ALGORITMO PARA LA ASIGNACION TOPOLOGICA DE REDES DE TERMINALES

Victor S. Schweikart

Universidad del Norte
Casilla 1280
Antofagasta
Chile

En el trabajo presente se analizan los algoritmos de diseño topológico de redes de terminales en un ambiente computacional multiusuario. Estos algoritmos tienen como objetivo determinar el árbol de mínimo costo que permite conectar un conjunto de terminales a un sistema multiusuario, teniendo como entradas una matriz de flujo y una matriz de costos.

La matriz de flujos establece el flujo de información que se espera tener desde cada uno de los terminales hacia el procesador. En cambio la matriz de costos establece el gasto que significa realizar esta conexión. Se supone que la conexión puede realizarse mediante concentradores o mediante líneas de múltiple encuesta (multidrop lines). Se presentan las dos classes de algoritmos: los que contemplan el costo de incorporar concentradores y otros que no lo consideran.

En particular se propone un algoritmo que permite resolver el problema de asignación topológica con la misma efectividad que el algoritmo de mayor uso en la actualidad, pero que además permite resolver exitosamente las dificuldades con que se enfrentan los algoritmos heurísticos actualmente en uso.

1. INTRODUCCION

Uno de los problemas que debe resolverse en el diseño de una red de área local es el de la conexión de un conjunto de terminales a un computador central. Esta conexión se estabelece, ya sea mediante concentradores, o bien a través de una línea común utilizando algún protocolo de acceso adecuado, como ser encuesta a los terminales o multiplexing de tiempo sincrónico o asincrónico. Finalmente también se puede realizar una combinación de las técnicas anteriores.

Una vez conocida la ubicación que tendrán los terminales y el computador central es necesario determinar de qué manera debe efectuarse la conexión, a modo de optimizar la asignación de los recursos. Es posible llevar a efecto esta asignación en forma óptima, si se pueden estimar las siguientes cantidades:

- el costo asociado a la conexión de cada uno de los terminales al computador central o a un concentrador.
- la capacidad máxima del canal de transmisión.
- el flujo de información que se espera en cada uno de los terminales.
- el costo asociado a los concentradores.

El costo asociado a la conexión de cada uno de los terminales contempla los siguientes aspectos: costos de instalación del cable y el costo del cable mismo.

Al especificar la capacidad máxima del canal de transmisión se toma en consideración la tasa de transmisión que estará presente en el canal. Esta cantidad debe adecuarse convenientemente para que pueda compararse con el flujo de información presente en cada terminal.

En lo que se refiere a estimar el flujo de información generado por cada terminal, es evidente que se trata de una de las cantidades menos precisas, por cuanto tiene relación con la actividad que se desarrolla en el punto donde se instala el terminal, y la realidad dentro de la cual se inserta el sistema computacional. Además, se trata de un fenómeno que no es constante en el tiempo, presentando variaciones que van desde su instalación a un instante posterior, como también variaciones semanales, mensuales y anuales, de acuerdo a la actividade desarrollada.

A modo de ejemplo, en ambientes industriales es probable que la frecuencia de uso de un terminal sea relativamente constante, con algunas variaciónes menores. En ambientes universitarios, en cambio, el uso de terminales depende del periodo de actividades. En ambientes de oficina, la frequencia de uso de un terminal está sujeta a las variaciones de las actividades económicas.

Finalmente, en lo que se refiere a costo de concentradores, el problema de asignación de terminales se complica, por cuanto deben estudiarse las alternativas de:

- conectar el terminal directamente al computador central
- conectar el terminal al computador central, via línea multidrop

- conectar el terminal al computador, previo paso por concentrador.

Normalmente se tiene que resolver el problema en forma paramétrica, considerando la incorporación progresiva de concentradores hasta ubicar la solución factible de mínimo costo.

El problema básico que subyace en este análisis es la conexión de los terminales vía línea multidrop a un concentrador o computador. Este es el tema central que trata este trabajo y corresponde al diseño topológico de una red de terminales. Resolviendo este problema, y aplicándolo en forma iterativa para la red completa considerando la inclusión de otros concentradores, se obtiene la asignación óptima |1|, |8|.

