3. Modelo de minimización de redes

Pienso que nunca veré un poema
tan querido como un árbol.
Joyce Kilmer

El modelo de minimización de redes o problema del *árbol de expansión mínima* (Minimum Spanning Tree) tiene que ver con la determinación de los ramales que pueden unir todos los nodos de una red, tales que, minimicen la suma de las longitudes o costos de los ramales escogidos. No se deben incluir ciclos en al solución del problema.

Para crear el árbol de mínima expansión se deben cumplir las siguientes características:

1. Se tienen los nodos de una red pero no las ligaduras. En su lugar, se proporcionan las ligaduras potenciales y la longitud positiva para cada una si se inserta en la red. (Las medidas alternativas para la longitud de una ligadura incluyen distancia, costo y tiempo.)

2. Se desea diseñar la red con suficientes ligaduras para satisfacer el requisito de que haya un camino entre cada par de nodos.

3. El objetivo es satisfacer el requisito anterior de manera que se minimice la longitud total de las ligaduras insertadas en la red.

Por otra parte, una red con n nodos requiere sólo (n-1) ligaduras para proporcionar una trayectoria entre cada par de nodos. Las (n-1) ligaduras deben elegirse de tal manera que la red resultante formen un árbol de expansión.

Por lo tanto, el problema consiste en, dado un grafo $G = (N, A)$ con $n = |N|$ nodos y $m = |A|$ arcos con una longitud o coste c_{ij} asociado a cada arco $(i, j) \in A$, encontrar un árbol extensión, llamado árbol de expansión minima, que tiene un coste total en manos de sus arcos constituyentes, es decir, medido como la suma de costes de sus arcos en el árbol de extensión.

3.1. Ejemplo introductorio

El tipo de problema que queremos resolver es:

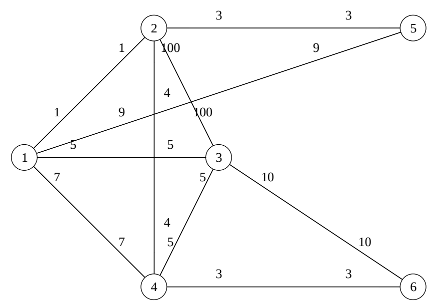
Dada una red de arcos no dirigidos dimensionados (i.e. en Kms), se trata de conectar todos los nodos directa o indirectamentea través de otros nodos.

Ejemplo 3.1 *La empresa Pau Pi Cable TV S.A., quiere conectar cinco barrios desde su central con fibra óptica para vender servicios de TV. La red del sistema de cable se resume en la tabla siguiente:*

	Nodo 1	*Nodo 2*	*Nodo 3*	*Nodo 4*	*Nodo 5*	*Nodo 6*
Nodo 1		*1*	*5*	*7*	*9*	
Nodo 2	*1*		*100*	*4*	*3*	
Nodo 3	*5*	*100*		*5*		*10*
Nodo 4	*7*	*4*	*5*			*3*
Nodo 5	*9*	*3*				
Nodo 6			*10*	*3*		

donde, el nodo 1 representa la estación de TV de la empresa y los números asociados a cada arco son la cantidad en Kms de cable necesario para conectar la central al barrio o entre los barrios. Tenga en cuenta que a más cable utilizado más costo de instalación. Un arco faltante indica que el costo es prohibitivo o imposible de realizar. Se necesita determinar los enlaces que dará lugar a un uso de cable mínimo a la vez que se garantiza que todas la áreas queden conectadas directa o indirectamente a la estación proveedora de los servicios.

Escrito en forma de grafo, resulta



Dado que cada arco tiene una dimensión, nosotros debemos encontrar una red de extensión mínima que nos proporciona la conexión **eficiente** entre todos los nodos de la red con la suma de las dimensiones mínima.

