



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA MATEMÁTICA

EL PROBLEMA DE LA DEGENERANCIA DE GRAFOS EN CONGESTED CLIQUE

TESIS PARA OPTAR AL GRADO DE MAGÍSTER EN CIENCIAS DE LA
INGENIERÍA, MENCIÓN MATEMÁTICAS APLICADAS
MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL MATEMÁTICO

SEBASTIÁN WALTER PÉREZ SALAZAR

PROFESOR GUÍA:
IVÁN RAPAPORT ZIMERMANN

MIEMBROS DE LA COMISIÓN:
MARTÍN MATAMALA VÁSQUEZ
HIỆP HÀN

Este trabajo ha sido parcialmente financiado por Proyecto Fondecyt 1130061

SANTIAGO DE CHILE
2016

EL PROBLEMA DE LA DEGENERANCIA DE GRAFOS EN CONGESTED CLIQUE

La computación distribuida, rama de las ciencias de la computación, se focaliza en estudiar sistemas distribuidos, tales como internet, redes sociales o protocolos de mensajería. La información se encuentra distribuida entre las distintas entidades que conforman el sistema y el objetivo último es resolver algún problema global. Para ello las distintas entidades se comunican mediante los canales de la red hasta que encuentran la solución.

Esta memoria comienza estudiando el problema de la degenerancia de un grafo en los modelos de computación distribuida UCAST y BCAST, donde la degenerancia de un grafo G se define como el máximo grado mínimo de un subgrafo F de G . Los modelos distribuidos UCAST y BCAST corresponden a redes completas de n individuos, los cuales se comunican de manera síncrona por los canales de la red en rondas. En el primer caso, cada individuo por ronda puede enviar diferentes mensajes por cada uno de sus canales. En el segundo caso, cada individuo por ronda envía a través de sus canales el mismo mensaje. En general, se suele decir que BCAST es una restricción de UCAST.

Primero, se construye un protocolo aleatorio en el modelo UCAST que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia en $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$. En el modelo BCAST se demuestra que el problema de calcular la degenerancia es difícil en 1 ronda. Más específicamente, se demuestra que todo protocolo aleatorio de 1 ronda que calcule exactamente la degenerancia debe enviar un mensaje de largo $\Omega(n)$. En el mismo modelo, se construye un protocolo aleatorio de 2 rondas con mensajes de largo $\mathcal{O}(\log^2 n)$ que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia para grafos α -densos. Finalmente, se construye un protocolo determinista que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia en $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$.

Como segunda parte de este trabajo, y motivado por el protocolo en BCAST que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia, se estudia la siguiente dinámica sobre grafos: *Durante cada iteración, eliminar todos los vértices que tengan grado a lo más el grado promedio del grafo +1*. Se conjetura que para todo grafo G de n vértices la dinámica toma $\mathcal{O}(\log n)$ iteraciones en vaciar el grafo. Se aborda el problema estudiando clases de grafos tales como: bosques, grafos planares, grafos con degenerancia acotada y grafos unión disjunta de cliques.

Finalmente, se estudian diversos problemas en el modelo BCAST. Se comienza estudiando el problema de calcular el conjunto independiente maximal con un vértice fijo. Se prueba que el problema es difícil en 1 ronda y luego se construye un protocolo que en $\mathcal{O}(\log n)$ rondas, usando mensaje de largo $\mathcal{O}(\log n)$, calcula el conjunto independiente. Se estudia también el problema de calcular el número cromático de un grafo. Se prueba que el problema resulta difícil en 1 ronda. Concluyendo el capítulo, se estudian los problemas de encontrar conjuntos dominantes de tamaño k y conjuntos ℓ -dominantes de tamaño k , en ambos casos se demuestra que los problemas son difíciles en 1 ronda.

A mi familia y amigos

Agradecimientos

Quisiera agradecer a toda la gente que me acompañó en este gran viaje a través de la Universidad de Chile que finalmente culminó en este trabajo de Tesis. Quiero agradecer a mi familia que me brindó todo su apoyo. A mi Mamá, que sin sus almuerzos no hubiese sobrevivido las largas tardes en la U y sin su apoyo no hubiese llegado a donde hoy estoy parado. A mi Papá, que a pesar que hubo periodos en los cuales no nos veíamos mucho, siempre estuvo preocupado e intentó brindarme lo que más pudo.

Quiero agradecer a todos mis amigos y compañeros de generación con los que compartí momentos inolvidables. A Álvaro, Emilio y al Bob que hicieron mis primeros años en el DIM maravillosos y me enseñaron que estudiando todo se puede. A Pancho Arana por ser el más genio de los genios. A Gianmarco por ser el italiano más metalero que he conocido. Al Ocho, a Garrido, a Patito, a Pablo, al Chopo (mi amigo el calvo) por todos los buenos momentos que vivimos.

También quiero agradecer a todos los cabros del DIM que conocí. A Calisto por todas las pizzas que engullimos, a Hasson por todos los maestros, a Martín por todas las risas locas que se mandó, al flaco Cabezas, al Coba, a Rodolfo y Mati Pavez por ser unos secos. A Nicolás Sanhueza por ser el mejor auxiliar y discreto que ha pasado por el DIM.

Además quiero agradecer a todos los funcionarios del DIM, con especial mención a Eterin y Karen por ser las secretarias estrella de la carrera y a Natacha por ser la mejor subjefa docente que ha pasado por el DIM.

Quisiera agradecer a mi profesor guía Iván Rapaport, su preocupación, buena onda y pizarras y pizarras de ideas, hicieron que este trabajo fuera posible. A los miembros de la comisión Martín Matamala por dictar excelentes cursos discretos y Hiệp Hàn que logró reencantarme con la belleza de las matemáticas. También quiero agradecer a Marcos Kiwi que motivó mi elección por el área discreta del DIM.

Finalmente, quiero agradecer al proyecto Fondecyt 1130061 que financió este trabajo de tesis.

Tabla de Contenido

Introducción	1
1. Preliminares	4
1.1. Definiciones básicas	4
1.2. Los modelos distribuidos	7
2. El problema de la degenerancia	11
2.1. DEG en UCAST	11
2.1.1. El lema de aproximación	11
2.1.2. Protocolo para una $(1+\varepsilon)$ -aproximación con $b = \mathcal{O}(\log n)$ y $R = \mathcal{O}(\log n)$	14
2.2. DEG en BCAST	16
2.2.1. En una ronda DEG es difícil	16
2.2.2. Protocolo para una $(1+\varepsilon)$ -aproximación con $b = \mathcal{O}(\log^2 n)$ de 2 rondas en grafos α -densos	18
2.2.3. Protocolo para una $(2+\varepsilon)$ -aproximación con $b = \mathcal{O}(\log n)$ y $R = \mathcal{O}(\log n)$	19
3. Dinámica en el grafo	21
3.1. La dinámica	21
3.1.1. Una 2-aproximación de la degenerancia	21
3.2. Tiempo de la dinámica	22
3.2.1. La dinámica en los bosques	23
3.2.2. La dinámica en los grafos planares	24
3.2.3. La dinámica en grafos con degenerancia k	25
3.2.4. Límites de acotar con el grado promedio	26
3.2.5. La dinámica en unión disjunta de cliques	26
4. Otros resultados en el modelo BCAST	31
4.1. El problema del MIS_x	31
4.1.1. MIS_x en modelo determinista	31
4.1.2. MIS_x en modelo aleatorio	32
4.1.3. Un protocolo aleatorizado en $\mathcal{O}(\log n)$ rondas en BCAST con mensajes de largo $\mathcal{O}(\log n)$	33
4.2. Colorabilidad	37
4.3. Conjuntos dominantes	38
4.3.1. Conjuntos ℓ -dominantes	41
Conclusión	42

Índice de Ilustraciones

1.	Representación del modelo <i>Congested Clique</i>	2
1.1.	El grafo de Petersen	4
1.2.	Ejemplo de direccionamiento	5
1.3.	Ejemplo de $G * H$	5
1.4.	Ejemplo de red distribuida	8
2.1.	Ejemplo de $G \cup (F * K_{\alpha n})$	16
3.1.	Ejemplo de camino que en que la dinámica toma $n/2$ iteraciones	22
3.2.	Ejemplo de árbol que alcanza la cota de tiempo	24
3.3.	Ejemplo que demuestra que en 1 iteración de la dinámica se puede eliminar un número constante de vértices	27
4.1.	Ejemplo de $G_{u,v}^k$ con $k = 3$	38
4.2.	Ejemplo de G_F^k con $k = 3$	40
4.3.	Ejemplo de $G_F^{k,\ell}$ con $k = 3$ y $\ell = 2$	41

Introducción

Motivación

Informalmente, un sistema distribuido es una red de entidades, por ejemplo computadores, que intercambian información usando los canales de la red. Un ejemplo de sistema distribuido es la humanidad, las personas corresponden a las entidades del sistema y los distintos medios de comunicación a los canales de la red. Los primeros ejemplos computacionales de sistemas distribuidos, comienzan a aparecer en los años 60 con el estudio y posterior creación de las redes locales de computadores. ARPANET¹, precursora de internet y creada a finales de los 60, probablemente constituye el primer ejemplo de sistema distribuido en computación a gran escala. Las redes sociales, internet e incluso el proyecto SETI@home², perteneciente a la Universidad de California en Berkeley, constituyen ejemplos actuales de sistemas distribuidos masivos.

Resolver problemas en sistemas distribuidos presenta múltiples dificultades que en general no aparecen en el modelo usual de computación. Primero, cada entidad del sistema, en un comienzo, sólo conoce su propia entrada, es completamente ignorante de la información que conocen los otros miembros de la red. Segundo, cada entidad sólo puede comunicarse directamente con aquellas entidades conectadas en la red. Finalmente, y no menos importante, si la red es muy grande, no es factible que cada entidad envíe mensajes muy grandes, por ejemplo los canales podrían tener ancho de banda acotado. Evidentemente existen otros problemas como son la asincronía al momento de comunicarse o de procesar las respuestas que dependen de cada entidad, o incluso y aún más trágico, puede ocurrir que alguna entidad falle.

La computación distribuida [22],[29], es la rama de las ciencias de la computación que se encarga de estudiar distintos modelos distribuidos y cómo diversos problemas pueden ser resueltos en estos modelos. En el modelo usual de computación, el tiempo de computo de un algoritmo resulta una buena medida de calidad. En computación distribuida, las medidas de calidad de los protocolos suelen variar ampliamente de modelo en modelo. Por ejemplo, en el modelo DataStream [27], las medidas de calidad pueden ser el número de rondas necesarias o el espacio utilizado durante la computación. Por otro lado, en el modelo *Congested Clique* [12][13][19][23][25][28], medidas como el espacio ocupado no tienen sentido por la ilimitada capacidad de computo de los procesadores, sin embargo, el volumen de datos enviados por

¹Advanced Research Projects Agency Network

²Search for Extraterrestrial Intelligence at home

cada canal resulta una buena medida de la calidad de un protocolo en *Congested Clique*.