La configuración final que adopta dicha solución es semejante a un árbol, en cuya raíz se encuentra el computador central y que conecta los terminales a dicho computador vía las ramas. (Fig. 1).

Dentro del diseño topológico de esta red se encuentra el problema de determinar como serán conectados los terminales a los concentradores. A este problema se le llama "problema de la distribución de terminales" y se le puede formular de la siguinte forma |1|:

Supóngase que se tiene n terminales Ti, i=1 ...n, en

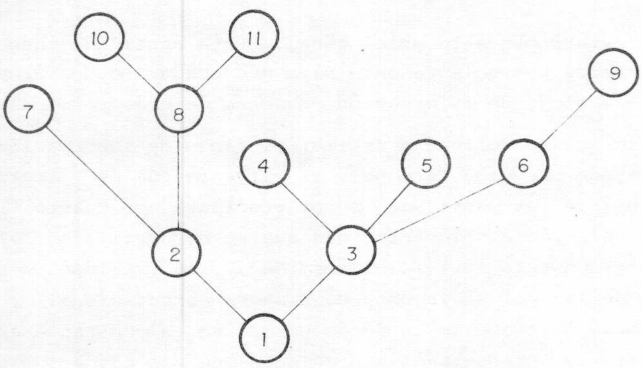

Fig. 1: Red de terminales

lugares conocidos y sólo un sitio central TØ que también es conocido. Aún más, supóngase que se conocen los costos (usualmente costos de línea proporcionales a la distancia y capacidad) de establece las conexiones entre los terminales y desde los terminales al lugar central (para la mayoria de los algoritmos que solucionan el problema se puede considerar este lugar central como un terminal). El problema es encontrar un árbol que conecte todos los terminales con el mínimo costo. Este costo se define como la suma de los costos de cada rama usada en la conexión. A este árbol se le denomina Arbol de Mínima Extensión (minimum spanning tree MST) y existen algoritmos óptimos y eficientes para encontrarlo.

Para operar la red de terminales conectados en forma de árbol se utiliza un esquema de muestreo o "polling". El enlace que conecta a los terminales conectados de forma multipunto al lugar central debe ser capaz de suportar el tráfico generado por los terminales en la línea multipunto. Debido a que el enlace posee una capacidad limitada, ésta limita la cantidad de terminales que se pueden colgar en una línea multipunto. Se supondrá que esta restrición de tráfico es igual para todas las líneas multipunto que se conectan al lugar central. Esto lleva finalmente a que el problema a resolver es encontrar un MST conrestricciones de tráfico en las ramas del árbol. Este MST con restrición es más difícil de resolver, existiendo varios algoritmos óptimos para problemas pequeños, donde el número de terminales no es muy elevado. Además se han desarrollado muchos algoritmos heurísticos para resolver problemas de mayor envergadura.

Los algoritmos óptimos |7| dan solución exacta, pero su problema es que son muy ineficientes en el uso de recursos computacionales (tiempo UPC y memoria). Por otro lado, los algoritmos heurísticos son más eficientes. Algunos generan soluciones muy cercanas al óptimo. El problema con estos algoritmos es que son sensibles a posibles "patologías topológicas" de la red, generando soluciones lejos del óptimo en estos casos.

En este trabajo se indicarán los algoritmos que actualmente se usan para resolver el problema. Luego se indicará un nuevo algoritmo deducido en base a los anteriores, se verá que este algoritmo mejora alguna de las cualidades de los algoritmos existentes.

2. ALGORITMOS ACTUALES

Consideraciones

Desde el punto de vista de eficiencia, ya se estableció la conveniencia del uso de los algoritmos heurísticos. De entre estos, se analizarán los algoritmos de Essau-Williams y Prim. Conviene destacar, sin embargo que la forma del algoritmo de Prim que se usará aqui difiere de las que se analizan en |4| y |5| en el sentido que se agrega la restricción de satisfacer un flujo máximo. En efecto, en el estabelecimiento de estos algoritmos se hace la suposición de que la restriccion de tráfico será la misma para todas las líneas, y la llamaremos flujo máximo (en un caso más general, cada línea puede tener distinta restricción).

