NP-COMPLETOS

Dense Subgraph Problem

Autores
Cledy Elizabeth Becerra Sipiran
Massiel Oviedo Sivincha
Harold Alejandro Villanueva Borda

Tutor
Rensso Victor Hugo Mora Colque

Universidad Católica San Pablo
Análisis y Diseño de Algoritmos
Arequipa, Peru
Junio de 2022

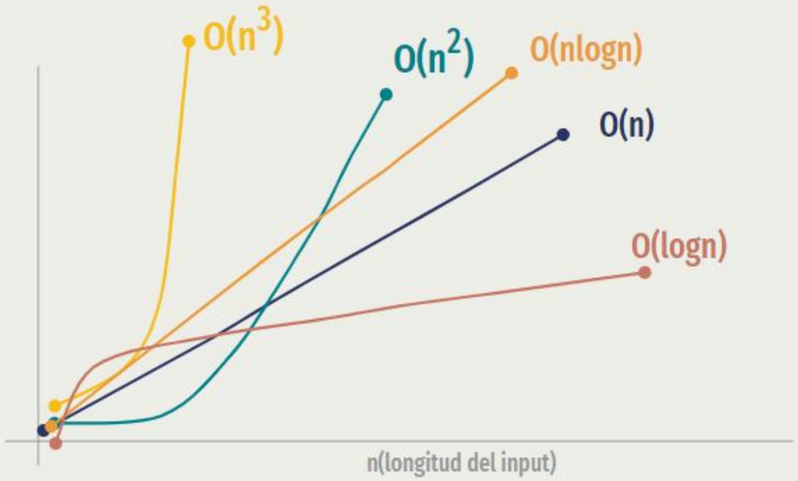
01 Conceptos Básicos

Tiempo Polinomial

Tiempo menor o igual que un polinomio

Algoritmo Strassen

$$O(n^{\log_2 7}) \approx O(n^{2.807})$$



MergeSort

Multiplicacion Matricial

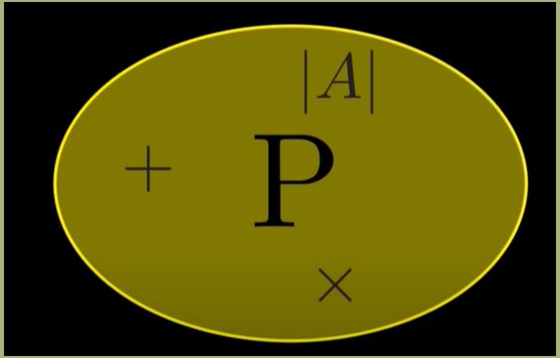
Busqueda Lineal

Insertion Sort

Busqueda Binaria

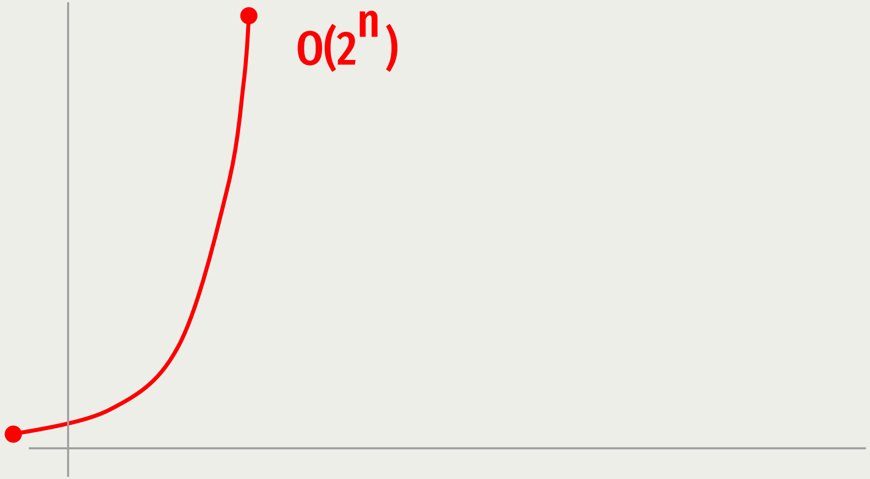
Clase P

Problemas decidibles que se pueden ***resolver*** con una **Máquina Turing determinista** usando una cantidad de tiempo de computación polinomial, en función a la longitud de la entrada.