Este trabajo se focaliza en el estudio de la degenerancia de grafos en el modelo de computación distribuida *Congested Clique*. El modelo *Congested Clique* está conformado por una red completa de n procesadores. De manera cooperativa, enviando mensajes por los canales de la red, los procesadores intentan resolver un problema global. Al ser una red completa, típicamente el *Congested Clique* se estudia de dos formas diferentes, denotadas en este trabajo como UCAST y BCAST. En UCAST, cada procesador puede enviar distintos mensajes por cada uno de sus canales. Por otra parte, en BCAST, cada procesador está obligado a enviar el mismo mensaje por cada uno de sus canales.

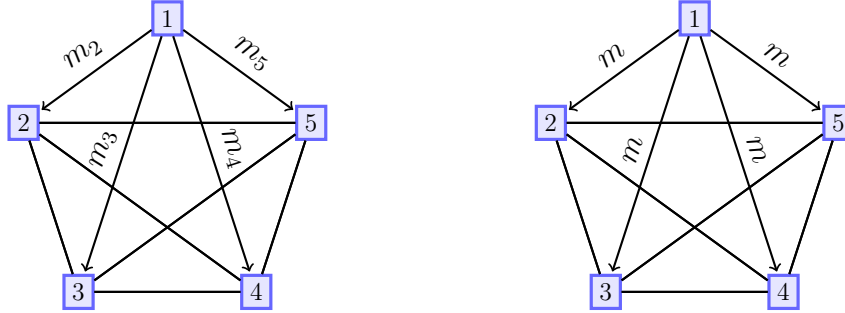


Figura 1: Representación del modelo *Congested Clique*. Al lado izquierdo, se ejemplifica la comunicación en UCAST, el procesador «1» envía cuatro mensajes que pueden ser distintos. Al lado derecho, se ejemplifica la comunicación en BCAST, el procesador «1» solo puede enviar cuatro mensajes iguales al resto de los procesadores.

La *degenerancia* de un grafo G , denotada $d(G)$, es el número natural más pequeño tal que cada subgrafo de G tiene un vértice de grado a lo más d [8][9][17]. Por ejemplo, la degenerancia de los árboles no triviales es 1, y la degenerancia de los grafos planares es a lo más 5. La degenerancia de un grafo G es un parámetro fundamental de G y ha sido reintroducida bajo diferentes nombres tales como *width* [16] o *linkage* [21] y surge en muchos resultados combinatoriales. La degenerancia $d(G)$ está dentro de un factor constante de la *arboricidad* $\alpha(G)$ de un grafo [11], donde $\alpha(G)$ es el mínimo número de bosques disjuntos necesarios para cubrir todas las aristas de G .

Organización

En el primer capítulo, se introducen los conceptos necesarios para el desarrollo de la tesis. Se definen conceptos básicos de la teoría de grafos y además se introduce el concepto de *degenerancia* de un grafo. Se describen algunas caracterizaciones clásicas de la degenerancia. Luego, se define formalmente el modelo *Congested Clique* y sus dos formas de estudio, el UCAST y BCAST.

En el segundo capítulo, se estudia el problema de calcular la degenerancia de un grafo en los modelos UCAST y BCAST. En el modelo UCAST, usando una técnica de muestreo, se presenta un protocolo aleatorio que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia en $\mathcal{O}(\log n)$ rondas, con largo de mensaje $\mathcal{O}(\log n)$. En el modelo BCAST, se prueba que en

una ronda calcular la degenerancia resulta difícil. Más específicamente, se prueba que todo protocolo aleatorio de una ronda que calcule la degenerancia, debe enviar un mensaje de largo $\Omega(n)$. A continuación, se diseña un protocolo aleatorio, de dos rondas, que usa largo de mensaje $\mathcal{O}(\log^2 n)$ y que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia cuando la entrada es un grafo α -denso. Finalmente, se explicita un protocolo determinista, de $\mathcal{O}(\log n)$ rondas, que usa un mensaje de largo $\mathcal{O}(\log n)$ para calcular una $(2 + \varepsilon)$ -aproximación de la degenerancia para cualquier grafo de n vértices.

En el tercer capítulo, se estudia una dinámica sobre grafos que iterativamente elimina del grafo aquellos vértices que tienen grado a lo más el grado promedio más uno. Se conjetura que el número de iteraciones, para que la dinámica vacíe el grafo, es $\mathcal{O}(\log n)$. Se demuestra que la conjetura es cierta en bosques, grafos planares, grafos con degenerancia acotada y grafos que son unión disjunta de cliques.

En el cuarto capítulo, se estudian otros problemas de grafos en el modelo BCAST. Primero, se estudia el problema de calcular un conjunto independiente maximal que contenga a un vértice fijo. Se prueba que el problema es difícil en una ronda. Adicionalmente, se construye un protocolo aleatorio que en $\mathcal{O}(\log n)$ rondas, con mensaje de largo $\mathcal{O}(\log n)$, calcula el conjunto independiente. Como segundo problema, se prueba que calcular el número cromático de un grafo es difícil en una ronda. Finalmente, se prueba que identificar si un grafo contiene un conjunto dominante de tamaño k es difícil en una ronda. Este último resultado se extiende a conjuntos ℓ -dominantes.

Capítulo 1

Preliminares

Este capítulo está dedicado a introducir las definiciones, notaciones y herramientas que se utilizarán en el resto de este trabajo. Se comienza entregando definiciones básicas de grafos con el fin de fijar notación, se introduce además la definición de degenerancia de un grafo y se entregan ciertas caracterizaciones de ella. Finalmente, se introducen los modelos de computación distribuida UCAST y BCAST que se utilizarán en el resto de este trabajo.

1.1. Definiciones básicas

Un *grafo simple no dirigido*, de ahora en adelante un *grafo*, es un par de conjuntos $G = (V, E)$ donde V es un conjunto finito, llamado conjunto de *vértices*, y E es un subconjunto de los 2-conjuntos de V , llamado conjunto de *aristas*. Se denota por n_G al número de vértices de G y m_G al número de aristas de G y cuando se subentienda el grafo G se evitará escribir los subíndices.

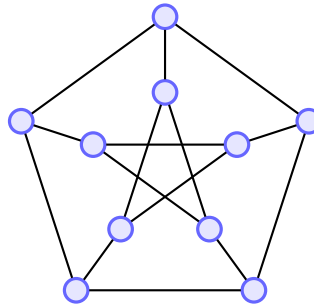


Figura 1.1: Un ejemplo de un grafo, el famoso grafo de Petersen. Los puntos azules representan los vértices y las líneas entre un par de vértices representan una arista.

Un *grafo dirigido* es un par de conjuntos $G = (V, E)$ donde V es un conjunto finito y $E \subseteq V \times V$. Dado un grafo simple no dirigido $G = (V, E)$, una *orientación* de G es un grafo dirigido $G' = (V, E')$ con $E' \subseteq \{(u, v) \in V \times V : uv \in E\}$ y tal que $uv \in E$ implica que o bien $(u, v) \in E'$ o bien $(v, u) \in E'$.



Figura 1.2: A la izquierda el grafo simple K_4 , a la derecha uno de sus direccionamientos.

Sea $G = (V, E)$ un grafo y $U \subseteq V$, se denota por $G_U = G[U]$ al *subgrafo de G inducido* por los vértices de U , en otras palabras para todo $u, v \in U$ la arista uv pertenece a G_U si y solo si uv pertenece a G . Las definiciones de grado d_G , grado mínimo δ se dan por entendidas, para más detalles ver el capítulo 1 de [11]. Sean H y G grafos vértice-disjuntos, se denota por $G * H$ al grafo con vértices $V' = V(G) \cup V(H)$ y aristas $E' = E(G) \cup E(H) \cup \{uv\}_{u \in G, v \in H}$.

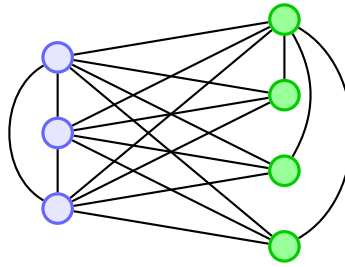


Figura 1.3: Ejemplo de $G * H$. $G = K_3$ a la izquierda en color azul, H una estrella a la derecha en color verde.

Un grafo se dice *k -degenerado* si existe un vértice v de grado a lo más k y el grafo $G - v$ sigue teniendo esta propiedad. Es claro que todo grafo G es $(n - 1)$ -degenerado. El k más pequeño tal que G es k -degenerado se llama la *degenerancia* de G , denotada $d(G)$. La degenerancia fue introducida por Lick y White en [24] y ampliamente estudiada en [8]. La degenerancia es usada como una medida de esparcidad de un grafo. Por ejemplo los árboles y bosques son 1-degenerados y los grafos planares son 5-degenerados, por otro lado el grafo más denso o con mayor cantidad de aristas posibles, el grafo completo K_n , resulta ser $(n - 1)$ -degenerado.

Matula y Beck en [26] describen un algoritmo que en tiempo lineal calcula la degenerancia de un grafo, a continuación se esboza el algoritmo:

Algoritmo 1: Calcula la degenerancia

Input: G grafo
Result: degenerancia de G

- 1 $G' \leftarrow G, k \leftarrow 0$
- 2 **while** $G' \neq \emptyset$ **do**
- 3 v vértice de grado mínimo en G'
- 4 **if** $d_{G'}(v) > k$ **then**
- 5 $k \leftarrow d_{G'}(v)$
- 6 $G' \leftarrow G' - v$

Un *ordenamiento* de los vértices de G es una tupla $\vec{x} = (v_1, \dots, v_n)$ que define una orien-

tación de las aristas de G , denotado $G_{\vec{x}}$, donde

$$V(G_{\vec{x}}) = \{v_1, \dots, v_n\} = V(G),$$

y

$$(v_j, v_k) \in E(G_{\vec{x}}) \iff v_j v_k \in E(G) \text{ y } j < k,$$

notar que el grafo $G_{\vec{x}}$ es acíclico, es decir, no tiene un ciclo dirigido. La relación entre degenerancia y ordenamiento queda resumida en la siguiente proposición.

Proposición 1.1 Sea G un grafo simple no dirigido. Entonces

$$d(G) = \max_{H \subseteq G} \delta(H) = \min_{\substack{\vec{x}=(v_j)_j \\ \text{ordenamiento}}} \max_j d_{\vec{x}}^+(v_j),$$

$d_{\vec{x}}^+(v)$ es el grado de salida del vértice v en el grafo dirigido $G_{\vec{x}}$.

