Problema del viajante de comercio aproximado

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Problema del viajante de comercio (TSP)

Sea

Un conjunto de n ciudades “C”

Un conjunto de rutas con costo de tránsito

Existe una ruta que une cada par de ciudades

El costo de cada ruta puede ser simétrico o asimétrico (diferente valor ida y vuelta)

Queremos

Obtener el circuito de menor costo

que inicia y finalice en una ciudad inicial

y pase por el resto de las ciudades 1 y solo 1 vez

Soluciones

Por fuerza bruta

Se resuelve en $O(n!)$

Mediante programación dinámica

Hemos logrado un algoritmo exponencial $O(n^2 2^n)$

Se ha demostrado

Que corresponde a un problema NP-Completo

Por lo que (A menos que P=NP)

No podemos encontrar un algoritmo de resolución totalmente polinomial

Podremos aproximarlo de alguna manera?

Simplificaremos nuestro problema para lograrlo

Problema “simplificado” del viajante de comercio

Sea

Un conjunto de n ciudades “C”

Un conjunto de rutas con costo no negativo de tránsito

Existe una ruta que une cada par de ciudades

El costo de cada ruta es simétrico y cumple con la desigualdad triangular

Queremos

Obtener el circuito de menor costo

que inicia y finalice en una ciudad inicial

y pase por el resto de las ciudades 1 y solo 1 vez

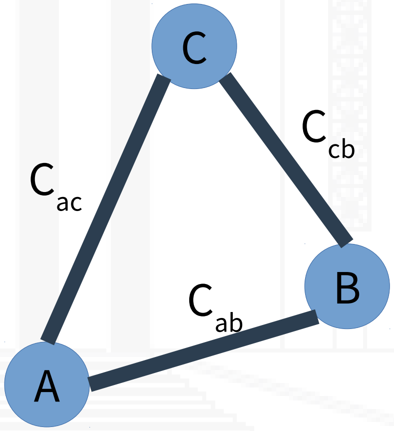
Desigualdad triangular

Viajar de la ciudad A a la ciudad B

De forma directa tiene un costo siempre menor o igual

Al de hacer el mismo viaje utilizando ciudades intermedias

$C_{ab} \leq C_{ac} + C_{cb}$



Nomenclatura

Podemos

Representar el problemas mediante un Grafo G=(V,E) con V: conjunto de ciudades y E: conjunto de rutas