Observación 3.2 *En el modelo tenemos en cuenta que cada arco tiene una dimensión y podemos asumir que el coste varía linealmente con la cantidad de flujo.*

3.2. Formulación en forma de un programa lineal

Sea el grafo $G = (N, A)$ y $A(S)$ conjunto de arcos contenidos en el subgrafo S. el problema del árbol de expansión mínima se modela con el siguiente programa lineal

$$\begin{array}{ll} \text{mín} & \sum\limits_{(i,j)\in A} c_{ij}x_{ij} \\ \text{sujeta a:} & \sum\limits_{(i,j)\in A} x_{ij} = n-1, \\ & \sum\limits_{(i,j)\in A(S)} x_{ij} \leq |S| - 1, \quad \text{para todo conjunto } S \text{ de nodos.} \\ & x_{ij} \in \{0,1\}. \end{array}$$

La variable $x_{ij} \in \{0,1\}$ indica cuando seleccionamos el arco (i,j).

La primera restricción es una restricción de cardinalidad, indicándonos que nosotros elegimos exactamente $n-1$ arcos. La segunda restricción nos dice que el conjunto de arcos elegidos no contiene ciclos (si alguna solución fuese seleccionada y contuviese un ciclo y S fuese el conjunto de nodos del ciclo elegido, la solución violaría la restricción).

Notese que el número de restricciones crece exponencialmente dado que su número es función del número de nodos.

3.3. Caracterización del óptimo

Caracterización externa: Condición de optimalidad de corte.

Teorema 3.3 *Un árbol de extensión T^* es uno de mínima extensión si y sólo si satisface:*

Para cada arco $(i, j) \in T^*$ ocurre que $c_{ij} \leq c_{kl}$ para cada arco (k, l) contenido en el corte formado eliminando (i, j) en T^*.

Esta condición de optimalidad implica que cada arco en el Minimum Spanning Tree es un arco de mínimo coste a lo largo del corte que está definido eliminando el arco (i, j). Nos habla de la relación existente entre un arco en el árbol y muchos arcos fuera del arco, esto es, aquellos en el corte que producimos eliminando el arco del árbol.

Caracterización interna: Condición de optimalidad de camino.

Teorema 3.4 *Un árbol de extensión T^* es uno de mínima extensión si y sólo si satisface:*

Para cada arco no del árbol $(k, l) \in G$ tenemos que $c_{ij} \leq c_{kl}$ para cada arco (i, j) contenido en el camino en T^* que conecta los nodos k y l.

3.4. Algoritmos de Kruskal

Este algoritmo aparece por primera vez en Proceedings of American Mathematical Society, pp 48-50 in 1956. Este algoritmo fue publicado inicialmente por Otakar Borůvka (1926)

Los arcos deben tener diferentes valores.

Este algoritmo realiza repetidamente estas dos operaciones básicas:

1. El vecino más cercano (N_k, i_k, j_k).

Esta operación toma como entrada un arbol de expansión de nodos N_k y determina un arco (i_k, j_k) con el coste mínimo entre todos los arcos que emanan de N_k.

i.e. $c_{i_k, j_k} = \text{mín} \{c_{ij} \mid (i, j) \in A, i \in N_k, j \notin N_k\}$.

Para llevar a cabo esta operación necesitamos mirar todos los arcos en la lista de los nodos adjacentes en N_k y encontrar un arco de coste mínimo entre aquellos arcos que tienen un final no perteneciente a N_k.

2. Unimos (i_k, j_k).

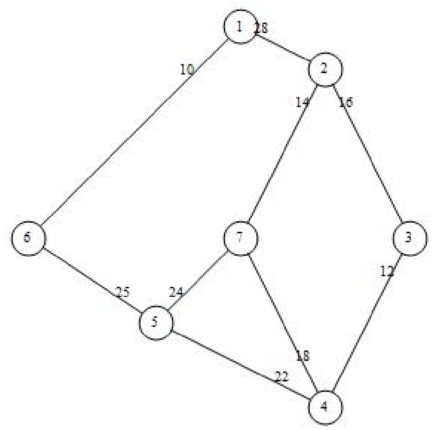
Esta operación toma como input dos nodos i_k, j_k y si los dos nodos pertenecen a dos diferentes árboles, entonces miramos esos dos árboles como un sólo árbol.