La restricción de flujo máximo no es la única que se puede tener. De hecho se pueden incorporar restricciones tales como tiempo de respuesta, pero en este caso el problema se hace más difícil de resolver.

Bajo estas consideraciones se puede plantear el problema a resolver de la siguinte forma:

Dados:

- Un conjunto de nodos (terminales) y un nodo central al que se denomina nodo 1.
- Una matriz de costo de conexión Cij, donde Cij represente el costo de tender una línea entre el nodo i y el nodo j.

- Tráfico generado por cada terminal
- Restricción de tráfico máximo en las líneas de conexión.

Encontrar un árbol de mínima extensión con restricciones daua por el tráfico máximo en las líneas, para conectar todos los nodos al nodo 1.

Algoritmos

A continuación se plantas el algoritmo de Essau-Williams y el de Prim, después se incluy un ejemplo. El planteamiento corresponde al dado por Schwartz en |2|.

- Algoritmo de Essau-Williams

Ø. Inicie calculando la matriz de "compromiso" (trade-off) a partir de la relación.

$$Tij = Cij - Cil \qquad (1)$$

para todo i,j. Siendo Cij la entrada apropiada de la matriz de costo. Este parámetro mide así el costo entre conectar el nodo i al nodo j, y el nodo i directamente al centro. Es una medida del ahorro en conectar el nodo i al nodo j en lugar de conectar el nodo i al centro. (Fijarse que (1) corresponde a restar a todas las columnas de la matriz Cij la primera columna).

1. Seleccione el mínimo Tij. Denomine al (i,j) encontrado (i*,j*). (Busca la conexión de mayor ahorro en costo).
2. Chequee si las restricciones son satisfechas. Si lo son, vaya al paso 3. Si no, ponga Ti*j* = infinito, vuelva a 1.
3. Agregue enlace (i*,j*). Borre fila i* de la matriz. Etiquete nodo i* con la etiqueta del nodo j* para mostrar que i* está conectado con j*, reevalúe las restricciones. Borre Ti*j*. Vaya a 1.

- Algoritmos de Prim

Ø. Inicie asignandc a cada nodo un factor de peso Wi, con Wi=Ø; Wi=-infinito, para toda i≠1. La función de desventaja Tij está definida como Tij=Cij-Wi. Inicialmente todos los Tij=infinito, excepto tij (sólo queda primera fila de la matriz Cij).

1. Encontrar Ti*j* mínimo. (Aquí se ve que el algoritmo busca el costo mínimo de conectar un nuevo nodo a alguno que ya está en la red).
2. Chequee las restricciones. Si son violadas, ponga Ti*j*=infinito, vuelva a 1.
3. Agregue enlace (i*,j*), ponga Wj*=Ø, reajuste las restricciones. Calcule nuevamente los Tij (este cálculo es fácil, ya que equivale a copiar la fila j* de la matriz de costo Cij en la fila j* de la matriz Tij. El resto de los Tij siguen iguales). Borre Ti*j* y Tj*.

Ejemplo

El ejemplo corresponde a la siguinte situación:

Cij=

Nodo	1	2	3	4	5
1	-	3	3	5	10
2	3	-	6	4	8
3	3	6	-	3	5
4	5	4	3	-	7
5	10	8	5	7	-

Flujo por nodo

Nodo	2	3	4	5
Flujo	2	3	2	1

Flujo máximo: 5 unidades.

La distribución de los nodos es la seguinte:

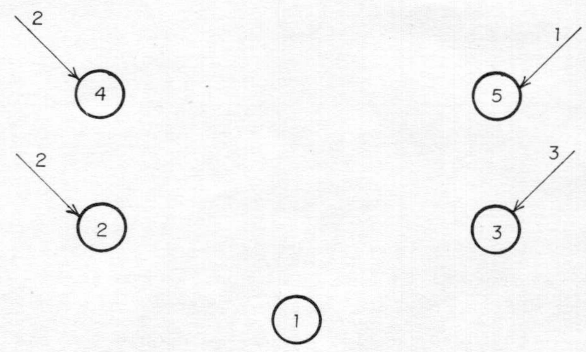

Fig. 2: Distribución de nodos

El algoritmo Essau-Williams comienza calculando la matriz Tij, la cual queda:

