si el largo de la ruta es menor a l_i para todos los nodos que cambian su tiempo de llegada.

   Dada una ruta descrita por el arreglo `pos[i]`, si eliminamos los arcos en la l y k, con $l < k$ de la ruta tenemos que la ruta reconstituida desde `pos[0]` hasta `pos[l]` es la misma. Pero luego hay que verificar que $w_j \leq l_j$ para todo j en $\{$`pos[j]`$\}_{j=l}^{n}$.

Si w_i es la variable que contiene el tiempo en que se llega a `pos[i]`, entonces hay que verificar que

$$w_l + t_{pos[l]pos[k]} + \sum_{h=k}^{k-s} t_{pos[h]pos[h-1]} \leq l_{pos[k-s-1]} \qquad \text{para } k-s-1 \in 0, ..., k-l$$

y también verificar que

$$w_j + t_{pos[l]pos[k]} + t_{pos[l+1]pos[k+1]} - t_{pos[l]pos[l+1]} - t_{pos[k]pos[k+1]} \leq l_j \qquad j \in k+1, ..., n$$

El resto de la rutina es idéntica.

Pregunta 2

Considere un problema de vendedor viajero en n ciudades, partiendo de la ciudad 0, y donde el costo de ir de i a j es c_{ij}. Este problema se puede interpretar como scheduling de n trabajos que estan inicialmente todos disponibles en una máquina sin preemption ni deadlines. Para esto debe considerar que todos los trabajos se demoran un tiempo constante en la máquina pero existe un tiempo de set up al hacer el trabajo j después del i igual a c_{ij}.

1. Para este problema de scheduling, describa la heurística *Shortest Remaining Processing Time* (SRPT). ¿A qué tipo de heurística corresponde esta para el problema de vendedor viajero original?
2. Explique cómo puede utilizar la relajación con preemption de este problema de scheduling para obtener una solución factible. ¿Qué tipo de garantías tiene de esta solución?

Solución

1. La heurística SRPT escoje el trabajo que tiene el tiempo de ejecución más corto y hace ese trabajo. Como no hay preemption, solo decide el sgte trabajo una vez que el trabajo anterior ha terminado.

   Como no hay release ni due dates, en cada momento están disponibles todos los trabajos que no han sido realizados, y si el último trabajo que se hizo fue el trabajo k, entonces se ejecuta a continuación el trabajo j que tenga el menor c_{kj} entre los j que falta por ejecutar. Al inicio se escoje el trabajo j que tenga el menor c_{0j}.

   Esto corresponde a una heurística de vecino más próximo (NN) En cada etapa de la ruta visita a continuación al trabajo que se encuentra más cerca del punto actual. En extenso, visita al j que no ha sido visitado que tiene el menor c_{kj}.

2. Dado que el problema no tiene release ni due dates, no existe ningún motivo para interrumpir un trabajo. Si un trabajo tenía el menor tiempo de procesamiento y se comienza a ejecutar, durante toda su ejecución continua siendo el trabajo con menor tiempo de servicio. Note que asumo que el tiempo de servicio de un trabajo depende del último trabajo que ha sido completado. Si el último trabajo completo fue i y luego trabajo en j. Mientras quede algo de j los tiempos de procesamiento de los otros trabajos continúa siendo $c_{ik} \geq c_{ij}$. Pero cuando termina j entonces los tiempos de los otros trabajos pasan a ser c_{jk} que pueden estar ordenados differente.

   Entonces el problema con preemption obtiene la misma respuesta entera que el problema sin preemption. Es decir es factible para el problema sin preemption.

   Sabemos que la solución sin preemption es mayor el valor óptimo del schedule.

   Como esta solución corresponde a la solución que entrega la heurística de vecino más próximo en el TSP, podemos usar la garantía que nos da esa heurística: I.e. sabemos que $NN \leq 1/2(log_2(n)+1)OPT$. para problemas en los que los valores c_{ij} sean una distancia euclidiana.