Tiempo Exponencial

Tiempo mayor a un polinomio dado siempre y cuando n sea lo suficientemente grande



0/1 KnapSack

Suma de SubConjuntos

Circuito Hamiltoniano

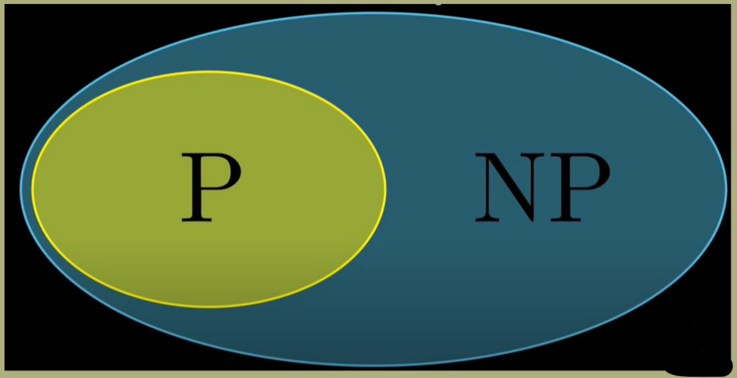
Agente Viajero

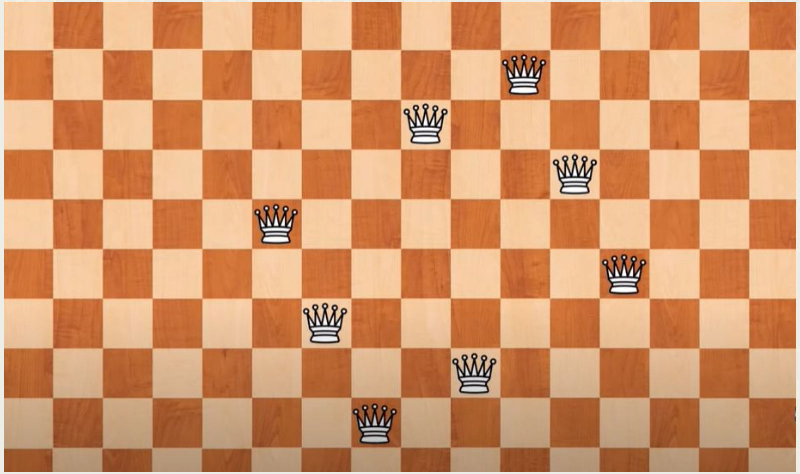
Coloreo de Grafos

Clase NP

Problemas decidibles que se pueden *verificar* en una **Máquina Turing no determinista** usando una cantidad de tiempo de computación polinomial, en función a la longitud de la entrada.