Cada ruta e ∊ E, une dos ciudades u,v ∊ V tendrá un asociado el costo c(u,v)≥0

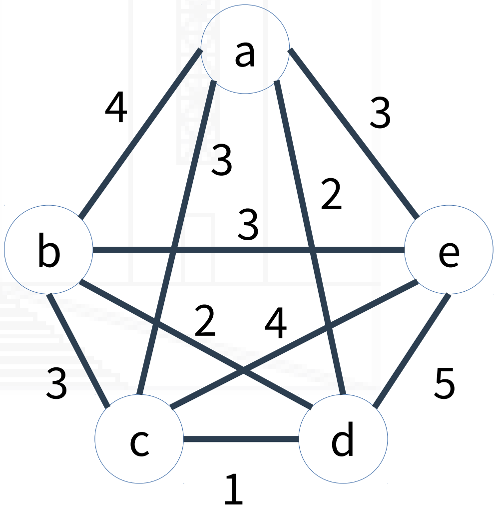
La solución

Estará computa por una secuencia de ciudades $(c_1,c_2,\ldots c_n,c_1)$

Constará de un subset A ∊ E de rutas a utilizar

El costo total incurrido será

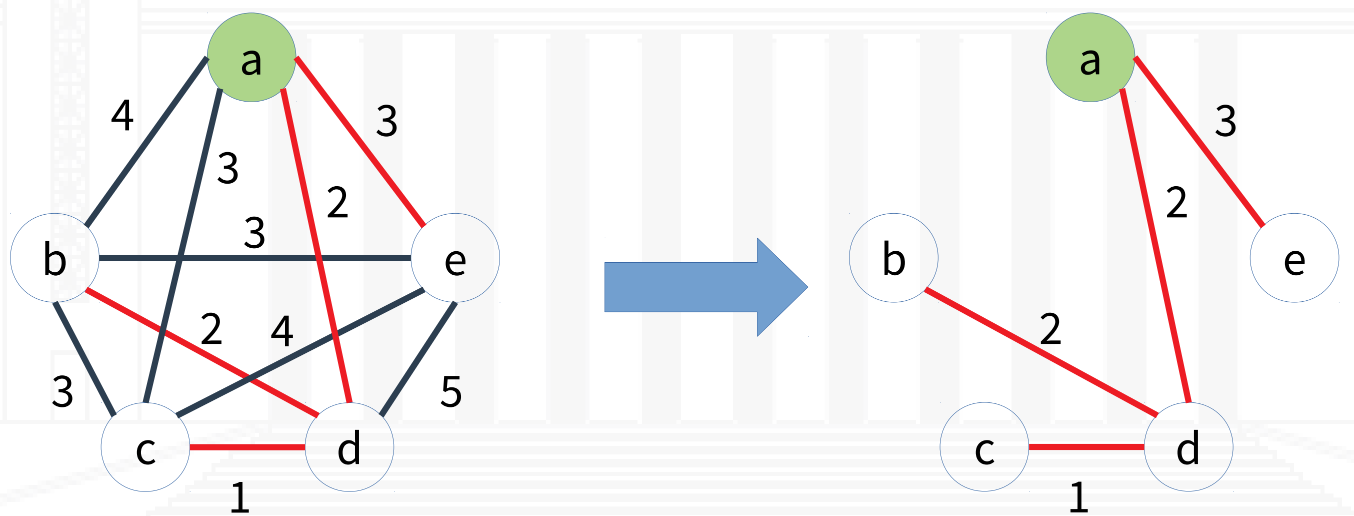
$$c(A) = \sum_{(u,v)\in A} c(u,v)$$



Iniciando con un árbol recubridor

Seleccionamos un vértice r ∈ V

- Calculamos T el árbol recubridor mínimo de G usando r como raíz
- Llamamos c(T) a la sumatoria de los ejes del arbol T



c(T)=8

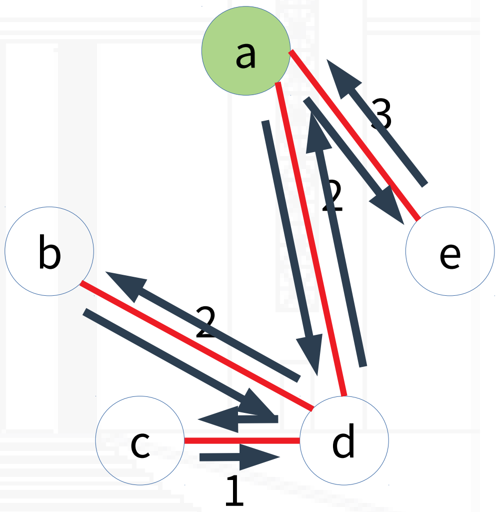
Full Walk

Podemos construir W el full walk de T

- Iniciando en la raiz y recorriendo todo el árbol hasta volver al inicio
- Con este recorrido visitamos todas las ciudades al menos una vez
- (pero solo tendría que ser una vez)

Llamamos c(W) al costo de este recorrido

- c(w)=2 c(T)



W: a,d,b,d,c,d,a,e,a

$c(W)= 2\ C_{ad} + 2\ C_{db} + 2\ C_{cd} + \dots$

Aplicando desigualdad triangular

Debemos asegurarnos que todas las ciudades (excepto la inicial)

