.UBAfiuba
FACULTAD DE INGENIERÍA

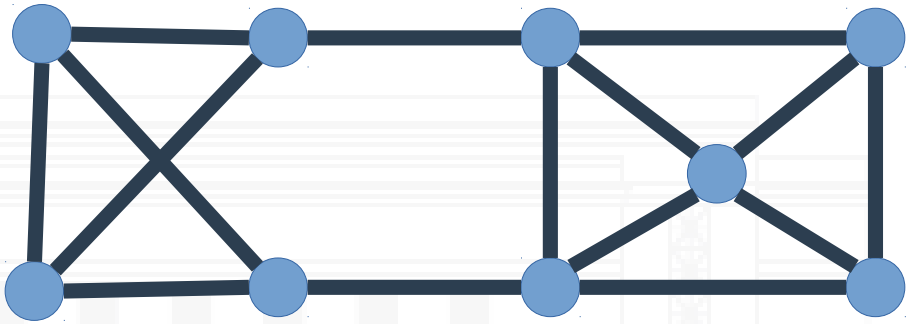
k-conectividad de ejes de un grafo

Teoría de Algoritmos I (75.29 / 95.06)
Ing. Víctor Daniel Podberezski
vpodberezski@fi.uba.ar

Enunciado

Sea

G=(V,E) grafo conexo y no dirigido



Deseamos saber

¿Cuantos ejes se pueden remover antes que G deje de ser conexo?

Este problema se conoce como

K-conectividad de ejes en un grafo

Análisis del problema

Podemos pensar el problema

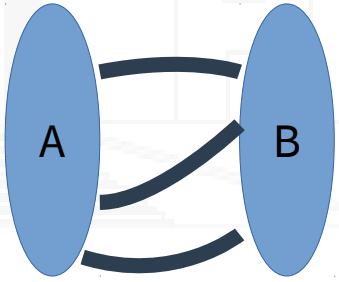
Como encontrar el corte global mínimo del grafo

Analizamos

toda posible subdivisión A-B del grafo en 2 conjuntos disjuntos

Contamos para cada corte

la cantidad de ejes entre conjuntos



Reducción a problema de flujos

Por cada eje e=(u,v)

Creamos 2 ejes dirigidos (u,v) y (v,u)