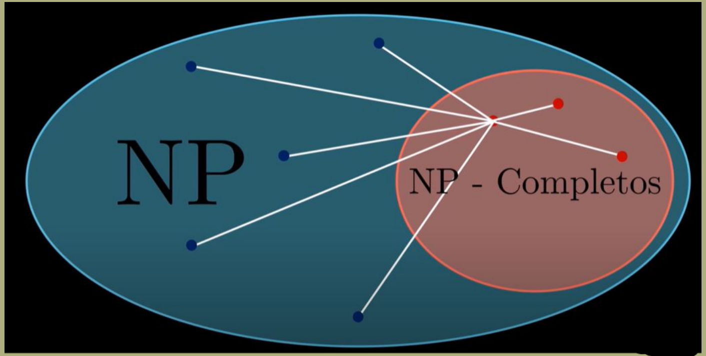
$$P \subseteq NP$$



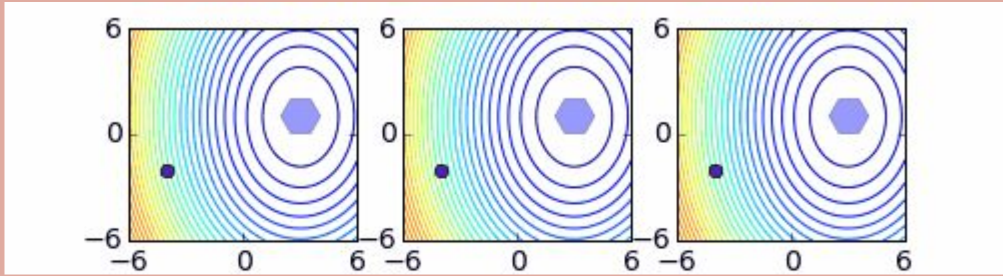


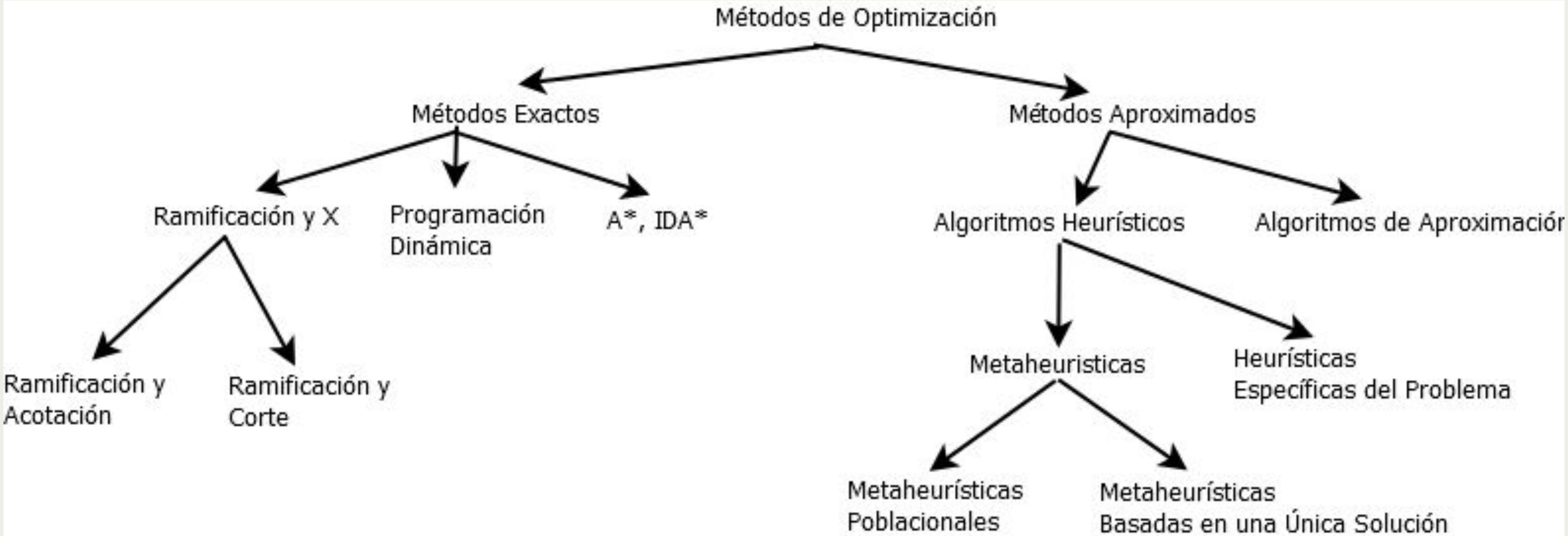
Clase NP-Completos

Problemas decidibles que se pueden *reducir* en una **Máquina Turing determinista** usando una cantidad de tiempo de computación polinomial, en función a la longitud de la entrada.Si uno puede resolver un problema de NPC de manera eficiente, entonces todos los problemas de NP pueden resolverse de manera eficiente.

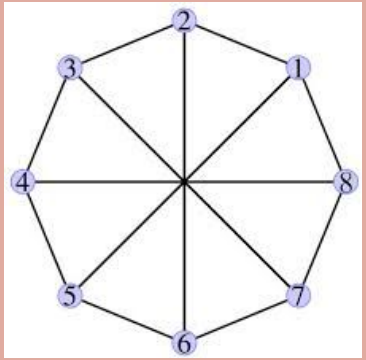


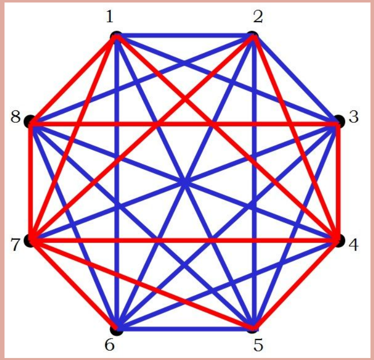
02 Métodos de Optimización




Métodos de Optimización
Métodos Exactos
Métodos Aproximados
Ramificación y X
Programación Dinámica
A*, IDA*
Algoritmos Heurísticos
Algoritmos de Aproximación
Ramificación y Acotación
Ramificación y Corte
Metaheuristicas
Heurísticas Específicas del Problema
Metaheurísticas Poblacionales
Metaheurísticas Basadas en una Única Solución