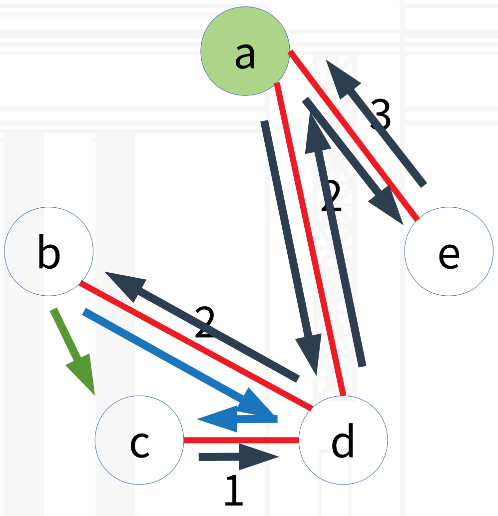
Solo sean visitadas 1 vez

Conceptualmente

Implica no volver para atrás a una ciudad visitada e ir a la siguiente sin visitar.

Lo podemos hacer gradualmente ciudad por ciudad

La desigualdad triangular nos asegura que el costo final obtenido sera menor o igual al de c(W)



$C_{bc} \leq C_{bd} + C_{dc}$

Circtuito hamiltoneano resultante

Al finalizar nos quedara un circuito hamiltoneano H

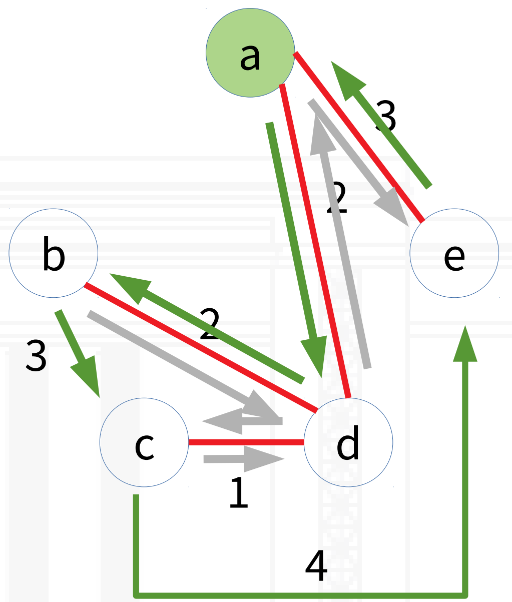
De menor costo al fullwalk y que cumple con los requerimientos del problema del viajante

No es necesario construir el full walk

Pero el concepto del full walk es útil a la hora de la demostración de aproximación

Podemos utilizar sobre el arbol T

Depth first search enumerando los nodos mediante preorden y obtenemos H



W: a,d,b,d,c,d,a,e,a

H: a,d,b,c,e,a

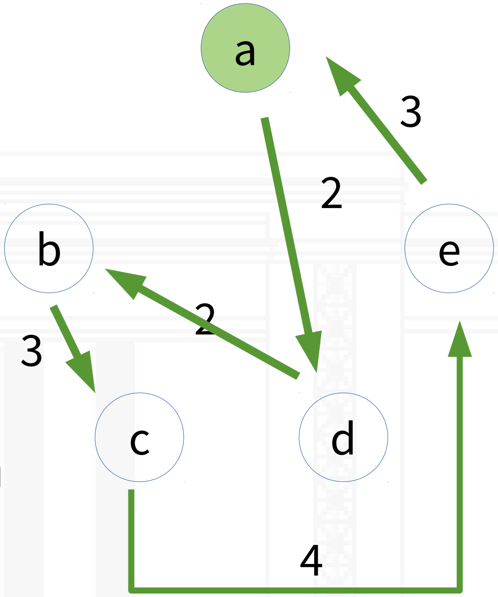
$c(W) = 2\,c(T) \geq c(H)$

$16 \quad = 2*8 \quad \geq \; 14$

Aproximación del viajante

El circuito del viajante aproximado

Corresponde al ciclo hamiltonano H encontrado



¿Qué tan buena aproximación es?

Podría pasar que la solución encontrada sea la optima

Pero de no serlo, hay alguna desviación máxima que nos asegura el algoritmo?

Análisis de la solución

Sea

H* el circuito optimo para el grafo G

T el árbol recubridor mínimo del grafo G

Eliminando solo un eje de H*

Obtenemos un árbol (que podría ser T)

Por lo tanto

El costo H* es mayor o igual al de T

$$C(T) \leq C(H^*)$$

Análisis de la solución (cont.)