Demostración. Sea $k = d(G)$. La primera igualdad se deduce del algoritmo de Matula y Beck, en alguna iteración $\delta(G') = k = d(G)$. Para la segunda igualdad, sea el ordenamiento $\vec{x} = (v_1, \dots, v_n)$ donde v_i es el i -ésimo vértice removido por el algoritmo de Matula y Beck. Entonces por las condiciones del algoritmo

$$\max_i d_{\vec{x}}^+(v_i) \leq k$$

y así $d(G) \geq \min_{\vec{x}} \max_i d_{\vec{x}}^+(v_i)$. Para la otra desigualdad suponga que $\min_{\vec{x}} \max_i d_{\vec{x}}^+(v_i) < k$, entonces existe un ordenamiento $\vec{x} = (v_1, \dots, v_n)$ tal que para todo i

$$d_{\vec{x}}^+(v_i) < k.$$

Considere los siguientes subgrafos

$$\begin{aligned} G_1 &= G \\ G_{i+1} &= G_i - v_i \end{aligned}$$

entonces $d_{\vec{x}}^+(v_i) = d_{G_i}(v_i)$ para todo i . Por la igualdad ya probada, sea F el subgrafo tal que $k = \delta(F)$ y sea i tal que $G_i \supseteq F$ y $G_{i+1} \not\supseteq F$. Entonces $v_i \in F$ y así

$$k = \delta(F) \leq d_F(v_i) \leq d_{G_i}(v_i) = d_{\vec{x}}^+(v_i) < k,$$

una contradicción. □

Un ordenamiento \vec{x} de G se dice un *ordenamiento degenerado* si $d(G) = \max_j d_{\vec{x}}^+(v_j)$. Es fácil ver que el orden en que son eliminados los vértices por el algoritmo de Matula y Beck es un ordenamiento degenerado.

Como se mencionó anteriormente la degenerancia de un grafo es utilizada para medir la densidad de un grafo. Otra medida de densidad de grafos es la *arboricidad*, definida como el mínimo número de bosques arista-disjuntos necesarios para particionar el grafo. Un grafo con muchas aristas requerirá una gran cantidad de bosques. Se denota por $a(G)$ a la arboricidad de G . La relación entre la arboricidad y la degenerancia queda plasmada en la siguiente proposición

Proposición 1.2 $a(G) \leq d(G) \leq 2a(G) - 1$.

Demostración. Sea \vec{x} un ordenamiento degenerado de G . Entonces el máximo grado de salida de $G_{\vec{x}}$ es $d(G)$. Para cada bosque F_i se elige una arista de cada vértice por arco de salida del ordenamiento, a lo más hay $d(G)$ de estos bosques y por lo tanto $a(G) \leq d(G)$.

Sea $F_1, \dots, F_{a(G)}$ una partición en bosques de G , entonces

$$m_G = \sum_{i=1}^{a(G)} m_{F_i} \leq a(G)(n-1).$$

Por lo tanto el grado promedio de G cumple $\hat{d}_G \leq 2a(G) - 2a(G)/n$ lo que implica la existencia de un vértice de grado a lo más $2a(G) - 1$. Además, esta partición en bosques induce una partición en bosques para todo subgrafo de G , por lo tanto G es $(2a(G) - 1)$ -degenerado, lo que termina de probar la proposición. \square

Otro parámetro de grafos ampliamente utilizado corresponde al *grosor* de un grafo. El *grosor* de G , denotado $\theta(G)$, corresponde al mínimo número de grafos planares necesarios para cubrir G . Una demostración similar a la realizada anteriormente demuestra que $\theta(G) \leq d(G) \leq 6\theta(G)$.

Se denotará por \mathbb{P}_X a la de probabilidad sobre el espacio X y cuando no haya ambigüedad sobre el espacio, simplemente se anotará \mathbb{P} . En probabilidades es útil estimar cuánto se desvía una variable aleatoria de su promedio, herramientas clásicas como la desigualdad de Markov o la desigualdad de Chebyshev entregan cotas para la probabilidad de las colas en variables aleatorias generales. La siguiente proposición entrega cotas exponencialmente decrecientes sobre las colas de la suma de variables aleatorias de tipo Bernoulli independientes. Para una demostración de la proposición ver [4].

Proposición 1.3 (Cotas de Chernoff) Sean X_1, \dots, X_n variables aleatorias independientes tales que $\mathbb{P}[X_i = 1] = p_i$. Sea $X = \sum_{i=1}^n X_i$ y $\mu = \mathbb{E}[X] = \sum_{i=1}^n p_i$. Entonces para todo $\varepsilon \geq 0$ se cumplen las siguientes desigualdades

- $\mathbb{P}[X \geq (1 + \varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{2 + \varepsilon}\mu\right)$
- $\mathbb{P}[X \leq (1 - \varepsilon)\mu] \leq \exp\left(-\frac{\varepsilon^2}{2}\mu\right)$

1.2. Los modelos distribuidos

Un sistema distribuido [22],[29], es un conjunto de n entidades conectadas mediante una red. Estas entidades se comunican mediante mensajes a través de los canales de la red con el fin de calcular una función $f(x_1, \dots, x_n)$, donde x_j es la entrada de la j -ésima entidad.

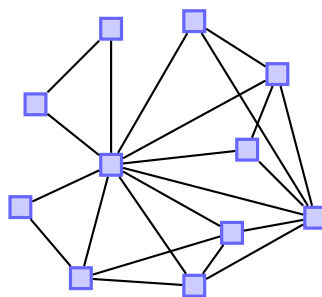


Figura 1.4: *Un sistema distribuido se puede representar como un grafo. Las entidades del sistema están representadas por los vértices cuadrados y solo pueden comunicarse con aquellas entidades con las cuales tienen una conexión representada por una arista.*

El modelo de computación distribuida *LOCAL*[29], consiste de una red representada por un grafo simple de n vértices. Cada vértice representa una entidad que inicialmente tiene un identificador de largo $\mathcal{O}(\log n)$. Cada entidad en la red tiene capacidad ilimitada de cómputo y además, cada una es capaz de enviar mensajes de largo arbitrario a sus vecinos de la red. Las distintas entidades se comunican de manera síncrona en rondas. Inicialmente, todos conocen su identificador, el tamaño de la red, n , y su parte de la entrada. Mediante los mensajes las entidades colaboran para determinar algún problema global f . El protocolo distribuido termina cuando todas las entidades conocen la respuesta. Este modelo se focaliza en estudiar el efecto de la localidad en la ejecución de protocolos distribuidos, dejando de lado factores como la congestión de los canales de comunicación y la asincronía de cómputo de cada entidad. Cuando se restringe el largo de los mensajes a $\mathcal{O}(\log n)$, se obtiene el llamado modelo de computación distribuida *CONGEST*. Este último modelo, se focaliza en estudiar cómo afecta el *volumen* de la comunicación enviada por cada canal.

El modelo de computación distribuida *Congested Clique* [12][13][19][23][25][28], consiste en un conjunto de n entidades conectadas a través de una red completa. Formalmente, el modelo consiste en una red completa descrita por el grafo K_n . Cada entidad de la red tiene un identificador distinto de largo $\mathcal{O}(\log n)$. Al ser una red completa, dos tipos de comunicación son usualmente considerados: uno a uno y uno a todos, dependiendo del tipo de comunicación llamaremos al *Congested Clique* *UCAST*¹ o *BCAST*², respectivamente. Más formalmente, en *UCAST* cada entidad por ronda puede enviar diferentes mensajes de largo b a través de sus conexiones. Por otro lado, en *BCAST* cada entidad por ronda debe enviar un mensaje de largo b a cada una de sus conexiones. Al inicio de la computación, cada entidad conoce su identificador, el tamaño de la red, n , y la entrada que le corresponde. Las distintas entidades colaboran mediante mensajes para calcular la solución. El protocolo termina una vez que todas las entidades conocen la solución. Los protocolos pueden ser de tipo determinista o de tipo aleatorio con moneda pública. Más precisamente, cada entidad tiene acceso a una única palabra de largo infinito de bits aleatorios. Los protocolos aleatorios pueden entregar una respuesta equivocada con probabilidad a los más $\varepsilon > 0$. El modelo *Congested Clique* se origina como una restricción del modelo *CONGEST* a una red completa con el fin de estudiar el efecto de la congestión y evitar el efecto de los cuellos de botella.

¹En computación la transmisión de tipo *Unicast* consiste en enviar mensajes 1 a 1 a través de la red.

²En computación la transmisión de tipo *Broadcast* consiste en enviar el mismo mensaje a todos los posibles destinatarios.

En esta tesis, se está interesado en estudiar funciones de grafos en el modelo *Congested Clique*. Por ejemplo, la propiedad «ser un árbol»,

$$\mathcal{P} = \{G : G \text{ es árbol}\},$$

para n entidades, formalmente se estudia como la función $f_{\mathcal{P},n} : \mathcal{G}_n \subseteq \{0,1\}^{n \times n} \rightarrow \{0,1\}$ donde \mathcal{G}_n es el conjunto de matrices de adyacencia de todos los grafos de n vértices. La función $f_{\mathcal{P},n}(G)$ es 1 si y sólo si G es un árbol. La entrada de la j -ésima entidad es la vecindad del vértice j del grafo de entrada, codificado como un vector de largo n . En otras palabras, cada entidad toma el rol de un vértice en el grafo G , es por esta razón que en la literatura es usual llamar a las entidades vértices, y para no romper la tradición se realizará lo mismo en este trabajo.

De ahora en adelante, $R_{\mathcal{P}}$ denotará al *número de rondas* del protocolo \mathcal{P} y $b_{\mathcal{P}}$ denotará el *largo del mensaje del protocolo* \mathcal{P} , definido como el tamaño del mensaje más largo que debe enviar algún vértice en alguna ronda del protocolo \mathcal{P} . Cuando se subentienda el protocolo se evitará escribir el subíndice. Sea \mathcal{H} un problema de grafos, para $b = b(n)$ se define la *complejidad de rondas* del problema \mathcal{H} como el mínimo número de rondas que requiere un protocolo para resolver \mathcal{H} , enviando, en cada ronda, mensajes de largo a lo más b . Análogamente, si $R = R(n)$, se define la *complejidad comunicacional* del problema \mathcal{H} como el mínimo largo de mensaje de un protocolo que resuelva \mathcal{H} en, a lo más, R rondas.

Como cada entidad recibe la vecindad de un vértice del grafo, la complejidad de rondas y la complejidad comunicacional están acotadas por n . En efecto, con n rondas o con mensajes de largo n , se puede reconstruir la matriz de adyacencia del grafo y por ende decidir cualquier problema gracias al poder ilimitado de cómputo de las entidades.

Notar que el número de bits intercambiados en cada ronda es $\Theta(bn^2)$, i.e., el modelo no tiene «cuellos de botella» [23], incluso cuando $b = 1$. A la fecha, no se conocen cotas inferiores en UCAST. En efecto, Drucker et al en [13], demostraron que en este modelo es posible simular clases de circuitos. Por lo tanto, encontrar cotas inferiores en UCAST implicaría encontrar cotas inferiores en clases de circuitos.