03 Sub-grafos Densos





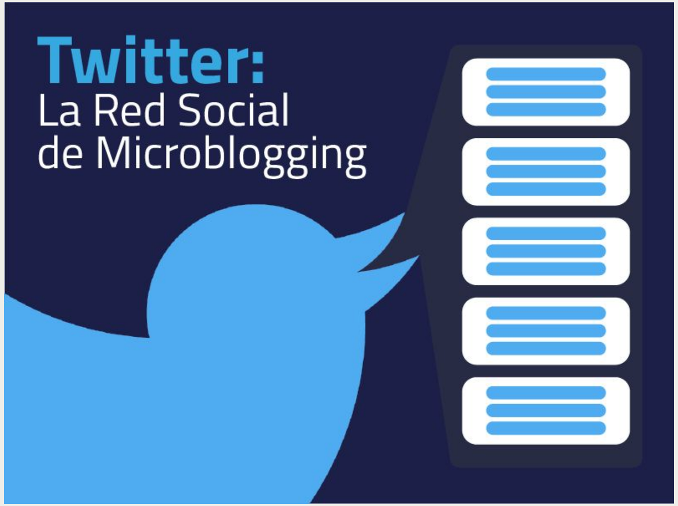
Sea G = (V, E) con n = |V | v´ertices y m = |E| aristas.

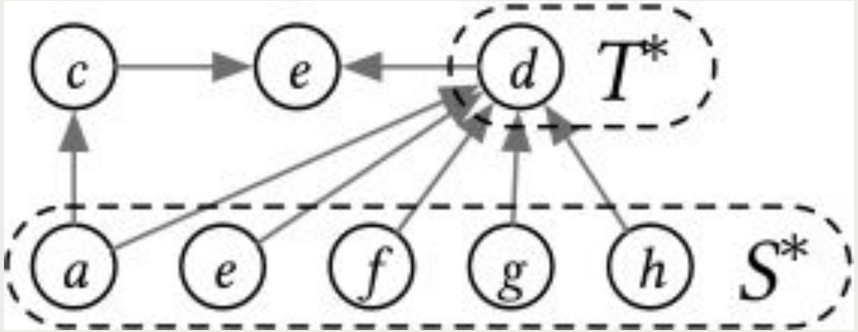
Para un grafo simple (sin bucles) y que el grafo es no dirigido, su densidad Δ :

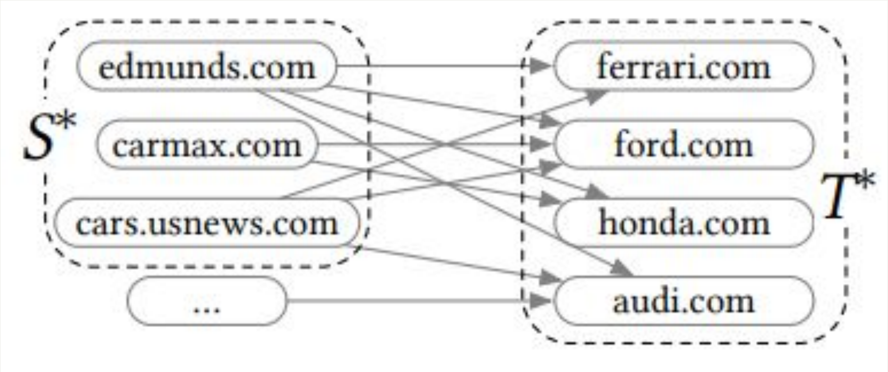
$$\Delta = \frac{m}{n(n-1)/2} = \frac{2m}{n(n-1)}$$