3.4.1. Ejemplo

Resolvemos el ejemplo que viene dado por la siguiente tabla de grafo:

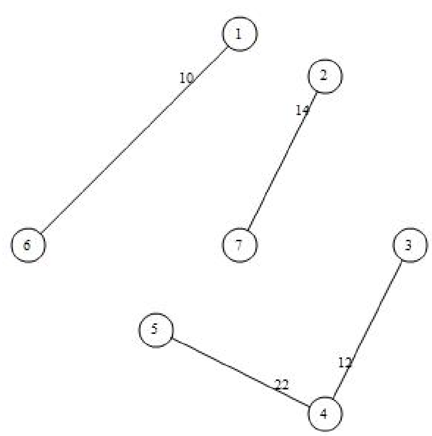
	1	2	3	4	5	6	7
1		28				10	
2			16				14
3				12			
4					22		18
5						25	24
6							
7							

El gráfico correspondiente es:



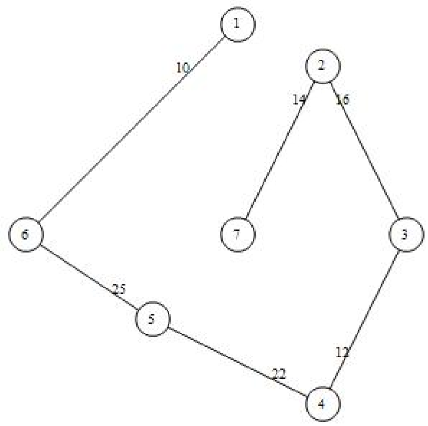
Paso 1. Formamos árboles de expansión a partir de los nodos eligiendo el arco de menor coste.

Tenemos los conjuntos (N_k, i_k, j_k) siguientes:
$(\{1,6\},(1,6))$, $(\{2,7\},(2,7))$ y $(\{3,4,5\},(3,4),(4,5))$



Unimos $(\{2,7\},(2,7))$ y $(\{3,4,5\},(3,4),(4,5))$ con el arco $(2,3)$ y tenemos $(\{1,6\},(1,6))$ y $(\{2,3,4,5,7\},(3,4),(4,5),(2,7),(2,3))$

volvemos a unir los dos áboles de expansión mediante el arco $(5,6)$, resultando



Que es un ábol de expansión $\{(1,6),(6,5),(5,4),(4,3),(3,2),(2,1)\}$ de valor 101.

3.4.2. Algoritmo de Kruskal modificado

Tenemos que construir un árbol T^* (subgrafo sin ciclos) formado por arcos seleccionados de forma que su pesos sea mínimo de entre los que podemos elegir en ese momento. Siempre que se añada un arco $(i,j) \in A$, éste, será siempre la conexión más corta (menos coste) desde el nodo i al resto del grafo G. Puede existir más de una solución especialmente cuando los costes de los arcos se repiten.

Paso 1. Se seleccionan los arcos de mínimo coste con el mínimo coste que no pertenezcan al árbol T^*.

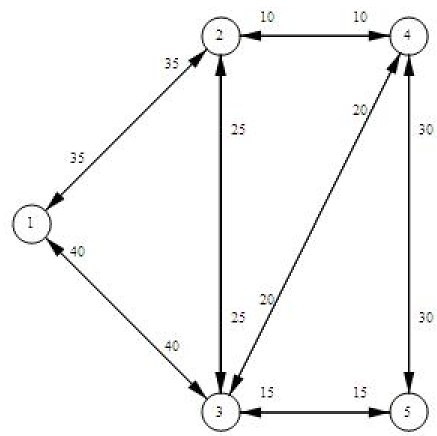
Paso 2. Si no forman ciclo, continua. Si algún arco forma un ciclo, se elimina.

Paso 3. Se añaden al árbol T^*. Si el número de arcos del árbol T^* es $n-1$, se **acaba**. En caso contrario vamos al Paso 1.

Empates: los empates para arcos distintos (Paso 1), se pueden romper en forma arbitraria y el algoritmo debe llegar a una solución optima. No obstante, estos empates son señal de que pueden existir (pero no necesariamente) soluciones optimas múltiples. Todas esas soluciones se pueden identificar si se trabaja con las demás formas de romper los empates hasta el final.

3.4.3. Ejemplo

Supongamos que en la red que se muestra en la figura, los nodos son centros de consumo eléctrico y, los números en los arcos son distancias en kilómetros. Se trata de encontrar el árbol que, con una longitud mínima, una todos los nodos.