El modelo UCAST tiene la capacidad de realizar «balance de carga» eficientemente. Tal poder ha permitido a algunos autores diseñar protocolos eficientes. Por ejemplo, encontrar un 3-ruling set se puede realizar con complejidad de rondas $\mathcal{O}(\log \log \log n)$ [19]; contar triángulos, contar ciclos de largo 4 y calcular la cintura de un grafo tiene complejidad de rondas $\mathcal{O}(n^{0,158})$ [10]; detectar ciclos de largo 4 tiene complejidad de rondas $\mathcal{O}(1)$ [10]; construir un árbol generador mínimo tiene complejidad de rondas $\mathcal{O}(\log \log \log n)$ [18].

El modelo BCAST consiste en el clásico modelo *Number-In-Hand con pizarra compartida* [1][2][5][7][13] pues el mensaje enviado por cada entidad es el mismo que para todos, lo cual resulta análogo a que cada entidad escriba su mensaje en una pizarra, y por ende todos sean capaces de observar los mensajes escritos allí. Más aún, cuando se restringe R a 1 entonces BCAST se suele llamar *modelo con referee*, pues es análogo a que cada entidad le envíe el mensaje a una entidad todopoderosa llamada el *referee* y que finalmente éste decida la solución del problema. En este marco, el número de bits transmitidos por ronda decrece de $\Theta(bn^2)$ a $\Theta(bn)$. Por lo tanto, es posible obtener cotas inferiores usando reducciones desde

la complejidad comunicacional entre dos agentes. Por ejemplo, detectar determinísticamente un triángulo en el grafo de entrada G requiere $\Omega(n/(e^{\mathcal{O}(\sqrt{\log n})}b))$ rondas [13].

Capítulo 2

El problema de la degenerancia

En [6] se prueba que reconstruir un grafo de degenerancia k en 1 ronda requiere mensaje de largo $\mathcal{O}(k^2 \log n)$. Si se restringe el análisis a grafos con degenerancia acotada el resultado dice que basta mensaje de largo $\mathcal{O}(\log n)$ para reconstruir el grafo en 1 ronda. En este capítulo se estudia el problema de calcular la degenerancia de un grafo en los modelos UCAST y BCAST. Se denotará por DEG al problema de calcular la degenerancia de un grafo G .

En el modelo UCAST se construye un protocolo aleatorio que en $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$ calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia. En el modelo BCAST se demuestra que en 1 ronda el problema resulta ser difícil en el sentido que el mensaje enviado es de largo $\Omega(n)$. En el mismo modelo se construye un protocolo aleatorio de 2 rondas que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia con mensaje de largo $\mathcal{O}(\log^2 n)$ en el caso de grafos α -densos. Finalmente, en el mismo modelo se construye un protocolo determinista que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia en $\mathcal{O}(\log n)$ rondas con mensaje de largo $\mathcal{O}(\log n)$.

2.1. DEG en UCAST

En esta sección se construirá un protocolo aleatorio en el modelo UCAST que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia del grafo de entrada en $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$. Primero se probará el lema central que permite aproximar bien el ordenamiento degenerado del grafo mediante un muestreo del mismo y en la siguiente subsección se construirá el protocolo.

2.1.1. El lema de aproximación

La técnica que se empleará en el protocolo principal consiste en tomar un «buen muestreo» H del grafo G de manera que el ordenamiento degenerado de H «preserve» el ordenamiento de G con una fracción pequeña de error. Se denotará por $G(p)$ al subgrafo $H \subseteq G$ tal que

cada arista de G pertenece a H con probabilidad p de manera independiente. El siguiente lema, demostrado en [15], prueba que existe ese muestreo para grafos con al menos $\Omega(n \log n)$ aristas. Se presenta a continuación del lema una demostración que corrige algunos detalles de la demostración original.

Lema 2.1 Sea $\varepsilon > 0$ y sea $H = G(p)$ con G un grafo de n vértices y m aristas. Sea $U \subseteq V$ y sea \hat{v} la variable aleatoria que entrega un vértice de grado mínimo en H_U , entonces

$$\mathbb{P} \left[d_{G_U}(\hat{v}) \leq \max \left\{ (1 + \varepsilon)\delta(G_U), \frac{m}{n} \right\} \right] \geq 1 - \frac{1}{n^{\Omega(1)}}$$

si $p = \Omega(n \log n/m)$.

Demostración. Sea Q_U el conjunto de candidatos malos para \hat{v} , es decir,

$$Q_U = \{x \in G_U : d_{G_U}(x) > \max\{(1 + \varepsilon)\delta(G), m/n\}\}.$$

Se probará que $\mathbb{P}[\hat{v} \in Q_U] \leq 2/n^{\Omega(1)}$. Sea $c = \varepsilon/(2 + \varepsilon)$ y sea v un vértice de grado mínimo en G_U . Se analizarán dos casos: $\delta(G_U) = d_{G_U}(v) \geq (1 - c)\frac{m}{3n}$ y $\delta(G_U) = d_{G_U}(v) < (1 - c)\frac{m}{3n}$.

Caso $\delta(G_U) = d_{G_U}(v) \geq (1 - c)\frac{m}{3n}$.

Trabajando $\mathbb{P}[\hat{v} \in Q_U]$ se obtiene

$$\begin{aligned} \mathbb{P}[\hat{v} \in Q_U] &= \mathbb{P}[\exists x \in Q_U : x \text{ tiene grado mínimo en } H_U] \\ &= \sum_{x \in Q_U} \mathbb{P}[x \text{ tiene grado mínimo en } H_U] \\ &\leq \sum_{x \in Q_U} \mathbb{P}[d_{H_U}(x) \leq d_{H_U}(v)]. \end{aligned}$$

Ahora bien, cuando $x \in Q_U$, el evento $\{d_{H_U}(x) \leq d_{H_U}(v)\}$ queda contenido en el evento $\{d_{H_U}(x) \leq (1 - c)pd_{G_U}(x)\}$ o $d_{H_U}(v) \geq (1 + c)pd_{G_U}(v)$ pues en caso contrario se cumple

$$\begin{aligned} d_{H_U}(x) &> (1 - c)pd_{G_U}(x) \\ &\geq (1 - c)p(1 + \varepsilon)d_{G_U}(v) && [x \in Q_U] \\ &= (1 + c)pd_{G_U}(v) && [(1 - c)(1 + \varepsilon) = (1 + c)] \\ &> d_{H_U}(v) \end{aligned}$$

lo cual contradice $d_{H_U}(x) \leq d_{H_U}(v)$. Notando que las variables aleatorias $d_{H_U}(u)$ son binomiales de parámetro $(d_{G_U}(u), p)$ se puede utilizar cotas de Chernoff 1.3

$$\begin{aligned} \mathbb{P}[\hat{v} \in Q_U] &\leq \sum_{x \in Q_U} \mathbb{P}[d_{H_U}(x) \leq (1 - c)pd_{G_U}(x)] + \mathbb{P}[d_{H_U}(v) \geq (1 + c)pd_{G_U}(v)] \\ &\leq n \left[\exp \left(-\frac{c^2}{2}pd_{G_U}(x) \right) + \exp \left(-\frac{c^2}{2 + c}pd_{G_U}(v) \right) \right] \\ &\leq n \left[\exp \left(-\frac{c^2}{2}p\frac{m}{n} \right) + \exp \left(-\frac{c^2}{2 + c}p\frac{(1 - c)m}{3n} \right) \right] \\ &\leq 2n \exp \left(-\frac{c^2}{2 + c}p\frac{(1 - c)m}{3n} \right) \end{aligned}$$

imponiendo que este último número sea menor a $2/n^k$, con $k \geq 2$, se obtiene que

$$p \geq \left(\frac{3}{1-c}\right) \left(\frac{2+c}{c^2}\right) \frac{(k+1)n \log n}{m}.$$

Caso $\delta(G_U) = d_{G_U}(v) < (1-c)\frac{m}{3n}$.

Realizando el mismo cálculo que en el caso anterior

$$\mathbb{P}[\hat{v} \in Q_U] \leq \sum_{x \in Q_U} \mathbb{P}[d_{H_U}(x) \leq d_{H_U}(v)].$$

Ahora bien, el evento $\{d_{H_U}(x) \leq d_{H_U}(v)\}$ está contenido en el evento $\{d_{H_U}(x) \leq (1-c)pd_{G_U}(x) \text{ o } d_{H_U}(v) \geq (1-c)pd_{G_U}(x)\}$. Entonces aplicando cotas de Chernoff 1.3

$$\begin{aligned} \mathbb{P}[\hat{v} \in Q_U] &\leq \sum_{x \in Q_U} \mathbb{P}[d_{H_U} \leq (1-c)pd_{G_U}(x)] + \mathbb{P}[d_{H_U}(v) \geq (1-c)pd_{G_U}(v)] \\ &\leq \sum_{x \in Q_U} \exp\left(-\frac{c^2}{2}pd_{G_U}(x)\right) + \exp\left(-\frac{\delta^2}{2+\delta}pd_{G_U}(v)\right) \end{aligned}$$

donde

$$\delta = (1-c)\frac{d_{G_U}(x)}{d_{G_U}(v)} - 1,$$

y este número resulta positivo puesto que $x \in Q_U$ implica $d_{G_U}(x) > (1+\varepsilon)d_{G_U}(v)$ y $(1-c)(1+\varepsilon) = 1+c$, con lo cual

$$(1-c)\frac{d_{G_U}(x)}{d_{G_U}(v)} - 1 > (1-c)(1+\varepsilon) - 1 = c > 0.$$

Nombrando $A = d_{G_U}(x)$ y $B = d_{G_U}(v)$ se obtiene

$$\begin{aligned} \frac{\delta^2}{2+\delta}pd_{G_U}(v) &= \frac{\left((1-c)\frac{A}{B} - 1\right)^2}{(1-c)\frac{A}{B} + 1}pB \\ &= \frac{\left((1-c)A - B\right)^2}{(1-c)A + B}p \\ &= \left[(1-c)A - 3B + \frac{4B^2}{(1-c)A + B}\right]p \\ &= \left[\frac{B^2}{(1-c)A + B} + (1-c)A \left(1 - \frac{3B}{(1-c)A + B}\right)\right]p \\ &\geq (1-c)\frac{A}{4}p \\ &\geq (1-c)\frac{m}{4n}p. \end{aligned} \quad [x \in Q_U]$$

Juntando todo

$$\begin{aligned} \mathbb{P}[\hat{v} \in Q_U] &\leq n \left[\exp\left(-\frac{c^2}{2}\frac{m}{n}p\right) + \exp\left(-\frac{(1-c)}{3}\frac{m}{4n}p\right) \right] \\ &\leq 2n \exp\left(-\frac{c^2}{2}\frac{m}{n}p\right) \end{aligned}$$

e imponiendo que este último número sea menor a $2/n^k$, con $k \geq 2$, se obtiene

$$p \geq \frac{2}{c^2} \frac{(k+1)n \log n}{m}.$$

Eligiendo $p \geq \frac{3}{1-c} \frac{2+c}{c^2} \frac{(k+1)n \log n}{m}$ se asegura en ambos casos que $\mathbb{P}[\hat{v} \in Q_U] \leq 2/n^k$ que era lo que se quería probar. \square

Observación Notar que si $\varepsilon < 1$, entonces $3/(1-c) \leq 9/2$ y así $p = \frac{9}{2} \frac{2+c}{c^2} \frac{(k+1)n \log n}{m}$ cumple el lema.