Pregunta 3

Una empresa de generación eléctrica debe decidir la capacidad a instalar de dos generadores (indexados por $j = 1, 2$) con diferentes costos fijos y de operación, de forma de satisfacer la demanda de la zona. Cada día se divide en 3 partes de igual duración, indexados por $i = 1, 2, 3$. Estos corresponden a las partes del día en los cuales la demanda puede ser baja, media o peak respectivamente. Los costos fijos por unidad de capacidad del generador j es amortizada por su vida útil en una suma c_j por día. Los costos de operación del generador j durante la parte i-ésima del día es f_{ij}. Si la demanda durante la i-ésima parte del día no puede ser satisfecha, se incurre en un costo adicional g_i para suplirla. Finalmente, la capacidad de cada generador j debe ser al menos b_j.

Hay dos fuentes de incertidumbre, la demanda d_i durante cada parte del día, y la disponibilidad a_j del generador j. La demanda d_i durante el día puede tomar cuatro valores $d_{i,1}$,...,$d_{i,4}$, con probabilidad $p_{i,1}$,...,$p_{i,4}$, respectivamene. La disponibilidad del generador 1 es $a_{1,1}$,...,$a_{1,4}$, con probabilidad $q_{1,1}$,...,$q_{1,4}$, para cada uno de los eventos. Similarmente, la disponibilidad del generador 2 es $a_{2,1}$,...,$a_{2,5}$, con probabilidad $q_{2,1}$,...,$q_{2,5}$.

1. Formule un problema de programación estocástica que minimice los costos y cumpla con las restricciones.
2. Escriba el problema *Maestro* y el *Subproblema* de la descomposición de Benders para este problema.
3. Encuentre la solución incial y haga una iteración del algoritmo de Benders. Escriba el problema Maestro modificado.

Solución

1. Se utilizará la variable x_j, $j = 1, 2$ para indicar las unidades de capacidad instalada de cada generador. También se incluirá la variable y_{ij}^w para denotar los niveles de operación del generador j durante la parte i-ésima del día, bajo el escenario w. Finalmente, la variable y_i^w es la capacidad extra necesaria en el escenario w, durante la parte i del día.

$$
\begin{aligned}
\min \; & \sum_{j=1}^{2} c_j x_j + E\left[\sum_{i=1}^{3}\left(\sum_{j=1}^{2} f_{ij} y_{ij}^w + g_i y_i^w\right)\right] & \\
s.t. \; & x_j \geq b_j & \forall j \\
& y_{ij}^w \leq a_j^w x_j & \forall i, j, w \\
& \sum_{j=1}^{2} y_{ij}^w + y_i^w \geq d_i^w & \forall i, w \\
& x_j, y_{ij}^w y_i^w \geq 0 & \forall i, j, w
\end{aligned}
$$

2. La cantidad de escenarios w es $4^3 \cdot 4 \cdot 5 = 1280$. Cada *Subproblema* se ve de la siguiente forma:

$$
\begin{aligned}
\min \; & \sum_{i=1}^{3}\left(\sum_{j=1}^{2} f_{ij} y_{ij}^w + g_i y_i^w\right) & \\
s.t. \; & y_{ij}^w \leq a_j^w x_j & \forall i, j \\
& \sum_{j=1}^{2} y_{ij}^w + y_i^w \geq d_i^w & \forall i \\
& y_{ij}^w, y_i^w \geq 0 & \forall i, j
\end{aligned}
$$

Recordando que cada subproblema se resuelve considerando x_j, $j = 1, 2$, fijo.

Si utilizamos p_{ij}^w y q_i^w como las variables duales, el dual para cada w es

$$\begin{aligned}
\max \sum_{i=1}^{3}\left(\sum_{j=1}^{2} a_j^w x_j p_{ij}^w + q_i^w d_i^w\right) &= z_w \\
s.t.\ p_{ij}^w + q_i^w \leq f_{ij} & \qquad \forall i,j \\
q_i^w \leq g_i & \qquad \forall i \\
p_{ij}^w \leq 0 & \qquad \forall i,j \\
q_i^w \geq 0 & \qquad \forall i
\end{aligned}$$

El *Problema Maestro* es, por lo tanto,

$$\begin{aligned}
\min \sum_{j=1}^{2} c_j x_j + \sum_{w=1}^{W} \pi_w z_w & \\
s.t.\ x_j \geq b_j & \qquad \forall j \\
\sum_{i=1}^{3}\left(\sum_{j=1}^{2} a_j^w x_j p_{ij}^w + q_i^w d_i^w\right) \leq z_w & \qquad \forall w \\
x_j \geq 0 & \qquad \forall j
\end{aligned}$$

La segunda desigualdad puede considerar un rayo extremo, con lo cual debe imponerse que la parte izquierda de la desigualdad sea menor a 0.