Les asignamos una capacidad de 1

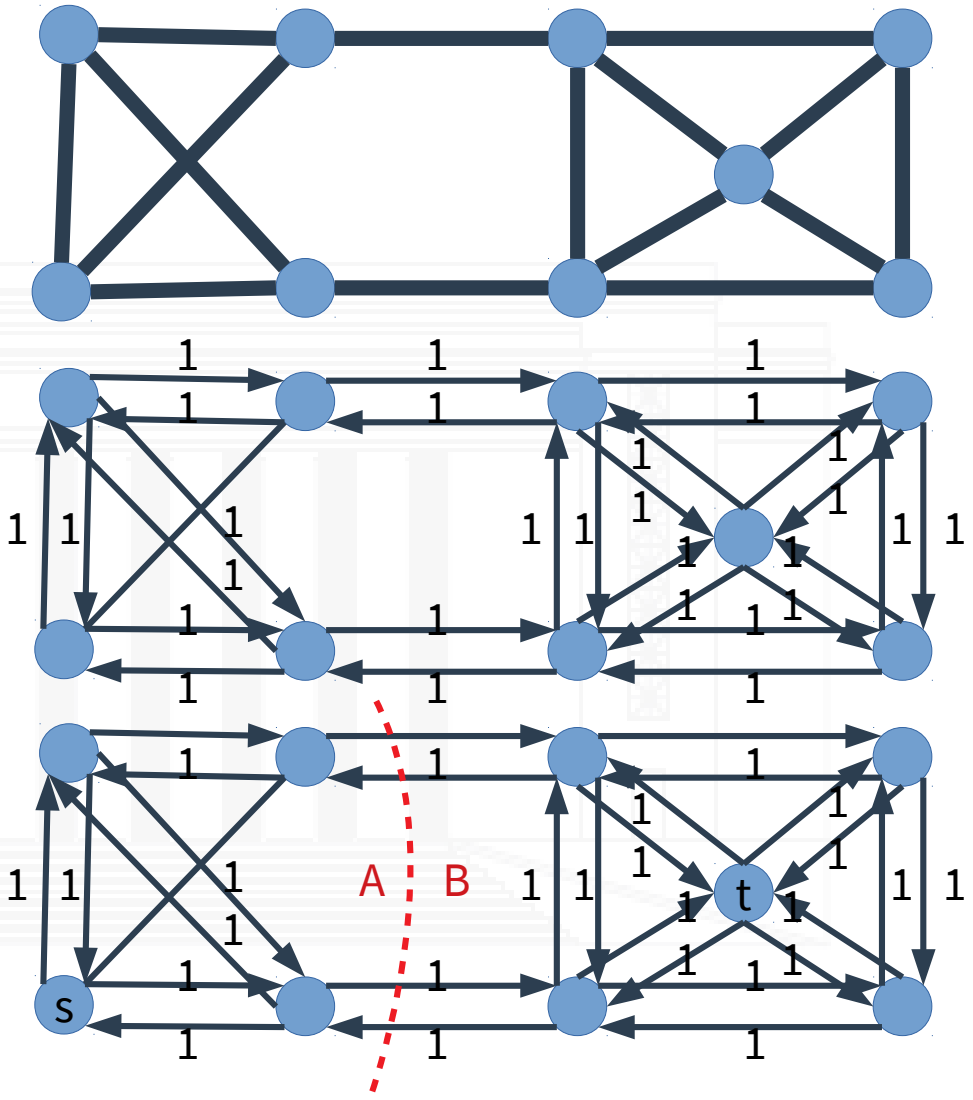
Por cada combinación posible de 2 nodos

Etiquetarlos como s y t respectivamente

Resolver “MAX-FLOW MIN-CUT”

El menor de los cortes mínimos

Corresponde al valor de la K-conectividad de ejes del grafo



Complejidad de la solución

Debemos repetir

El problema de flujo máximo $\binom{2}{|V|} = \frac{|V|!}{2! \cdot (|V|-2)!} = O(|V|^2)$

Cada problema de flujo

Si usamos Ford-Fulkerson → O(C|E|),

Siendo C, la sumatoria de las capacidades que salen de la fuente.

Como todas las capacidades son 1, y como mucho puede estar conectado con |V|-1 vértices.

Si expresamos la cantidad de ejes en función de los vertices, en el peor de los casos tendremos → |E| = |V|*(|V|-1)/2

La complejidad será O(|V|^3)

Finalmente nos queda como complejidad

O(|V|^5)

Una mejora de una magnitud

No hace falta realizar

Las $|V|^2$ combinaciones de nodos para fuente y sumideros

Si analizamos el problema, veremos que muchos de los cortes A-B se repiten entre diferente elección del nodo fuente y sumidero