2.1.2. Protocolo para una $(1 + \varepsilon)$ -aproximación con $b = \mathcal{O}(\log n)$ y $R = \mathcal{O}(\log n)$

Sea $H = G(p)$, se obtiene un ordenamiento degenerado de H usando el algoritmo de Matula y Beck removiendo en cada iteración el vértice de grado mínimo de H hasta que este se vacía. Sea $\vec{x} = (\hat{v}_1, \dots, \hat{v}_n)$ el ordenamiento que este algoritmo glotón entrega y sean

$$\begin{aligned} G_0 &= G \\ G_k &= G_{k-1} - \hat{v}_k \quad \text{para } k = 1, \dots, n \end{aligned}$$

Utilizando el lema 2.1 y cota de la unión

$$\mathbb{P}[\exists k = 1, \dots, n : d_{G_{k-1}}(\hat{v}_k) > \max\{(1 + \varepsilon)\delta(G_{k-1}), m/n\}] \leq n \frac{1}{n^{\Omega(1)}} = \frac{1}{n^{\Omega(1)}}$$

lo que equivale a que con probabilidad al menos $1 - 1/n^{\Omega(1)}$ se tiene que

$$d_{G_{k-1}}(\hat{v}_k) \leq \max\{(1 + \varepsilon)\delta(G_{k-1}), m/n\} \quad \text{para todo } k = 1, \dots, n.$$

Notando que

$$d_{\vec{x}}^+(\hat{v}_k) = d_{G_{k-1}}(\hat{v}_k),$$

se obtiene que (con alta probabilidad)

$$\begin{aligned} d(G) &= \min_{\vec{w} \text{ ordenamiento}} \max_k d_{\vec{w}}^+(v_k) \\ &\leq \max_k d_{\vec{x}}^+(\hat{v}_k) \\ &= \max_k d_{G_{k-1}}(\hat{v}_k) \\ &\leq \max_k \max\{(1 + \varepsilon)\delta(G_{k-1}), m/n\} \\ &\leq \max_k \max\{(1 + \varepsilon)d(G), d(G)\} \\ &= (1 + \varepsilon)d(G), \end{aligned}$$

donde se ha usado que

- $d(G) \geq m/n$ pues en todo grafo dirigido

$$m = \sum_v d^+(v) \leq n \max_v d^+(v)$$

y por lo tanto para cualquier ordenamiento \vec{y} , $\max_j d_{\vec{y}}(v_j) \geq m/n$, así tomando mínimo en el ordenamiento \vec{y} se obtiene $d(G) \geq m/n$.

- $d(G) \geq \delta(F)$ para cualquier F subgrafo de G usando la definición equivalente de degenerancia

$$d(G) = \max_{F \subseteq G} \delta(F).$$

En resumen, tomando un subgrafo H de G donde cada arista se elige con probabilidad $p = \Omega(n \log n/m)$, este tiene un ordenamiento $\hat{v}_1, \dots, \hat{v}_n$ de manera tal que el valor $\max_k d_{G_{k-1}}(\hat{v}_k)$ está entre $d(G)$ y $(1 + \varepsilon)d(G)$ con alta probabilidad.

Antes de construir el protocolo en UCAST es útil recordar la técnica de *balance de carga*. Si los n vértices conocen ℓ mensajes en total distribuidos de cualquier manera y todos de largo $\mathcal{O}(\log n)$, entonces en $2\lceil \ell/n \rceil$ rondas con mensajes de largo $\mathcal{O}(\log n)$ se pueden distribuir los ℓ mensajes. Las primeros $\lceil \ell/n \rceil$ rondas son usadas para que cada vértice quede en conocimiento de a lo más $\lceil \ell/n \rceil$ mensajes, las siguientes $\lceil \ell/n \rceil$ rondas son usadas para transmitir esos $\lceil \ell/n \rceil$ mensajes.

El protocolo consiste de dos etapas. La primera etapa, verificará cuantas aristas tiene el grafo, todos los vértices envían su grado y entonces $m_G = \frac{1}{2} \sum_v d_G(v)$. En caso de tener $\mathcal{O}(n \log n)$ aristas, entonces se continua con el grafo $H = G$ en la siguiente etapa. En caso de tener $\Omega(n \log n)$ aristas, entonces se toma el muestreo $H = G(p)$ y se continua con el grafo H en la siguiente etapa. La segunda etapa realiza un balance de carga sobre el grafo H para que al final del procedimiento todos conozcan H y puedan encontrar el ordenamiento degenerado de este.

Teorema 2.1 Existe un protocolo aleatorio UCAST de $\mathcal{O}(\log n)$ rondas y mensaje de largo $\mathcal{O}(\log n)$ para determinar una $(1 + \varepsilon)$ -aproximación de la degenerancia de grafos con n vértices con probabilidad al menos $1 - 1/n^{\Omega(1)}$.

Demostración. Sea $\varepsilon > 0$ fijo y sea $p = Cn \log n/m$ como en el lema 2.1. El siguiente protocolo en UCAST calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia del grafo de entrada.

Protocolo 1: Calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia

- 1 Todos los vértices envían su grado a todos, así todos calculan m
 - 2 **if** $m \geq Cn \log n$ **then**
 - 3 $H \leftarrow G(p)$ construido con moneda pública
 - 4 **else** $m < Cn \log n$
 - 5 $H \leftarrow G$
 - 6 Realizar balance de carga de H . Si el balance de carga requiere más de $2\lceil 3C \log n \rceil$ rondas, terminar el protocolo
 - 7 Cada vértice calcula sin comunicación el ordenamiento degenerado de H
 - 8 Cada vértice envía su propio $d_{G_{k-1}}(\hat{v}_k)$. El máximo $\max_k d_{G_{k-1}}(\hat{v}_k)$ es la aproximación
-

La correctitud del protocolo está dada por el análisis previo. Ahora se probará que el protocolo toma $\mathcal{O}(\log n)$ rondas con mensaje de largo $\mathcal{O}(\log n)$. Sea G un grafo con n vértices y m aristas. Si el grafo tiene $m < Cn \log n$ aristas, entonces el balance de carga toma $\mathcal{O}(\log n)$ rondas. Si el grafo tiene $m \geq Cn \log n$ aristas, entonces con moneda pública se obtiene el subgrafo H donde cada arista de G pertenece a H con probabilidad p . Este subgrafo tiene un número esperado de aristas

$$\mathbb{E}[m_H] = pm = Cn \log n,$$

y por lo tanto usando cotas de Chernoff

$$\mathbb{P}[m_H > 3Cn \log n] \leq \exp(-Cn \log n) \leq \frac{1}{n^{\Theta(n)}}$$

por lo que con (muy) alta probabilidad $m_F \leq 3Cn \log n$ y por lo tanto el balance de carga se realiza pues necesita a lo más $2\lceil 3C \log n \rceil$ rondas. En ambos casos se hace $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$ lo que termina de probar el resultado. \square

2.2. DEG en BCAST

2.2.1. En una ronda DEG es difícil

En [13] se prueba que en el modelo BCAST aleatorio reconocer si un grafo contiene un clique de tamaño k es difícil, incluso cuando k es una fracción de n . En esta sección se probará que cualquier protocolo aleatorio de 1 ronda en BCAST para DEG puede resolver detección de cliques grandes.

Sea G de n vértices y $F \subseteq V(G)$ no vacío, sea además $K_{\alpha n}$ disjunto de G y $\alpha \in (1/2, 1)$ de manera que αn sea un entero. Durante esta subsección se denotará por G_F al grafo definido por $G_F = G \cup (F * K_{\alpha n})$.

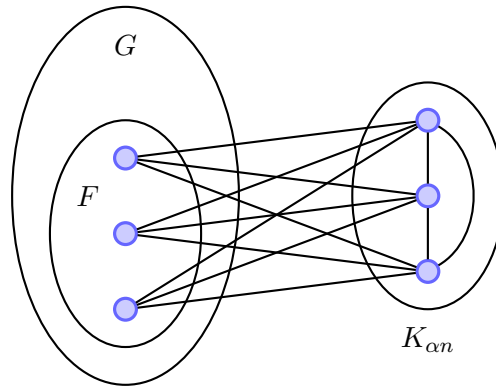


Figura 2.1: Ejemplo de G_F . A la izquierda se encuentra el grafo G , el subgrafo F está representado por los vértices azules encerrados por la elipse. A la derecha se encuentra el grafo $K_{\alpha n}$. Todos los vértices de F se encuentran unidos a todos los vértices de $K_{\alpha n}$.

Lema 2.2 Sea $F \subseteq V(G)$ de tamaño αn . Entonces $G[F] \cong K_{\alpha n}$ si y solo si $d(G_F) \geq 2\alpha n - 1$.

Demostración.

(\implies) Si $F \cong K_{\alpha n}$, entonces $G[F * K_{\alpha n}] = K_{2\alpha n}$ de lo cual se deduce que $d(G_F) \geq 2\alpha n - 1$.

(\impliedby) Si $d(G_F) \geq 2\alpha n - 1$, entonces el subgrafo $F' \subseteq G_F$ tal que $d(G_F) = \delta(F')$ debe estar contenido en $G[F * K_{\alpha n}]$, pues en caso contrario tendría un vértice en $G \setminus F$ y por ende grado a lo más $n - 1 < 2\alpha n - 1$, contradicción pues $\alpha > 1/2$. Como $\delta(F') \geq 2\alpha n - 1$, entonces $|F'| \geq 2\alpha n$ y de lo anterior se deduce que $F' = G_F[F * K_{\alpha n}]$. Además

$$d_{G_F[F * K_{\alpha n}]}(v) = \begin{cases} 2\alpha n - 1 & \text{si } v \in K_{\alpha n} \\ d_{G[F]}(v) + \alpha n & \text{si } v \in F \end{cases}$$

de lo cual se obtiene $d_{G[F]}(v) + \alpha n \geq \delta(F') \geq 2\alpha n - 1$, para todo $v \in F$, de lo cual resulta $d_{G[F]} \geq \alpha n - 1$. Como $|F| = \alpha n$ se concluye. \square

Observación Notar que el lema anterior se puede fortalecer a $G[F] \cong K_{\alpha n}$ si y solo si $d(G_F) = 2\alpha n - 1$ pues $d(G_F) \leq \Delta(G_F) \leq 2\alpha n - 1$.