Iter.	Coste	Arco selec	ciclos	T^*
1	10	$(2,4)$	no	$\{(2,4)\}$
2	15	$(3,5)$	no	$\{(2,4),(3,5)\}$
3	20	$(2,3)$ $(4,5)$ $(4,3)$	si si no	$\{(2,4),(3,5),(4,3)\}$
5	35	$(1,2)$ $(1,3)$	no si	$\{(2,4),(3,5),(4,3),(1,2)\}$

Luego el resultado es $\{(1,2),(2,4),(4,3),(3,5)\}$ y el coste $35+10+20+15 = 80$.

3.5. Algoritmo de Prim

El algoritmo fue desarrollado independientemente uno del otro, por Vojtěg Jarnik (1930) y Robert C. Prim (1957) Shortest Connection Networks And Some Generalizations.Bell System Technical Journal 36, 6. Fue redescubieto por Dijkstra (1959).

3.5.1. Algoritmo

Sea el grafo $G = (N, A)$ y c_{ij} el coste asociado con cada arco $(i, j) \in A$.

Paso 1. Elegimos un nodo arbitrariamente y lo apuntamos en el conjunto N_{nuevo}.

Paso 2. Elegimos de los arcos salientes del conjunto N_{nuevo} el de valor más pequeño. Añadimos el nuevo nodo al conjunto N_{nuevo} y nos quedamos con el arco correspondiente.

Paso 3. Si en número de nodos es N paramos el proceso y si es menor que N vamos a Paso 2.

3.5.2. Ejemplo

Resolvemos el mismo ejemplo que el de Kruskal.

Paso 1. Elegimos un nodo inicial. El nodo 1. $N_{nuevo} = \{1\}$.
Paso 2. $\min_{\text{valores de los arcos salientes}} \{35, 40\} = 35$.
Añadimos el Nodo 2. $N_{nuevo} = \{1, 2\}$. Añadimos el Arco $(1, 2)$.

Camino: $\{(1, 2)\}$.
Paso 2. $\min_{\text{valores de los arcos salientes}} \{25, 40, 10\} = 10$.
Añadimos el Nodo 4. $N_{nuevo} = \{1, 2, 4\}$. Añadimos el Arco $(2, 4)$.

Camino: $\{(1, 2), (2, 4)\}$.
Paso 2. $\min_{\text{valores de los arcos salientes}} \{25, 40, 20, 30\} = 20$.
Añadimos el Nodo 3. $N_{nuevo} = \{1, 2, 4, 3\}$. Añadimos el Arco $(4, 3)$.

Camino: $\{(1, 2), (2, 4), (4, 3)\}$.
Paso 2. $\min_{\text{valores de los arcos salientes}} \{30, 15\} = 15$.
Añadimos el Arco $(3, 5)$.
Añadimos el Nodo 5. $N_{nuevo} = \{1, 2, 4, 3, 5\}$.

Camino: $\{(1, 2), (2, 4), (4, 3), (3, 5)\}$.

3.5.3. Problemas propuestos

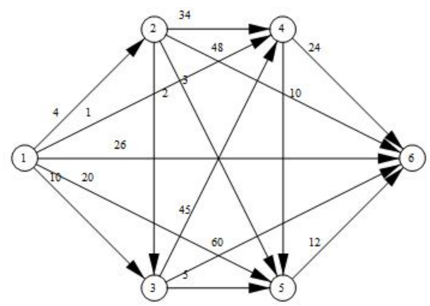
Problema 3.5 *Optimal message passing (Prim,1957). Seis agentes secretos tienen establecido un sistema para el intercambio de información. La frecuencia de encuentros no es la misma para todos y, por motivo de seguridad, tampoco todos hablan con todos.La siguiente tabla resume el tiempo transcurrido, en horas, entre las conversaciones para cada dos agentes.*

	001	*002*	*003*	*004*	*005*	*006*
001		*4*	*10*	*1*	*20*	*26*
002			*2*	*34*	*3*	*48*
003				*45*	*5*	*60*
004					*10*	*24*
005						*12*
006						

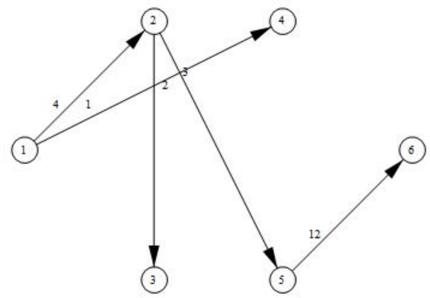
¿Cuánto tiempo transcurre como mínimo desde que un agente cualquiera tiene una información y la conocen todos?.