Alcanza

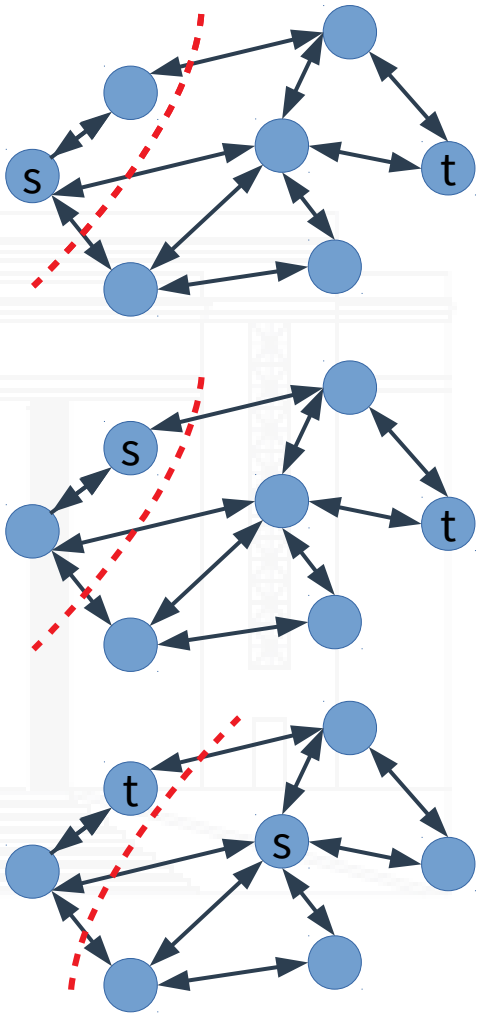
Seleccionando 1 nodo como fuente

Y utilizando en cada subproceso de MAX-FLOW MIN-CUT un sumidero diferente entre los $|V|-1$ nodos restantes

Son $O(|V|)$ en total

La complejidad

Baja a $O(|V|^4)$



Una propuesta superadora

Se puede resolver más rápido?

Propondremos utilizar un algoritmo randomizado

Presentaremos

Karger's algorithm

Propuesto en 1993 por David Karger

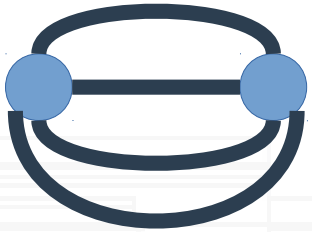
En "Global Min-cuts in RNC and Other Ramifications of a Simple Mincut Algorithm"

Algoritmo de Karger

Es un algoritmo

Randomizado que funciona en tiempo polinomial

Y puede retornar un resultado erroneo



Multigrafo de 2 nodos

Funciona

Para multigrafos (pueden existir más de un eje que une a dos nodos)

Utiliza el proceso

De contracción del grafo (por lo que se lo conoce también como “contraction Algorithm”)

Va reduciendo el tamaño del grafo iterativamente

Proceso de contracción

Seleccionar

Un eje e=(u,v) de forma aleatoria y uniforme

Reemplazar

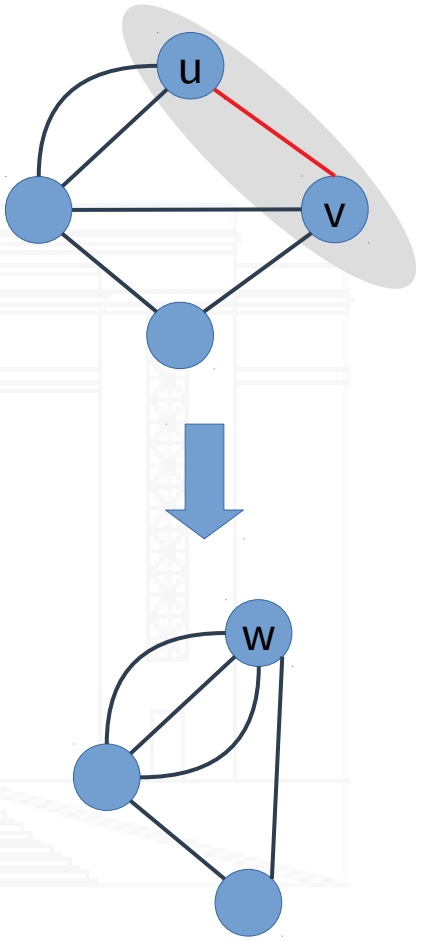
los nodos u y v por un nuevo nodo w

Todos los ejes (u,v) se eliminan

Los ejes (u,a) y (v,a) con a ∈ E – {u,v} se reemplaza por (w,a)