Si el grafo es dirigido, su densidad Δ:

$$\Delta = \frac{m}{n(n-1)}$$


Twitter:
La Red Social
de Microblogging


c
e
d
T^*
a
e
f
g
h
S^*

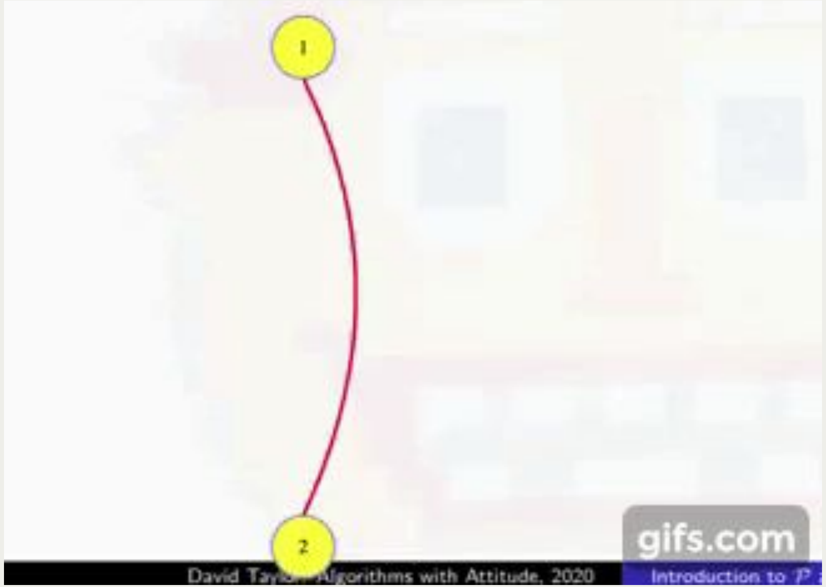

S^*
edmunds.com
carmax.com
cars.usnews.com
...
ferrari.com
ford.com
honda.com
audi.com
T^*

04 Demostración Generalizada del Problema NP-Completo

Problema del clique

Problema NP-completo según la Teoría de la complejidad computacional.

El problema de clique es un problema de decisión para determinar cuándo un grafo contiene un clique de al menos un tamaño k. Una vez que tenemos k o más vértices que forman un clique, es trivial verificar que lo son, por eso es un problema NP.



Dado el grafo $G = (V, E)$ y dos enteros a y b. Sea a un conjunto de varios vértices de G tales que hay al menos b aristas entre ellos se conoce como subgrafo denso del grafo G.

Prueba

Conversion de entrada:

Convertir la entrada de Clique a la entrada del Subgrafo denso.

<u>Clique Input:</u> un gráfico no dirigido G(V, E) y un entero k.

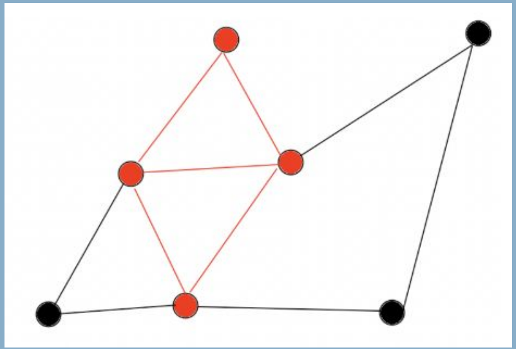
<u>Dense-Subgraph Input:</u> un gráfico no dirigido $G'(V, E)$ y dos enteros a y b.

Conversión de salida:

Solución del problema Clique. La solución del gráfico denso dará como resultado un conjunto a que sería un Clique de tamaño k como k = a. Por lo tanto, Clique puede utilizar la salida directa de Dense Graph. Dado que no se requiere conversión, nuevamente es de naturaleza polinomial.

Prueba

Correccion



Conclusion:

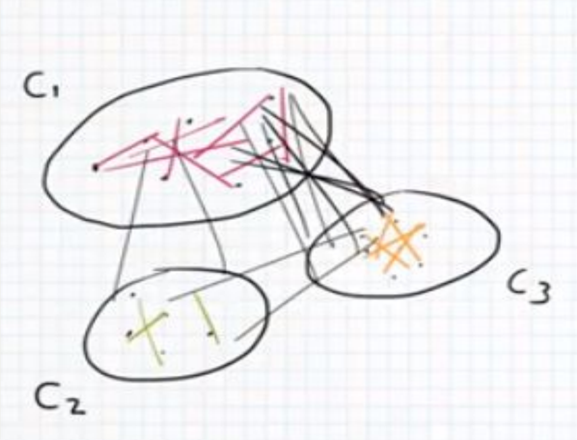
Por lo tanto, podemos concluir que Dense-Subgraph es NP-Completo.