Si 004 *tiene una información, ¿cuánto tiempo tardarán en conocerla todos?*

El gráfico correspondiente es:



La solución es



Por consiguiente, el tiempo mínimo de comunicarse es de 22 horas. Salga de donde salga el mensaje dado que no tienen que ser sincronizados.

3.6. Algoritmo de Sollin

Nosotros podmos ver este algoritmo como una versión híbrida de los algoritmos de Kruskal y Prim.

El algoritmo del árbol de extensión mínima necesita comenzar con *cualquier* nodo y conectar éste con el más *cercano* en la red.

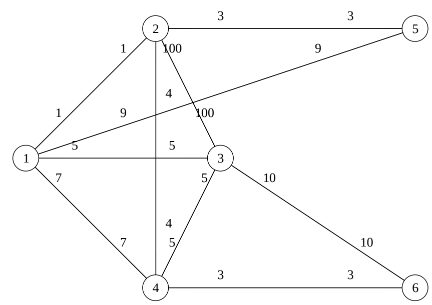
Los nodos resultantes constituyen un conjunto que llamamos *conjunto conectado* y los nodos restantes constituyen el *conjunto no conectado*.

A continuación, se escoge un nodo del conjunto no conectado que sea el más cercano a *cualquier* nodo del conjunto conectado. El nodo escogido se elimina del conjunto no conectado y se incorpora al conjunto conectado.

El proceso se repite hasta que el conjunto no conectado queda vacío.

3.6.1. Solución del ejemplo 3.1

El grafo no dirigido que nos representa el problema es:



Aplicando el algoritmo comenzando por el Nodo 1.

Iteración 0:

Sea $C = \{1\}$ el conjunto de nodos conectados.
Sea $NC = \{2, 3, 4, 5, 6\}$ el conjunto de nodos no conectados.
Sea $A = \{\varnothing\}$ el conjunto de arcos solución.

Iteración 1:

$C = \{1, 2\}$ y $NC = \{3, 4, 5, 6\}$.
$A = \{(1, 2)\}$.

Iteración 2:

$C = \{1, 2, 5\}$ y $NC = \{3, 4, 6\}$.
$A = \{(1, 2), (2, 5)\}$.

Iteración 3:

$C = \{1, 2, 4, 5\}$ y $NC = \{3, 6\}$.
$A = \{(1, 2), (2, 5), (2, 4)\}$.

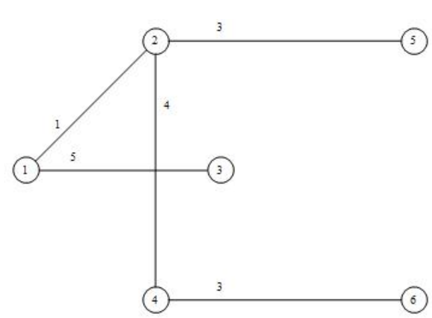
Iteración 4:

$C = \{1, 2, 4, 5, 6\}$ y $NC = \{3\}$.
$A = \{(1, 2), (2, 5), (2, 4), (4, 6)\}$.

Iteración 5:

$C = \{1, 2, 3, 4, 5, 6\}$ y $NC = \{3\}$.
$A = \{(1, 2), (2, 5), (2, 4), (4, 6), (1, 3\}$ ó
$A = \{(1, 2), (2, 5), (2, 4), (4, 6), (4, 3\}$.

El gasto final será de 16 km. de cable y la primera solución tendrá como grafo:



En el algoritmo de Kruskal para un árbol maximal el proceso es el mismo que en el caso anterior, cambiando el criterio de elección de los arcos; en lugar de tomar el arco de valor mínimo tomaríamos el de valor máximo.

3.6.2. Problemas propuestos

Problema 3.6 *(a) Repetir el ejemplo 3.1 mediante el uso del Nodo* 4 *como conjunto conectado inicial.*

(b) Ver si la solución es la misma que la anterior y si podemos deducir que la elección del conjunto conectado inicial es arbitraria.