3. Una posible solución inicial puede ser $x_j = b_j,\ j = 1,2$ y $z_w = -\infty,\ \forall w$. Con esto el dual para cada subproblema w es

$$\begin{aligned}
\max \sum_{i=1}^{3}\left(\sum_{j=1}^{2} a_j^w b_j p_{ij}^w + q_i^w d_i^w\right) & \\
s.t.\ p_{ij}^w + q_i^w \leq f_{ij} & \qquad \forall i,j \\
q_i^w \leq g_i & \qquad \forall i \\
p_{ij}^w \leq 0 & \qquad \forall i,j \\
q_i^w \geq 0 & \qquad \forall i
\end{aligned}$$

Considerando el óptimo p_{ij}^{w*}, $\forall i,j,w$ y q_i^{w*}, $\forall i,w$ finito, el valor z_w^* es mayor a $-\infty$ por lo que se agregan al problema *Maestro*.

$$\sum_{i=1}^{3}\left(\sum_{j=1}^{2} a_j^w x_j p_{ij}^{w*} + q_i^{w*} d_i^w\right) \leq z_w \forall w$$

Pregunta 4

Considere un grafo dirigido $G = (N, A)$ con n nodos y m arcos. El siguiente algoritmo realiza una búsqueda en profundidad en el grafo partiendo del nodo 1.

Algorithm 1: BUSQUEDA EN PROFUNDIDAD

```
VISITA(i)=0; PRED(i)=NULL ∀i ∈ N
VISITA(1)=1; PRED(1)=0;
LISTA={1}
while LISTA≠ ∅ do
    Sea i el último nodo puesto en LISTA
    if i tiene un arco (i, j) ∈ A tal que VISITA(j)=0 then
        VISITA(j)=1; PRED(j)=i;
        LISTA=LISTA∪{j};
    else
        LISTA=LISTA\{i};
```

1. Explique que es el problema de búsqueda en profundiad en un grafo, que es una instancia de este problema y cual es el tamaño de una instancia.
2. Encuentre la menor cota del número de operaciones que realiza el Algoritmo 1 en el peor caso. Considere que todas las operaciones matemáticas y de comparación, asignación, lectura, demoran lo mismo. Utilize la notación $\mathcal{O}()$.

Solución

1. El problema encuentra todos los nodos que pueden ser accesados de 1. Lo de profundida se refiere a que después que encuentra un decendiente, digamos j del nodo actual, el algoritmo busca a un decendiente de j antes de buscar a nodos al mismo nivel que j. Una instancia es un grafo dado. El tamaño de una instancia es el número de nodos y el número de arcos, n y m respectivamente.
2. La inicializacion es $\mathcal{O}(n)$. El número de iteraciones es a lo más $2n$ ya que en cada iteración se agrega o retira un nodo a LISTA y cada nodo es agregado una sola vez (cuando se revisa el arco (i, j) y VISITA(j)=0). Pero el trabajo por iteración puede cambiar ya que encontrar un arco (i, j) tal que VISITA(j)=0 puede requerir mirar todos los arcos de i. A lo largo del algoritmo, revisamos cada arco una sola vez (imaginese que los borra si los reviso en una iteración anterior). Pasamos $\mathcal{O}(m)$ revisando arcos y $\mathcal{O}(n)$ iteraciones mas para borrar nodos de LISTA. Con esto tenemos que la parte principal del algoritmo demora $\mathcal{O}(n+m)$ ya que encontrar el último nodo en LISTA y las asignaciones y evaluaciones de condiciones son a lo maás un número constante en cada iteración (= 6 por iteración). Total del algoritmo $\mathcal{O}(n+m+n) = \mathcal{O}(n+m)$ $=\mathcal{O}(m)$.