En [13] se prueba que para todo $n \geq 4$ reconocer si G grafo de n vértices tiene un clique de tamaño αn es difícil para $\alpha \in (0, 1)$ en el modelo BCAST con aleatoriedad. Más específicamente se prueba que al tomar mensajes de largo b se requieren $\Omega(n/b)$ rondas para resolver el problema, este resultado implica que en 1 ronda se requiere un mensaje de largo $\Omega(n)$. El resultado de dificultad en 1 ronda sigue a continuación

Teorema 2.2 Cualquier protocolo aleatorio \mathcal{P} de 1 ronda en BCAST que resuelva DEG cumple $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Sea \mathcal{P} un protocolo de 1 ronda aleatorio en BCAST que resuelve DEG. El siguiente protocolo \mathcal{P}' resuelve detección de clique de tamaño αn con $\alpha \in (1/2, 1)$

Protocolo 2: Reconoce si G contiene $K_{\alpha n}$

- 1 Todos los vértices v escriben 2 mensajes m_v y m'_v realizados con el protocolo \mathcal{P} . El mensaje m_v corresponde al mensaje con $n + \alpha n$ vértices y vecindad $N_G(v)$. El mensaje m'_v corresponde al mensaje con $n + \alpha n$ vértices y vecindad $N_G(v) \cup \{1', \dots, (\alpha n)'\}$
 - 2 El referee verifica si la degenerancia de G_F es mayor o igual a $2\alpha n - 1$ corriendo \mathcal{P} en los mensajes $\{m_v\}_{v \notin F} \cup \{m'_v\}_{v \in F} \cup \{m_{1'}, \dots, m_{(\alpha n)'}\}$ donde $\{m_{i'}\}_{i=1}^{\alpha n}$ son los mensajes de los vértices extra $1', \dots, (\alpha n)'$ con vecindad F y $|F| = \alpha n$
 - 3 Si la degenerancia es mayor o igual a $2\alpha n - 1$, el referee concluye que hay un $K_{\alpha n}$ en G
 - 4 Si no, el referee itera sobre otro conjunto $F \subseteq V$ de tamaño αn
 - 5 Si para ningún F ocurre lo anterior se concluye que G no contiene a $K_{\alpha n}$
-

La correctitud está dada por el lema 2.2. El análisis de la complejidad comunicacional sigue de lo siguiente: el mensaje enviado por cada vértice es a lo más $2b_{\mathcal{P}}(n + \alpha n) \leq 2b_{\mathcal{P}}(2n)$

pues $\alpha < 1$ y detección de cliques de tamaño αn requiere un mensaje de largo a lo menos $\Omega(n)$. Entonces $b_{\mathcal{P}}(2n) = \Omega(n)$ lo que implica el resultado. \square

2.2.2. Protocolo para una $(1 + \varepsilon)$ -aproximación con $b = \mathcal{O}(\log^2 n)$ de 2 rondas en grafos α -densos

En la sección 2.1 se demostró que para grafos con al menos $m \geq Cn \log n$ aristas existe un «buen muestreo» que preserva el ordenamiento salvo un error ε . Más aún, si la probabilidad de elegir cada arista es $p = Cn \log n / m$ donde C es la constante del lema 2.1, entonces para todo $v \in G$

$$\mathbb{E}[d_H(v)] = pd_G(v) = C \frac{n \log n}{m} d_G(v),$$

esta igualdad motiva tomar grafos con un número de aristas m suficientemente grande de manera que el término $nd_G(v)$ se cancele. Un grafo G se dirá α -denso si $m \geq \alpha n^2$ con $\alpha \leq 1/4$ fijo. Usando la igualdad anterior en grafos α -densos se obtiene para todo $v \in G$

$$\mathbb{E}[d_H(v)] \leq \frac{C}{\alpha} \log n.$$

Sea $A > 1$, entonces la probabilidad de que algún vértice se desvíe A veces de la cota superior de la media es

$$\mathbb{P} \left[\exists v : d_H(v) > \frac{AC \log n}{\alpha} \right] \leq \sum_{\substack{v \\ d_G(v) > \frac{AC \log n}{\alpha}}} \mathbb{P}[d_H(v) > AC\alpha^{-1} \log n].$$

Aplicando Chernoff

$$\mathbb{P}[d_H(v) > AC\alpha^{-1} \log n] \leq \exp \left(-\frac{\delta^2}{2 + \delta} pd_G(v) \right),$$

donde

$$\delta = \frac{A\alpha^{-1}m}{nd_G(v)} - 1.$$

Se probará que cada exponencial es menor a $1/n^{\Omega(1)}$. Por simplicidad $d = d_G(v)$, ahora bien

$$\begin{aligned} \frac{\delta^2}{2 + \delta} pd &= \frac{\left(\frac{A\alpha^{-1}m}{nd} - 1 \right)^2}{1 + \frac{A\alpha^{-1}m}{nd}} C \frac{n \log n}{m} d \\ &= C \log n \frac{(A\alpha^{-1}m - nd)^2}{m(A\alpha^{-1}m + nd)} \\ &= C \log n \frac{(A\alpha^{-1}m + nd - 2nd)^2}{m(A\alpha^{-1} + nd)} \\ &= C \log n \left[\frac{A\alpha^{-1}m + nd}{m} - \frac{4nd}{m} + \frac{(2nd)^2}{m(A\alpha^{-1}m + nd)} \right] \\ &\geq C \log n (A - 3)\alpha^{-1} \end{aligned}$$

entonces tomando $A \geq 4$ se obtiene

$$\exp\left(-\frac{\delta^2}{2+\delta}pd_G(v)\right) \leq \exp(-\alpha^{-1}C \log n) \leq \frac{1}{n^{\Omega(1)}},$$

donde C es la constante del lema 2.1. Como cada exponencial queda acotada uniformemente se obtiene

$$\mathbb{P}\left[\exists v : d_H(v) > \frac{AC \log n}{\alpha}\right] \leq \frac{1}{n^{\Omega(1)}}.$$

Teorema 2.3 Existe un protocolo aleatorio en BCAST de 2 rondas con mensajes de largo $b = \mathcal{O}(\log^2 n)$ que calcula una $(1 + \varepsilon)$ -aproximación en grafos α -densos.

Demostración. Usando moneda pública y sin comunicación los vértices de G forman su vecindad en el subgrafo $H = G(p)$. Por lo probado anteriormente el subgrafo H tiene grado máximo $\mathcal{O}(\log n)$ con alta probabilidad. Entonces usando mensajes de largo $\mathcal{O}(\log^2 n)$ en 1 ronda cada vértice logra transmitir su vecindad en el subgrafo H y al final de este proceso todos conocen el subgrafo H . Realizando el algoritmo glotón se construye el ordenamiento degenerado de H y mediante el mismo análisis hecho para el caso del modelo UCAST se prueba que este ordenamiento encuentra una $(1 + \varepsilon)$ -aproximación de la degenerancia de G . En 1 ronda adicional se decide la aproximación de la degenerancia. \square

Observación Notar que cada vértice también puede enviar los $\mathcal{O}(\log n)$ vecinos en $\mathcal{O}(\log n)$ rondas. Por lo tanto también se tiene como resultado un protocolo en $R = \mathcal{O}(\log n)$ rondas con largo de mensaje $b = \mathcal{O}(\log n)$.

2.2.3. Protocolo para una $(2 + \varepsilon)$ -aproximación con $b = \mathcal{O}(\log n)$ y $R = \mathcal{O}(\log n)$

En [14] se construye un algoritmo determinista en el modelo de Data Stream que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia. En esta sección se adaptará dicho algoritmo para construir un protocolo determinista en BCAST que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia.

Sea G un grafo de n vértices, $\varepsilon > 0$ y sean

$$G_1 = G$$

$$G_{j+1} = G_j \setminus \{v \in G_j : d_j(v) \leq (2 + \varepsilon)m_j/n_j\}$$

donde $m_j = |E(V_j)|$ y $n_j = |V_j|$ y $d_j(v)$ es el grado de v en el grafo G_j . El protocolo iterativamente construye los grafos G_j borrando en cada iteración aquellos vértices que tienen grado a lo más $(2 + \varepsilon)m_j/n_j$.

Teorema 2.4 Existe un protocolo determinista en el modelo BCAST de $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$ para determinar una $(2 + \varepsilon)$ -aproximación de la degenerancia de grafos de n vértices.

Demostración. Sea W_j los vértices eliminados durante la j -ésima iteración. El siguiente protocolo calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia en $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$

Protocolo 3: Calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia

- 1 Todos los vértices comienzan con $n' \leftarrow n, V' \leftarrow V$
 - 2 **while** $n' \neq 0$ **do**
 - 3 Todos dicen su grado en V'
 - 4 Todos calculan m' número de aristas en V'
 - 5 $V \leftarrow V' -$ vértices en V' con grado $\leq (2 + \varepsilon)m'/n'$
 - 6 $n' \leftarrow |V'|$
 - 7 Todos conocen W_1, W_2, \dots . Se considera el ordenamiento dado por $\vec{x} = (W_j)_j$ con los vértices dentro de W_j ordenados de manera arbitraria
 - 8 Todos dicen su grado en el ordenamiento. La aproximación de la degenerancia es el grado más alto en dicho ordenamiento
-

Primero notar que el número de iteraciones resulta ser logaritmico pues en la j -ésima iteración

$$2m_j = \sum_{v \in V_j} d_j(v) \geq \sum_{\substack{v \in V_j \\ d_j(v) \geq (2+\varepsilon)\frac{m_j}{n_j}}} d_j(v) \geq n_{j+1}(2 + \varepsilon)\frac{m_j}{n_j},$$

lo que implica $n_{j+1} \leq \frac{2}{2+\varepsilon}n_j$. Por lo tanto en $\mathcal{O}(\log n)$ rondas el protocolo termina.

Ahora bien, el ordenamiento \vec{x} construido durante las iteraciones cumple que

$$\forall j, \forall v \in W_j \quad d_{\vec{x}}^+(v) \leq (2 + \varepsilon)\frac{m_j}{n_j},$$

y así el resultado final es una $(2 + \varepsilon)$ -aproximación pues

$$\begin{aligned} d(G) &= \min_{\vec{x} \text{ ordenamiento}} \max_v d_{\vec{x}}^+(v) \\ &\leq \max_v d_{\vec{x}}^+(v) \\ &\leq \max_j (2 + \varepsilon)\frac{m_j}{n_j} \\ &\leq (2 + \varepsilon)d(G), \end{aligned}$$

donde se ha usado que $d(G) \geq \frac{m_F}{n_F}$ para todo $F \subseteq V$. □

En el siguiente capítulo se estudiará una dinámica en grafos motivada principalmente por el protocolo construido.