Repetir

Hasta que solo queden 2 nodos en el grafo G resultante



Funcionamiento

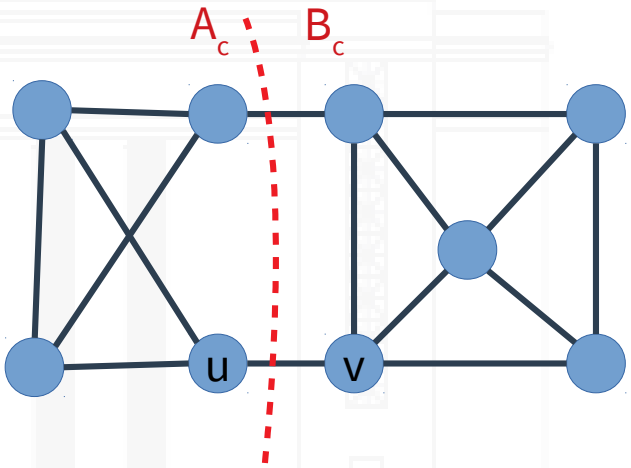
Si consideramos el corte mínimo “C”

Divide el grafo en 2 conjuntos que llamaremos A_c y B_c

El algoritmo no encontrará el valor k

Si en alguna iteración contrae nodos u, v que se encuentran en A_c y B_c respectivamente

Para eso debe seleccionar un eje que pase el corte mínimo global



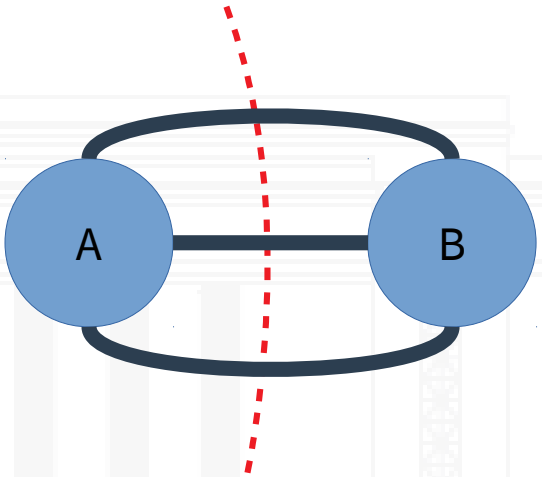
Solo hay 2 entre 15 de posibilidad de seleccionar un eje en el corte mínimo

Fin de la ejecución

Al finalizar la ejecución

Quedarán 2 “super nodos”

Con ejes entre ellos



Representan

Un posible corte A-B

La cantidad de ejes entre A y B

Son “probablemente” la K-conectividad de ejes del grafo

Pseudocódigo y Complejidad

La iteración principal

Se ejecuta |V|-2 veces

La construcción del nuevo grafo

Tiene como cota O(|V|) si se utiliza una matriz de adyacencia para representar el grafo

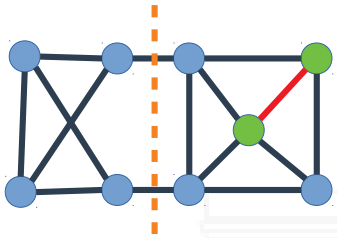
Una ejecución del algoritmo

Tiene una complejidad temporal $O(|V|^2)$

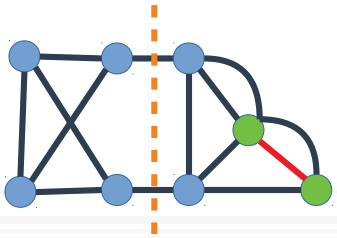
Y una complejidad espacial $O(|V|^2)$

```
Mientras |G.V|>2
    Seleccionar aleatoriamente eje
                        e=(u,v) de G.E

    Contraer G mediante e

Retornar |G.E|
```

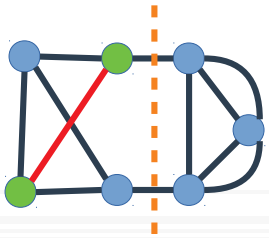
Ejemplo



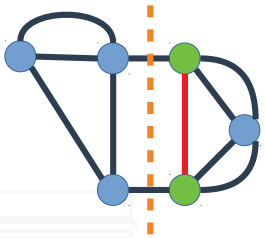
Probabilidad de elegir eje en corte: 2/15