Pregunta 5

1. Dado $f, g : \mathbb{N} \to \mathbb{R}^+$, demuestre que $f(n) + g(n) = \mathcal{O}(\max f(n), g(n))$
2. Ordene en términos de crecimiento asintótico ls siguientes funciones (i.e. ordenadas según notación $\mathcal{O}(g)$). n^3, $log(n!)$, $2^{log(n)}$, $n^{3+4\sqrt{n}}$, $4^{log(n)}$, n^{73}, $n!$, $log(log(n))$, n, 2^n, $n^{log(log(n))}$, $log(n)$, 1.
3. Suponga que puede obtener sampleos X_1, ..., X_q de una variable aleatoria X de distribución desconocida. Explique como estimar el cuantil 10% de la distribución (el valor que es superado un 90% del tiempo) con una confianza del 95 %.
4. Describa los distintos elementos presentes en una cadena de suministros y como cada uno contribuye al objetivo de la cadena.

Solución

1. Se tiene que:

$$\begin{aligned} f(n) &\leq \max(f(n), g(n)) && \forall n \\ g(n) &\leq \max(f(n), g(n)) && \forall n \\ g(n) + f(n) &\leq 2 \cdot \max(f(n), g(n)) && \forall n \end{aligned}$$

   Siendo 2 una constante, se tiene el resultado por definición.

2. Una transformación útil:

$$2^{log(n)} = 2^{\frac{log_2(n)}{log_2 10}} = n^{\frac{1}{log_2 10}} = n^{log(2)}$$

   Además de la siguiente desigualdad:

$$log(n!) = \sum_{i=1}^{n} log(i) \leq nlog(n) \leq n^2$$

   Y por último para demostrar que

$$n^{3+4\sqrt{n}} \leq n^{7\sqrt{n}} \leq 2^n$$

   se considera la siguiente igualdad. Dado que $log()$ es creciente estricta:

$$\lim_{n\to\infty} \frac{7\sqrt{n}log(n)}{nlog(2)} = \lim_{n\to\infty} \frac{7log(n)}{\sqrt{n}log(2)} = 0$$

   Finalmente se tiene:

$$1 \leq log(log(n)) \leq 2^{log(n)} \leq 4^{log(n)} \leq n \leq log(n!) \leq n^3 \leq n^{73} \leq n^{log(log(n))} \leq n^{3+4\sqrt{n}} \leq 2^n \leq n!$$

3. Realizamos varias simulaciones para obtener un set de N sampleos $X1$,..., $X_{q/N}$. Luego se ordenan los X_i de menor a mayor para cada simulación, de modo de poder separar el 10% más bajo. Luego, se toman los valores críticos (cada valor que marca el 10% menor) para cada simulación; se promedian los $\sum \frac{X_{10\%}}{N}$ y se obtiene su desviación estándar empírica $S_{10\%}$.

   Para lograr un intervalo de confianza del 95%, suponemos que el promedio sigue una distribución normal (implica N mayor que 30), se necesita crear uno que multiplique por $1,96$ a la desviación. Luego se tiene

$$\left\{\sum \frac{X_{10\%}}{N} - 1,96 \cdot S_{10\%}, \sum \frac{X_{10\%}}{N} + 1,96 \cdot S_{10\%}\right\}$$

4. Esta se compone de proveedores, centros de manufactura, bodegas, centros de distribución y locales de venta.

   Además considera materias primas, inventario de productos en proceso y productos terminados que fluyen por las instalaciones.

   En general, una cadena de suministros bien articulada, busca reducir costos y tiempos de ciclo, como también mejorar la calidad del servicio, en un proceso de reparto de productos. Específicamente:

   - Los proveedores son quienes entregan productos típicamente no procesados a la cadena.
   - Los centros de manufactura se utilizan para ensamblar o terminar el producto.
   - Las bodegas son utilizadas para almacenar el producto terminado.
   - Los centros de distribución se encargan de repartir de manera inteligente el producto a los distintos locales.
   - En los locales de venta se entrega el producto al cliente.