Capítulo 3

Dinámica en el grafo

En este capítulo se estudiará la siguiente dinámica sobre grafos

En cada iteración de la dinámica se eliminan aquellos vértices de grado a lo más el grado promedio del grafo $+1$.

Esta dinámica está motivada por el protocolo que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia en el modelo **BCAST**, ver subsección 2.2.3. Esta dinámica es interesante pues permite calcular una 2-aproximación de la degenerancia del grafo. Se conjetura que la dinámica toma $\mathcal{O}(\log n)$ iteraciones en vaciar el grafo.

La dinámica es estudiada en los bosques, los grafos planares, los grafos con degenerancia acotada y los grafos que son unión disjunta de cliques.

3.1. La dinámica

Sea $G = (V, E)$ un grafo de n vértices. Se define la siguiente dinámica sobre el grafo G

$$\begin{aligned} G_0 &= G \\ G_i &= G_{i-1} \setminus \{v \in G_{i-1} : d_{i-1}(v) \leq \hat{d}_{i-1} + 1\} \end{aligned}$$

donde $d_{i-1}(v)$ es el grado del vértice v en el grafo G_{i-1} y \hat{d}_{i-1} corresponde al grado promedio del grafo G_{i-1} . Es claro que esta dinámica es finita y por lo tanto existe t tal que $G_t = \emptyset$.

3.1.1. Una 2-aproximación de la degenerancia

Sean W_i los vértices eliminados en la i -ésima iteración, esto es

$$W_i = \{v \in G_{i-1} : d_{i-1}(v) \leq \hat{d}_{i-1} + 1\},$$

y sea $\vec{x} = (\vec{y}_i)_{i=1}^t$ un ordenamiento de V donde los \vec{y}_i son un ordenamiento arbitrario de W_i , entonces

$$\forall i, \forall v \in W_i \quad d_i(v) \geq d_{\vec{x}}^+(v),$$

y por lo tanto

$$\begin{aligned} d(G) &= \min_{\vec{w}} \max_v d_{\vec{w}}^+(v) \\ &\leq \max_i \max_{v \in W_i} d_i(v) \\ &\leq \max_i 2\hat{d}_i + 1 \\ &\leq 2d(G) + 1, \end{aligned}$$

donde se ha usado que $d(G) \geq m_i/n_i$. Así que el valor $\max_i \max_{v \in W_i} d_i(v)$ corresponde a una 2-aproximación de la degenerancia de G .

3.2. Tiempo de la dinámica

Al mínimo número de iteraciones necesarios para vaciar el grafo G utilizando la dinámica, es decir,

$$t(G) := \min\{i : G_i = \emptyset\}$$

se le llamará *el tiempo de la dinámica*. Se conjetura que el tiempo de la dinámica es $\mathcal{O}(\log n)$, con n el número de vértices del grafo inicial.

Conjetura 3.1 Para todo grafo G , $t(G) = \min\{i : G_i = \emptyset\} = \mathcal{O}(\log n)$.

Observación Si en la dinámica, en cada iteración i -ésima, sólo se eliminan los vértices con grado a lo más \hat{d}_{i-1} , entonces la conjetura falla. En efecto, en el grafo camino P_n eliminar los vértices de grado a lo más el grado promedio por iteración únicamente eliminará los extremos del camino, y por lo tanto la dinámica tomará a lo menos $n/2$ iteraciones en vaciar el grafo. Es por esto que es necesario el término «+1».

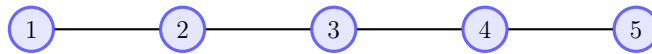


Figura 3.1: Ejemplo de un grafo que toma $3 \geq 5/2$ iteraciones en vaciarse. En la primera iteración, los vértices borrados son 1 y 5. En la segunda iteración, 2 y 4. Finalmente, es borrado 5. El caso P_n es análogo.

Notar que $t(P_n) = 1$ para todo $n \geq 1$. Un ejemplo directo de clase de grafo que cumple la conjetura son los grafos regulares, en efecto, si G es d -regular entonces $\hat{d} = d$ y por lo tanto $t(G) = 1$.

El estudio del tiempo de la dinámica se aborda para las siguientes clases de grafos: los bosques, los grafos planares, los grafos con degenerancia acotada y finalmente los grafos conformados por uniones disjuntas de cliques.

3.2.1. La dinámica en los bosques

Sea G un bosque con n vértices y c el número de componentes conexas de G . Se sabe que el número de aristas m es $n - c$ y por ende

$$\hat{d} = \frac{2(n - c)}{n} = 2 - \frac{2c}{n}.$$

Sea $n' = |\{v : d(v) > \hat{d} + 1\}|$ el número de vértices que no son eliminados en una iteración de la dinámica, entonces

$$\begin{aligned} 2m &= \sum_v d(v) \\ &\geq \sum_{d(v) > \hat{d} + 1} d(v) \\ &> n' \left(3 - \frac{2c}{n} \right), \end{aligned}$$

despejando se obtiene

$$n' \leq \left(\frac{2n - 2c}{3n - 2c} \right) n.$$

Ahora bien, para todo $n, c \geq 1$

$$\frac{2n - 2c}{3n - 2c} \leq \frac{2}{3},$$

y así

$$n' \leq \frac{2}{3}n.$$

Como cada subgrafo de un bosque sigue siendo un bosque, entonces el análisis resulta ser independiente de la iteración de la dinámica y por ende el tiempo de la dinámica resulta ser a lo más $\lceil \log_{3/2} n \rceil$. Así, la conjetura resulta ser cierta para bosques.

Se sabe que el tiempo del grafo camino P_n es 1 para todo n . A continuación, se construirá un ejemplo de árbol T tal que $t(T) = \Omega(\log n)$, de manera que la conjetura en árboles resulta ajustada.

Sea T_h el árbol donde la raíz tiene grado 3 y cada uno de sus hijos es la raíz de un árbol binario completo de altura $h - 1$, ver figura 3.2. Es claro que

$$d_{T_h}(v) = \begin{cases} 1 & \text{si } v \text{ es hoja} \\ 3 & \text{si } v \text{ no es hoja} \end{cases},$$

y como

$$\begin{aligned} \hat{d} + 1 &= 3 - 2/n \\ &< 3, \end{aligned}$$

en una iteración de la dinámica los únicos vértices borrados resultan ser las hojas. El árbol restante es T_{h-1} y el mismo argumento se puede iterar $h - 1$ veces más. El número de vértices del árbol T_h es

$$\begin{aligned} n &= \sum_{i=0}^{h-1} 3 \cdot 2^i + 1 \\ &= 3(2^h - 1) + 1 \\ &= 3 \cdot 2^h - 2 \end{aligned}$$

por lo tanto $h = \log_2((n + 2)/3)$ y así $t(T_h) = \Theta(\log n)$.

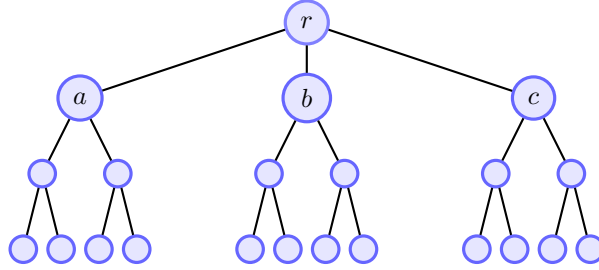


Figura 3.2: Árbol que alcanza la cota del tiempo de la dinámica. El vértice raíz r tiene 3 hijos a, b y c que son raíces de árboles binarios completos. Notar que cada vértice interno tiene grado 3.

3.2.2. La dinámica en los grafos planares

Se sabe que si G es un grafo planar con n vértices y m aristas entonces $m \leq 3n - 6$, ver [11]. Sea G un grafo planar, y n' el número de vértices tras una iteración de la dinámica, entonces

$$\begin{aligned} 2m &\geq \sum_{\substack{v \\ d(v) > \hat{d} + 1}} d(v) \\ &> n'(\hat{d} + 1) \end{aligned}$$

entonces

$$n' \leq \left(\frac{2m}{2m + n} \right) n.$$

La función

$$f(m) = \frac{2m}{2m + n},$$

tiene como derivada

$$f'(m) = \frac{2n}{(2m + n)^2}$$

siempre positiva, por lo tanto f resulta ser creciente y usando la cota $m \leq 3n - 6$ se deduce que

$$f(m) \leq \frac{6n - 12}{7n - 12},$$

pero

$$\frac{6n - 12}{7n - 12} \leq \frac{6}{7},$$

con lo cual se obtiene

$$n' \leq \frac{6}{7}n.$$

Como todo subgrafo de un grafo planar sigue siendo un grafo planar, el resultado anterior resulta ser independiente de la iteración de la dinámica. Por lo tanto el tiempo de la dinámica es a lo más $\lceil \log_{7/6} n \rceil$. La cota se alcanza para algún grafo planar pues los bosques son grafos planares.

Notar que, al igual que en el caso anterior, se usó fuertemente una cota sobre el grado promedio y que todo subgrafo de un grafo planar también resulta ser un grafo planar.

3.2.3. La dinámica en grafos con degenerancia k

Por inducción en el número de vértices se prueba que para todo grafo k -degenerado con $n \geq k$ vértices y m aristas se cumple la siguiente desigualdad:

$$m \leq kn - \binom{k+1}{2}.$$

Sea G un grafo k -degenerado con $n \geq k$ vértices y m aristas. Denotando por n' al número de vértices tras una iteración de la dinámica y usando la misma técnica que en los casos anteriores se obtiene

$$n' \leq \left(\frac{2m}{2m+n} \right) n.$$

Se sabe de la subsección pasada que la función

$$f(m) = \frac{2m}{2m+n}$$

es creciente y así usando la desigualdad anterior

$$f(m) \leq \frac{2kn - k(k+1)}{(2k+1)n - k(k+1)},$$

y este último término queda acotado por $2k/(2k+1)$. Por lo tanto

$$n' \leq \left(\frac{2k}{2k+1} \right) n.$$

Mientras G tenga $n \geq k$ vértices la dinámica cumple lo anterior pues todo subgrafo de un grafo k -degenerado es también k -degenerado. En $\lceil \log_{(2k+1)/2k}(n/k) \rceil$ iteraciones de la dinámica el grafo queda con $n < k$ vértices, pero como k es constante se obtiene que el tiempo de la dinámica sobre un grafo G k -degenerado es a lo más $k + \lceil \log_{(2k+1)/2k}(n/k) \rceil$. Por lo tanto la conjetura resulta cierta para grafos k -degenerados con k fijo.

Observación Este resultado implica de inmediato el caso de los bosques y los grafos planares pues los bosques son grafos 1-degenerados y los grafos planares son 5-degenerados.