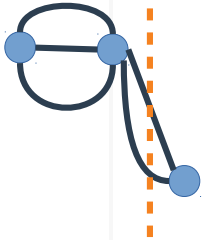
Probabilidad de elegir eje en corte: 2/14



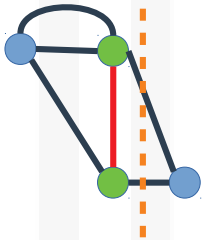
Probabilidad de elegir eje en corte: 2/12



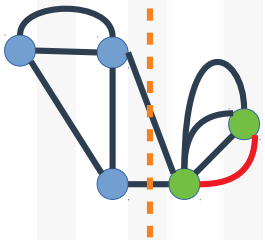
Probabilidad de elegir eje en corte: 2/11



Probabilidad de elegir eje en corte: 2/5

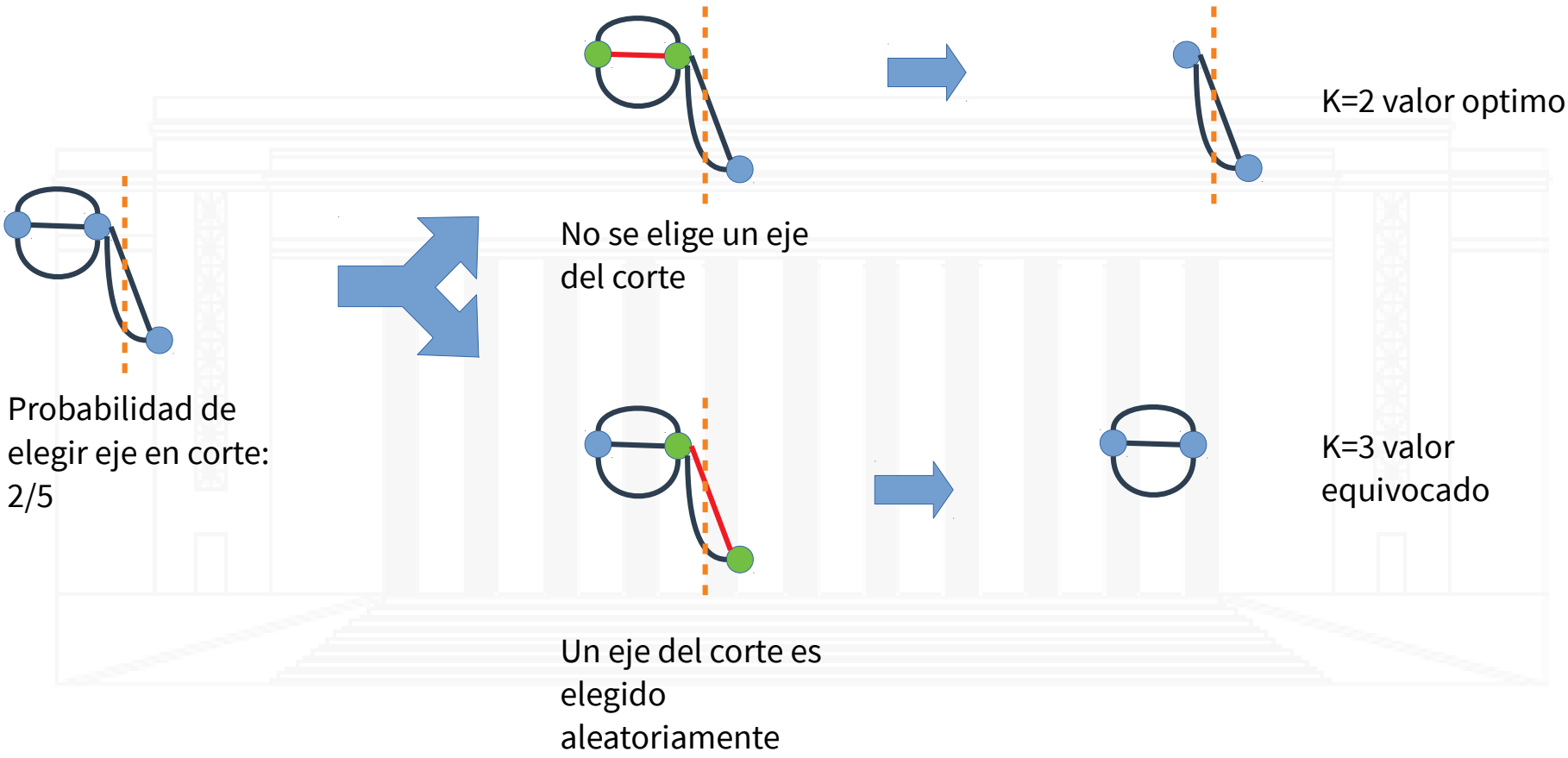


Probabilidad de elegir eje en corte: 2/6



Probabilidad de elegir eje en corte: 2/10

Ejemplo (cont.)



Probabilidad de éxito

Llamaremos

ε al evento de encontrar el corte mínimo global

Para lograr ε

Se tiene que cumplir que en cada iteración no se selecciona un eje de “O”

Llamaremos ε_k

Al evento de en la iteracion k no contraer un eje de ”O”

Probabilidad de éxito (cont.)

Por lo tanto

$$\varepsilon = \varepsilon_1 \cap \varepsilon_2 \cap \ldots \cap \varepsilon_{n-2}$$

La probabilidad que ocurra el evento ε

$$Pr(\varepsilon) = Pr(\varepsilon_1 \cap \varepsilon_2 \cap \ldots \cap \varepsilon_{n-2})$$

$$Pr(\varepsilon) = Pr(\varepsilon_1) * Pr(\varepsilon_2 / \varepsilon_1) * \ldots * Pr(\varepsilon_{n-2} / \varepsilon_{n-3}, \varepsilon_{n-4}, \ldots, \varepsilon_1)$$

Probabilidad de éxito (cont.)