05 Demostración Formal del Problema NP-Completo

Método del Bloque Estocástico

Podemos pensar que los vértices de una red pertenecen a clases y que la propensión a establecer vínculos entre pares de vértices depende de la pertenencia de clase de los dos vértices.Los enlaces se producen debido a la equivalencia estructural (structural equivalence), es decir, la similitud de los roles sociales.El vértice i ∈ V del grafo G = (V,E) pertenece a una sola clase de una partición P = (Ci...., Co) de V con Q comunidades. El modelo de bloques se puede escribir como:

$$p(\mathbf{y} \mid \boldsymbol{\theta}) = \frac{1}{\kappa} \exp\left\{ \sum_{q,r} \theta_{q,r} L_{q,r}(\mathbf{y}) \right\}$$



SBM

$$\underset{3\times 3}{\Pi} = \begin{array}{c c} & \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{array}{c} 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 0.8 & 0.12 & 0.93 \\ 0.12 & 0.2 & 0.03 \\ 0.93 & 0.03 & 0.87 \end{bmatrix} \end{array}$$

$$\mathbb{P}(y_{ij} = 1 \mid \xi_i, \xi_j, \pi) = \pi_{\xi_i, \xi_j}$$

$\xi_i = 2 \qquad \xi_j = 3$

$\pi_{2,3}$

Tenemos un grafo aleatorio $G = (V,E)$, con $[V]= n$, con la siguiente estructura:

- Un subconjunto $Q \subset V$, con $|Q|=k$, de los nodos forma una comunidad fuertemente conexa, las aristas se dan con probabilidad $1-\delta(\delta \in [0,½])$.
- El resto de las aristas se darán con probabilidad 1/2.

Stochastic Block Models (k,n-k), con matriz de conectividad:

$$\begin{pmatrix} 1-\delta & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$$

El interés está en identificar el subconjunto de los nodos que forman la componente fuertemente conexa.

PROBLEMA

(k,f(k))-DSP es NP-completo donde k es el número de vértices y f(k) el número de aristas. Consideramos el problema del Clique, que pregunta si existe un gráfico completo de n-vértices (n-clique) en un gráfico de 2n vértices dado G = (V; E). Se puede demostrar fácilmente que este problema de clique sigue siendo NP-completo por reducción del problema general de k-clique de la siguiente manera

PRUEBA

Para un gráfico de entrada I de n-vértices, agregamos un grafo completo de (n-k)-vértices K, conexión bipartita completa entre I y K, y k vértices aislados.

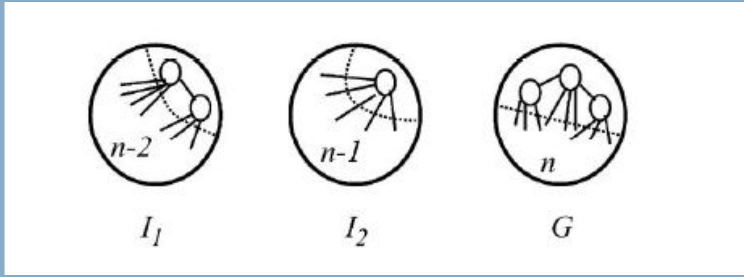
Sea $f(k)=nm(2n - 1) + n(n - 1)/2$, donde m es un polinomio en n y determinado luego. Construya un gráfico $H = (V', E')$ compuesto por una copia $G =(V; E)$ y m grafos completos, cada uno de los cuales tiene 2n vértices. H tiene $|V'|=2n(m + 1)$ vértices y $|E'|=|E| + nm(2n - 1)$ aristas en total. Luego establece $k = 2nm + n$. Esta construcción de H se puede hacer en tiempo polinomial obviamente.

Lema

Supongamos que hay m grafos completos I1,...,Im de 2n vértices y un grafo G de 2n vértices. Tome 2nm+n vértices entre esos 2nm + 2n. Entonces el número de aristas inducidas se vuelve máximo cuando tomamos todos los vértices de 2nm de I1,...,Im (y otros n de G).