Como

El costo del full walk W es el doble del costo del árbol T

Y

El costo del ciclo hamiltoneano es menor al costo del fullWalk

(por desigualdad triangular)

Podemos concluir que

El algoritmo presentado es un 2-algoritmo de aproximación

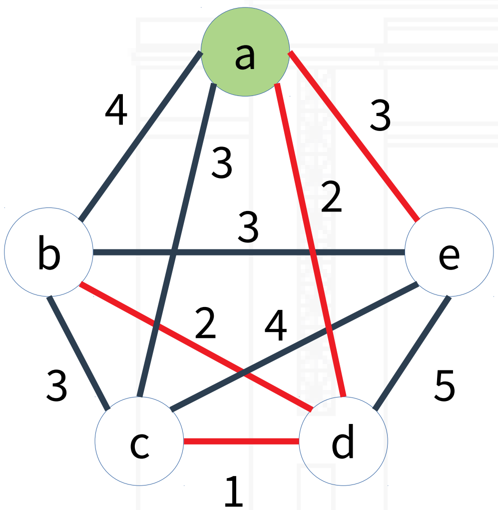
$$C(T) \leq C(H^*)$$
$$C(W)=2C(T)$$
$$C(H) \leq C(W)$$

$$\Downarrow$$

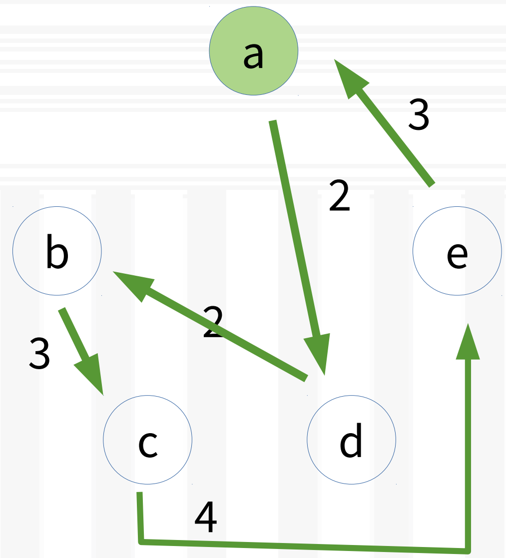
$$C(H) \leq 2C(H^*)$$

En el ejemplo

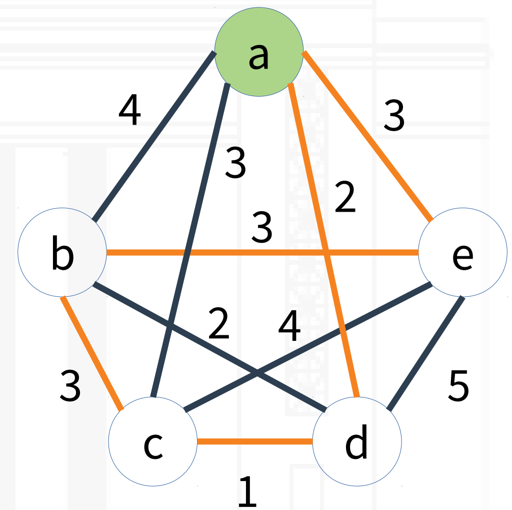
$C(H) \leq 2C(H^*)$



C(T)=8



C(H)=14



C(H*)=12

Complejidad Temporal

El algoritmo se puede dividir en las siguientes partes:

- Cálculo del árbol recubridor mínimo de G
- Recorrido de T mediante DFS para generar lista H (utilizando preorden)

Ambos algoritmos se pueden ejecutar en tiempo polinomial

- Árbol recubridor usando Kruskal → O(E log V)
- DFS en un árbol → O(V)

Por lo que nuestro algoritmo se ejecuta en tiempo polinomial

.UBA fiuba
FACULTAD DE INGENIERÍA

Presentación realizada en Enero de 2021