En la primera iteración

$$Pr(\varepsilon_1) = 1 - Pr(\bar{\varepsilon}_1)$$

Donde

$$Pr(\bar{\varepsilon}_1) = k / |E|$$

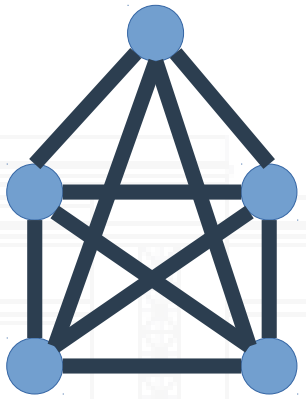
Podemos acotar

$|E| \geq k|V| / 2$ (pues todos los nodos tienen al menos k ejes)

Por lo tanto

$$Pr(\bar{\varepsilon}_1) \leq k / (k|V| / 2) = 2 / |V|$$

$$Pr(\varepsilon_1) \geq 1 - (2 / |V|) = (|V| - 2) / |V|$$



En un grafo completo:
k=|V|-1
|E| = |V|*(|V|-1)/2

Probabilidad de éxito (cont.)

Hasta el momento

acotamos $Pr(\varepsilon_1)$

De

$$Pr(\varepsilon) = Pr(\varepsilon_1) * Pr(\varepsilon_2 / \varepsilon_1) * \ldots * Pr(\varepsilon_{n-2} / \varepsilon_{n-3}, \varepsilon_{n-4}, \ldots, \varepsilon_1)$$

¿Cómo calculamos

$$Pr(\varepsilon_i / \varepsilon_{i-1}, \varepsilon_{i-2}, \ldots, \varepsilon_1) \ ?$$

Probabilidad de éxito (cont.)

Al momento del evento ε_i

Quedan $|V|-i+1$ nodos

Cada uno de esos nodos

Deben tener al menos k ejes incidentes

(es absurdo que algo menor, o el algoritmo podría retornar un valor menor a k! … y el algoritmo de contracción no funciona de esa manera)

Por lo tanto al momento de ε_i

$|E| \geq k * (|V|-i+1) / 2$

Probabilidad de éxito (cont.)