Prueba

Supongamos que tomamos 2nm-d vértices de I1,...,Im y n+d vértices de G.



Este lema muestra que un subgrafo de k-vértices más denso de H consta de todas los Cliques de 2n-vértices de Im y un subgrafo de n-vértices de G'.

06 Algoritmo Aproximado

Definición y notación

Definición 2. Sean $S, T \subseteq V$. Definimos la densidad $d(S,T)$ del par de conjuntos S, T como:

$$d(S,T) = \frac{|E(S,T)|}{\sqrt{|S||T|}}$$

Definimos la densidad $d(G)$ de una grafo dirigido $G(V,E)$ como:

$$d(G) = \max_{S,T \subseteq V} \{d(S,T)\}$$

Kanna y Vinay
Linear programming
Greedy 2-approximation algorithm

El grado de un vértice i en S es el número de aristas de i a T .
El grado de un v´ertice j en T se define de manera similar

1. Inicialmente, $S \leftarrow V, T \leftarrow V$.

2. Sea i_{min} el vértice $i \in S$ que minimiza $|E(i,T)|$. Sea $d_S \leftarrow |E(i_min,T)|$.

3. Sea j_{min} el vértice $j \in T$ que minimiza $|E(S,j)|$. Sea $d_T \leftarrow |E(S,j_{min})|$.

4. Si $\sqrt{c}.d_S \leq \frac{1}{\sqrt{c}}.d_T then$

conjunto $S \leftarrow S - i_{min}$

caso contrario

conjunto $T \leftarrow T - j_{min}$.

5. Si S y T no están vacíos, regrese al paso 2.

Lema 1

$$\max_{|S|/|T|=c} \{d(S,T)\} \leq \sqrt{c} \cdot d_{out}^{\max} + \frac{1}{\sqrt{c}} \cdot d_{in}^{\max}$$

PRUEBA:

Considere el par de conjuntos S, T que maximiza d(S, T) sobre todos los pares S, T como |S|/|T| = c . Ahora, cada arista en E(S, T) debe ser asignada en un vértice en S o un vértice en T . Entonces:

$$|E(S,T)| \leq |S| \cdot d_{out}^{\max} + |T| \cdot d_{in}^{\max}$$

$$d(S,T) = \frac{|E(S,T)|}{\sqrt{|S||T|}} \leq \sqrt{\frac{|S|}{|T|}} d_{out}^{\max} + \sqrt{\frac{|T|}{|S|}} d_{in}^{\max}$$

$$= \sqrt{c} \cdot d_{out}^{\max} + \frac{1}{\sqrt{c}} \cdot d_{in}^{\max}$$

Lemma 2. Sea v el valor máximo de $d(S,T)$ para todos los pares de conjuntos S,T obtenidos durante la ejecución algoritmo greedy para un valor en partiular de c. Luego $\sqrt{c}.d_{out}^{max} \leq v$ y $\frac{1}{\sqrt{c}}.d_{in}^{max} \leq v$.

Prueba: Consideramos una ejecución de los pasos 2 al 5 en cualquier punto del algoritmo.

$$\min(\sqrt{c} \cdot d_S, \frac{1}{\sqrt{c}} d_T) \leq \sqrt{d_S d_T} \leq \frac{|E(S,T)|}{\sqrt{|S||T|}} \leq v.$$

Juntando los lemas 1 y 2 se obtiene

Lema 3. Sea v el valor máximo de $d(S,T)$ para todos los pares de conjuntos S,T obtenidos durante la ejecución del algoritmo greedy para un valor particular de c. Después,

$$v \geq \frac{1}{2} \max_{|S|/|T|=c} \{d(S,T)\}.$$

Aplicando el lema anterior para el valor específico c = |S∗ |/|T∗|, obtenemos el siguiente límite en la relación de aproximaci ́on del algoritmo.

Teorema. El algoritmo greedy da una aproximación de 2 para $d(G)$.

Observación. En lugar de ejecutar el algoritmo para todos los valores de (n2) de c, podemos adivinar el valor de c en la soluci´on ´optima dentro de un factor $(1 + \varepsilon)$ usando solo $O(\log n)$ valores, esto perder´ia solo un factor $(1 + \varepsilon)$ en la relación de aproximaci´on. Necesitamos modificar el Lema 1 para incorporar el factor $(1 + \varepsilon)$