Nodo	1	2	3	4	5
1	-	-	-	-	-
2	Ø	-	3	1	5
3	Ø	3	-	Ø	2
4	Ø	-1	-2	-	2
5	Ø	-2	-5	-3	-

A continuación elige como mínimo Tij a T53. Las restricciones de flujo se cumplen, así es que conecta 5 a 3. Se borra T53 y T35. Se indica que 5 está conectado a 3 (no toma en cuenta la fila 5 para las siguintes búsquedas del mínimo). Luego se prosigue la búsqueda: selecciona T43, pero se viola restricción de flujo, en consecuencia se borran T43 y T34. Después selecciona T42, se cumplen restricciones de flujo y no se forma lazo, así es que se conecta 4 con 2. Se borran T42 y T24. Finalmente la red queda de la siguiente forma:

El algoritmo de Prim comienza con la siguinte matriz Tij (los * simbolizan infinito):

Node	1	2	3	4	5
1	-	3	3	5	10
2	*	*	*	*	*
3	*	*	*	*	*
4	*	*	*	*	*
5	*	*	*	*	*

A medida que se agrega un nodo a la red va apareciendo su fila de costos en la matriz Tij.

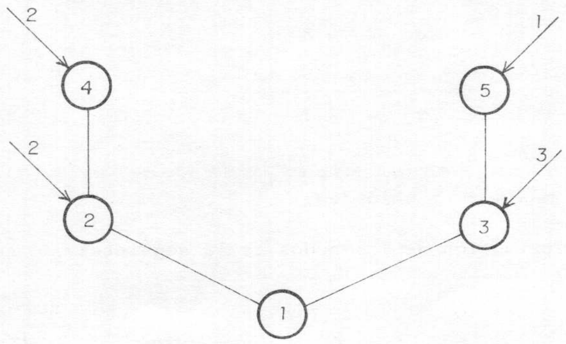

Fig. 3: Solución según E-W.

Su costo es de 15 unidades.

El algoritmo Prim primero elige T12. Se cumplen las restricciones así es que une 1 con 2. Aparece la fila 2 de la matriz de costo en la matriz Tij. Se borra T12. A continuación T13, ya que se cumple la restricción de flujo se une 1 con 3. Aparece la fila 3 y se borra T13. Después elige T34. Se cumple la restricción y no se forma lazo, luego une 3 con 4. Luego selecciona T34, pero formaría un lazo; en consecuencia no se unen y se borra T24. Finalmente selecciona T35. Se cumple la restricción de flujo, se une 3 con 5 y el algoritmo finaliza. La red resultante es:

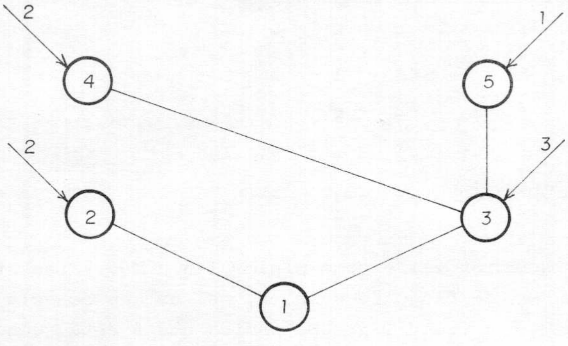

Fig. 4: Solución según Prim

Su costo es de 17 unidades.

Análisis

De los ejemplos se puede ver que las soluciones para un mismo problema varían debido a la naturaleza heurística de los algoritmos. Estas variaciones se deben a los distintos criterios que se usan para ir seleccionando los nodos a agregar a la red. Para el algoritmo Essau-Williams (E-W) el criterio es ir formando conglomerados de nodos desde los nodos más lejanos al nodo 1 (UCP) hacia los más cercamos |2|; en el momento que se detecta que ya no se puede hacer más ahorro en las conexiones entre nodos se conectan los nodos restantes al nodo 1.

En el caso del algoritmo Prim el criterio es ir agregando de un solo nodo a la vez |2|, empezando desde los nodos más cercamos al nodo 1.