Con

$$|E| \geq k * (|V|-i+1) / 2$$

Podemos acotar

$$Pr(\bar{\varepsilon}_i / \varepsilon_{i-1},\varepsilon_{i-2}, \ldots , \varepsilon_1) \leq k / (k * (|V|-i+1) / 2) = 2 / (|V| - i +1)$$

$$Pr(\varepsilon_i / \varepsilon_{i-1},\varepsilon_{i-2}, \ldots , \varepsilon_1) \geq 1 - 2 / (|V| - i +1) = (|V| - i -1) / (|V| - i + 1)$$

… y con esto podemos calcular la probabilidad total

Probabilidad de éxito (cont.)

La probabilidad de encontrar con éxito k

$$\Pr(\varepsilon) = \Pr(\varepsilon_1) * \Pr(\varepsilon_2 / \varepsilon_1) * \ldots * \Pr(\varepsilon_{n-2} / \varepsilon_{n-3}, \varepsilon_{n-4}, \ldots, \varepsilon_1)$$

$$Pr(\varepsilon) \geq \frac{|V| - (|V|-2) - 1}{|V| - (|V|-2) + 1} * \frac{|V| - (|V|-3) - 1}{|V| - (|V|-3) + 1} * \ldots * \frac{|V|-2}{|V|} = \frac{1}{3} * \frac{2}{4} * \ldots * \frac{|V|-2}{|V|}$$

$$Pr(\varepsilon) \geq \frac{2! * (|V|-2)!}{|V|!} = \binom{|V|}{2}^{-1} = 1 / \binom{|V|}{2}$$

$$Pr(\varepsilon) \geq \frac{2}{|V| * (|V|-1)}$$

Si el grafo es muy grande

… parece ser una probabilidad muy pequeña

Mejorar las chances de exito

Podemos

Ejecutar varias veces el programa y quedarnos con el mejor resultado encontrado.

Cada ejecución

Aumenta la probabilidad de hallar k

Podemos medir

La probabilidad que en n iteraciones no hallemos k como $\left(1 - 1/\binom{|V|}{2}\right)^n$

Mejorar las chances de éxito (cont.)

¿Cuántas iteraciones

Debemos realizar hasta tener una alta probabilidad?

Partiendo de

$$\left(1-1/\binom{|V|}{2}\right)^{n} \Rightarrow \left(1-1/\binom{|V|}{2}\right)^{\binom{|V|}{2}} \leq \frac{1}{e}$$

Por lo tanto

si ejecuto n = $\binom{|V|}{2} * \ln|V|$, entonces

$$\left(1-1/\binom{|V|}{2}\right)^{\binom{|V|}{2} * \ln|V|} \leq \left(\frac{1}{e}\right)^{\ln|V|} = 1/|V|$$

Hint:

$$\left(1-\frac{1}{x}\right)^{x} \leq \frac{1}{e} \quad , x \geq 1$$

Complejidad total

Ejecutar

el algoritmo Karger $O(|V|^2 * \log |V|)$ veces

Nos da una complejidad temporal

$O(|V|^4 * \log |V|)$ ← Todo esto y funciona peor comparado al método anterior ????

Y asegura con alta probabilidad

Encontrar el valor k

¿¿¿Y encima puede fallar???

ALTO! Esto no termina aca....

Mejoras: steiner-karger

En 1996

David Karger y Clifford Stein

Publicaron

"A new approach to the Minimum cut problem"

http://www.columbia.edu/~cs2035/courses/ieor6614.S09/Contraction.pdf

Es una variante del anterior

Si |V| es pequeño resuelve por fuerza bruta

Sino contrae |V| / sqrt(2) nodos y aplica técnica de división y conquista con el resto

Encuentra con alta probabilidad k

con en $O(|V|^2 \log^3 |V|)$ tiempo

.UBAfiuba

FACULTAD DE INGENIERÍA

Presentación realizada en Junio de 2020