Se pudo comprobar que el criterio E-W es el más efectivo en cuanto a minimización de costos (se implementaron los algoritmos y se corrieron los programas com otro tipo de redes, ver |3|).

Además se pude observar |3| que el algoritmo E-W no encuentra una solución muy buena cuando los nodos están agrupados em varios grupos cercanos entre sí. Esto hace que establezca las conexiones dentro de los grupos, no siendo éste el mejor criterio a veces.

Por otro lado, el algoritmo Prim tiene la falla que al preocuparse de conectar primero los nodos más cercanos al nodo 1, hace que estos nodos saturen luego las lineas, teniendo que establecer una línea larga (y por lo tanto de gran costo) entre los nodos más lejanos al nodo 1.

La literatura (|1|, |2|, |4|, |5| y |6|) considera además otros algoritmos, como ser el algoritmo de Kruskal y Kershenbaum. Si se considera el primero, debe tenerse especial cuidado en utilizarlo con la restricción de flujo máximo, restricción que no siempre figura en la descripción (|4| y |5|). El algoritmo de Kershenbaum es una generalización de los algoritmos existentes, de modo tal que adeccuando ciertos parámetros en la expresión que conduce al cálculo de la matriz de compromiso se obtienen los otros algoritmos |6|.

Sien embargo, el algoritmo de Kruskal produce generalmente soluciones de mayor costo que las resultantes por la aplicación de los algoritmos presentados. En cambio, el algoritmo de Kershenbaum adolece de la dificultad de que deben estimarse los parámetros adecuados a una solución óptima.

3. ALGORITMO PROPUESTO

Algoritmo

Al concebir un nuevo algoritmo la intención es lograr subsanar algunas de las fallas del algoritmo E-W. Se hace la comparación con este algoritmo, ya que es el que tiene el mejor funcionamiento.

Una observación importante, que ayudó a concebir el algoritmo que se explicará más adelante, es que, en el algoritmo Prim no es forzoso comenzar a conectar nodos emprezando por el nodo 1 sino que este algoritmo se puede comenzar desde cualquier nodo de la red. Se hicieron pruebas con este algoritmo en las cuales, para una misma red, se cambiaba el nodo de comienzo del algoritmo. Se vio que se lograba una reducción de costo, pero no una reducción notable de tiempo de procesamiento. Las soluciones encontradas eran mejores que las de E-W algunas veces, pero en general se mantenían con un costo superior. El hecho importante era que al comenzar el algoritmo Prim por uno de los nodos más alejados del nodo 1 se lograba la reducción en costo más significativa, llegando algunas veces a ser mejor que el algoritmo E-W.

A partir de esta idea se concibió el seguinte algoritmo:

1. Elegir el nodo más alejado del centro. (Se retiene la idea de E-W de empezar por nodos más alejados del nodo 1).
2. Se conecta este nodo con su vecino más cercano en dirección al nodo 1 (es decir, el vecino debe estar más cerca del nodo 1).
3. Si el nodo vecino ya está en la red o es el nodo 1, vaya a 5. Si no, vaya a 4.
4. Se repite el paso 2 pasando el nodo vecino a ser el nodo más lejano (retiene la idea de Prim de conectar los nodos más cercanos al centro primero).
5. Si faltan nodos por conectar, se va a 1. Si no, termina el algoritmo.

Como se puede ver, este algoritmo hace una combinación de los criterior usados por los algoritmos de E-W y Prim. Se podría decir que establece un compromiso entre los dos criterios.

A continuación se efectuará el algoritmo para el ejemplo propuesto en el punto anterior:

El algoritmo elige primero el nodo 5 como el nodo más alejado del centro. Luego el nodo más cercano al nodo 5 en dirección al centro es el nodo 3. Ya que se cumple restricción de flujo une 5 con 3. Luego busca el nodo más cercano al nodo 3 en direción al centro. Este resulta ser el nodo 1. Efectúa la conexión y, como ya se llegó al centro, procede a buscar otro nodo más alejado. Este resulta ser el nodo 4. Ahora busca el nodo más cercano al nodo 4 en dirección al centro. Resulta ser el nodo 3. Pero se viola restricción de flujo, así es que se busca otro nodo, éste resulta ser el nodo 2. Se cumple la restricción así es que une 4 con 2 y ahora procede a buscar el nodo más cercano al nodo 2 en dirección al centro. Este resulta ser el nodo 1. Como ya se conectaron todos los nodos el algoritmo termina con la siguiente red:

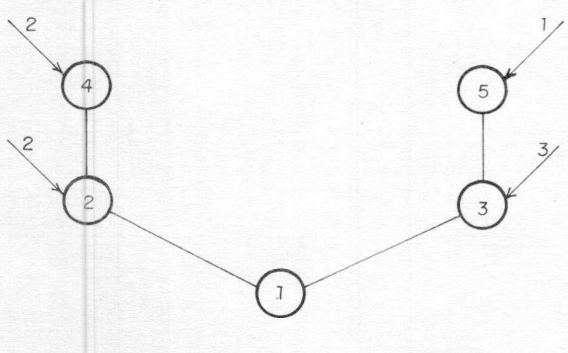

Fig. 5: Solución según G-L.

Su costo es de 15 unidades.

Análisis

Se probó el algoritmo propuesto para diversas configuraciones (matrices de costo) |3|. Se pudo deducir que el funcionamiento del algoritmo en cuesto a minimización de costo es tan bueno (a veces mejor) que el resultado de E-W y el resultado de Prim. El algoritmo en sí fue diseñado de forma de subsanar los defectos del E-W así es que su funcionamiento será claramente mejor que el E-W en aquellos casos en que éste falle.

Se comprobó que el costo mínimo alcanzado mediante la aplicación del presente algoritmo se encuentra por debajo del funcionamiento del de E-W, en aquellos casos en que una gran cantidad de nodos se encuentran agrupados lejos del nodo central (UCP). |3|.

Otro aspecto interesante es que el algoritmo propuesto es mucho más rápido que el E-W y que el algoritmo de Prim. Esta afirmación está basada en observaciones empíricas, ya que un análisis de los algoritmos implementados en |3| según |4| revela que los tres algoritmos tienen una cota superior para el tiempo de ejecución, dada por $0(n^3)$, siendo n el número de nodos.

En particular, el algoritmo propuesto tiende a ser más rápido que el E-W dado que efectúa una búsqueda en profundidad con cada uno de los nodos. En cambio, el algoritmo de E-W efectúa conexiones en la periferia antes de conectar al nodo central, lo cual obliga a recorrer la llsta completa de nodos aún no conectados en cada búsqueda.

4. CONCLUSIONES

Se puede concluir que el algoritmo propuesto de una solución de costo tan razonable como la del E-W pero en un tiempo menor. Además permite resolver situaciones que resultan conflictivas para el criterio de E-W, como ser aglomeraciones de nodos que resultan en costos similares o idénticos. En consecuencia, la idea desarrollada representa una contribución sustancial a la resolución de este tipo de problemas.

5. REFERENCIAS

|1| Boorstyn, Robert; Frank, Howard, "Large-Scale Network Topological Optimization". IEEE Transactions on Communications Vol. Com-25, Número 1, Enero 1977

|2| Schwartz, Mischa. ECentralized Network Design. Multipoint Conecctions". En "Computer Communication Network Design & Analysis"; Prentice-Hall, Inc. 1977.

|3| Llanos, Flavio, "Diseño topológico de redes de terminales". Trabajo de titulación Universidad Técnica Federico Santa María, 1985.

|4| Aho, A; Hoperoft, Uilman, J.: "Data Structures and Algorithms"; Addison Wesley, 1983.

|5| Hu, T.C. "Combinatorial Algorithms"; Addison 1982.

|6| Kershenbaum, A; Chou, W.: "Aunifield algoritm for designing multidrop Teleprocessing Networks. IEEE

Trans. on Communication Vol. Com-22, 11, Nov. 1974.

|7| Chandy, K.; Russel, R.: "The Design of Multipoint Linkages in a Teleprocessing Tree Network". IEEE Trans. on Comp. Vol. C-21, 10, Oct. 1972.

|8| Tannenbaum, A.: "Computer Networks", Prentice-Hall, Inc. 1981.