3.2.4. Límites de acotar con el grado promedio

De las secciones anteriores se deduce que tras una iteración de la dinámica se cumple

$$n' \leq n \left(\frac{\hat{d}}{1 + \hat{d}} \right).$$

Esta cota resulta ser limitada cuando $\hat{d} \approx n$. En efecto, considere $G = K_n$. En este caso $\hat{d} = n - 1$ y la cota anterior dice que tras una iteración

$$n' \leq n \frac{n-1}{n} = n-1,$$

que es la cota trivial pues en cada iteración al menos un vértice es eliminado. Claramente esto contrasta fuertemente con el resultado real que dice que en una iteración $n' = 0$.

Si se lograra demostrar que en cada iteración se elimina una fracción constante del grafo el resultado sería cierto haciendo un análisis similar al hecho en las subsecciones anteriores. Sin embargo esto no es cierto como se ilustra en el siguiente ejemplo. Sea $G = (A \cup B, E)$ grafo bipartito con $2n$ vértices con $|A| = |B| = n$. Los elementos $a_1, \dots, a_{n-2} \in A$ se unen completamente a los elementos b_1, \dots, b_{n-2} , ver 3.3. Es claro que $m_G = (n-2)^2$, entonces el grado promedio de G es

$$\hat{d} = \frac{2(n-2)^2}{2n} = \frac{n^2 - 4n + 4}{n},$$

y por lo tanto $\hat{d} + 1 = n - 3 + \frac{4}{n} < n - 2$. Así que en 1 iteración solo se eliminan los vértices a_{n-1}, a_n, b_{n-1} y b_n , más aún

$$2n \left(\frac{\hat{d}}{1 + \hat{d}} \right) = 2n - \frac{2n^2}{n^2 - 3n + 4}$$

que para n grande resulta ser cercano a $2n - 2$, es decir, la desigualdad resulta casi ajustada salvo por una diferencia de 2.

En resumen, la desigualdad usada en las subsecciones anteriores sirve en los casos con grado promedio acotado, en los otros casos como en el caso del grafo completo no entrega información útil. Además no se puede asegurar que una fracción del grafo sea eliminada en una iteración de la dinámica como lo muestra el ejemplo anterior.

En la siguiente subsección se estudiará la dinámica en unión disjunta de cliques. Se desarrollará una nueva dinámica sobre secuencias fijas de números que permitirá medir otros parámetros que el promedio no logra cubrir.

3.2.5. La dinámica en unión disjunta de cliques

En esta subsección se estudiará la dinámica en grafos formados por la unión disjunta de cliques.

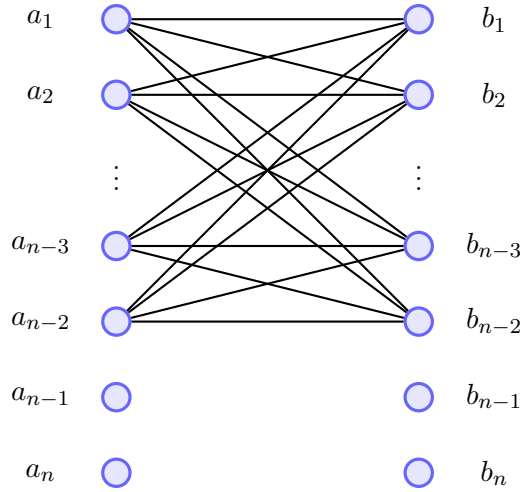


Figura 3.3: Ejemplo que demuestra que en 1 iteración de la dinámica se puede eliminar un número constante de vértices.

Sea $G = \bigcup_{i=1}^{\ell} K_{k_i}$ unión disjunta de cliques con $k_1 \leq \dots \leq k_{\ell}$. Lo primero que se observa es que en cada iteración de la dinámica cliques completos serán eliminados, más aún, los grados de los vértices restantes no se verán afectados. Esto motiva el estudio de una dinámica sobre secuencias de números, en la cual en cada iteración los elementos con valor a los más el promedio +1 sean eliminados. Formalmente, sea X una secuencia de n números, se define la siguiente dinámica sobre secuencias:

$$X_0 = X$$

$$X_i = X_{i-1} \setminus \{x \in X_{i-1} : x_i \leq \mu_{i-1} + 1\}$$

donde μ_{i-1} es el promedio de los elementos en X_{i-1} . Se define el tiempo de la dinámica como

$$t'(X) = \min\{n : X_i = \emptyset\}.$$

Teorema 3.1 Para toda secuencia X de n números, $t'(X) = \mathcal{O}(\log n)$.

Demostración. Sea $z^i = (x^i, y^i)$ los elementos tras la iteración $i - 1$ de la dinámica, donde $x^i = (x_j^i)_{j=1}^k$, $(y^i) = (y_j^i)_{j=1}^{\ell}$ y para todo j $x_j^i \leq \mu_i + 1$ y además para todo j $y_j^i > \mu_i + 1$. En otras palabras, los x^i son los elementos que se eliminarán en la iteración i y los elementos y_j son los que no.

Sean \bar{x}^i y \bar{y}^i el promedio de los elementos en x^i y el promedio de los elementos en y^i respectivamente. Sean además σ_{x^i} y σ_{y^i} la varianza de los elementos en x^i y la varianza de los elementos en y^i respectivamente, además sea σ_{z^i} la varianza total. Notar que $\mu_{i+1} = \bar{y}^i$ y $\sigma_{z^{i+1}} = \sigma_{y^i}$.

Sean $\rho_{x^i} = |x^i|/|z^i|$ y $\rho_{y^i} = |y^i|/|z^i|$ la proporción de elementos en x^i e y^i respectivamente, donde $|w|$ simboliza el número de elementos en w . Notar que $\rho_{x^i} + \rho_{y^i} = 1$, además

$$\mu_i = \rho_{x^i} \bar{x}^i + \rho_{y^i} \bar{y}^i. \tag{3.1}$$

Más aún,

$$\sigma_{z^i} = \rho_{x^i} \sigma_{x^i}^2 + \rho_{y^i} \sigma_{y^i}^2 + \rho_{x^i} \rho_{y^i} (\bar{y}^i - \bar{x}^i)^2. \quad (3.2)$$

En efecto, si $z = (x, y)$ entonces

$$\begin{aligned} \sigma_z^2 &= \frac{1}{|z|} \left(\sum_j x_j^2 + \sum_j y_j^2 \right) - \mu^2 \\ &= \frac{1}{|z|} \left(\sum_j x_j^2 + \sum_j y_j^2 \right) - (\rho_x \bar{x} + \rho_y \bar{y})^2 && [\text{por (3.1)}] \\ &= \frac{1}{|z|} \left(\sum_j x_j^2 + \sum_j y_j^2 \right) - \rho_x^2 \bar{x}^2 - 2\rho_x \rho_y \bar{x} \bar{y} - \rho_y^2 \bar{y}^2 \\ &= \frac{1}{|z|} \left(\sum_j x_j^2 + \sum_j y_j^2 \right) - \rho_x^2 \bar{x}^2 - 2\rho_x \rho_y \bar{x} \bar{y} - \rho_y^2 \bar{y}^2 - \rho_x \rho_y \bar{x}^2 + 2\rho_x \rho_y \bar{x} \bar{y} - \rho_x \rho_y \bar{y}^2 \\ &\quad + \rho_x \rho_y (\bar{y} - \bar{x})^2 \\ &= \frac{1}{|z|} \left(\sum_j x_j^2 + \sum_j y_j^2 \right) - \rho_x \bar{x}^2 - \rho_y \bar{y}^2 + \rho_x \rho_y (\bar{y} - \bar{x})^2 \\ &= \rho_x \left(\frac{1}{|x|} \sum_j x_j^2 - \bar{x}^2 \right) + \rho_y \left(\frac{1}{|y|} \sum_j y_j^2 - \bar{y}^2 \right) + \rho_x \rho_y (\bar{y} - \bar{x})^2 \\ &= \rho_x \sigma_x^2 + \rho_y \sigma_y^2 + \rho_x \rho_y (\bar{y} - \bar{x})^2. \end{aligned}$$

donde los superíndices se han evitado para alivianar la lectura.

Se dirá que una iteración i es *buen*a si $\rho_{x^i} \geq 1/8$. En caso contrario, se dirá que la iteración es *mal*a.

Sea el siguiente potencial

$$\varphi_i = |z^i| \sigma_{z^i}^2.$$

Se demostrará que $\varphi_{i+2} \leq \alpha \varphi_i$ para un cierto $\alpha \in (0, 1)$ fijo.

Si la iteración i -ésima de la dinámica es buena, entonces $\varphi_{i+1} \leq \frac{7}{8} \varphi_i$. En efecto, la iteración es buena si $\rho_{x^i} \geq 1/8$, lo cual implica que $|z^{i+1}| = |y^i| \leq \frac{7}{8} |z^i|$ y como $\sigma_{z^{i+1}} \leq \sigma_{z^i}$ pues los elementos están menos dispersos se obtiene que $\varphi_{i+2} \leq \varphi_{i+1} \leq \frac{7}{8} \varphi_i$. De manera análoga se procede si la iteración $(i+1)$ -ésima es buena.

La pregunta ahora es: ¿qué ocurre cuando hay dos iteraciones malas seguidas?. Se demostrará que en este caso la varianza disminuye una fracción constante. Se utilizará la igualdad (3.2) para acotar la varianza en la iteración $(i+2)$. Más específicamente, en

$$\sigma_{z^{i+2}} = \rho_{x^{i+2}} \sigma_{x^{i+2}}^2 + \rho_{y^{i+2}} \sigma_{y^{i+2}}^2 + \rho_{x^{i+2}} \rho_{y^{i+2}} (\bar{y}^{i+2} - \bar{x}^{i+2})^2$$

se acotará cada término del lado derecho de manera adecuada.

1. $\sigma_{y^{i+2}} = \sigma_{z^{i+3}} \leq \sigma_{z^{i+2}}$ pues los datos restantes están menos dispersos.

2. $\sigma_{x^{i+2}}^2 \leq \frac{1}{16}\sigma_{z^i}^2$. Primero notar que los elementos en x^{i+2} están entre $\mu_{i+1}+1$ y $\mu_{i+2} = \bar{y}^{i+1}$. Por lo tanto

$$\sigma_{x^{i+2}} \leq \bar{y}^{i+1} - \mu_{i+1},$$

pero usando (3.1) se obtiene que

$$\bar{y}^{i+1} - \mu_{i+1} = \rho_{x^{i+1}}(\bar{y}^{i+1} - \bar{x}^{i+1}),$$

ahora bien, como la iteración i es mala

$$\begin{aligned} \bar{y}^{i+1} - \bar{x}^{i+1} &= \rho_{x^{i+1}}(\bar{y}^{i+1} - \bar{x}^{i+1}) + \rho_{y^{i+1}}(\bar{y}^{i+1} - \bar{x}^{i+1}) \\ &\leq 2\rho_{y^{i+1}}(\bar{y}^{i+1} - \bar{x}^{i+1}) \\ &= 2(\mu_{i+1} - \bar{x}^{i+1}) \\ &\leq 2\rho_{x^i}(\bar{y}^i - \bar{x}^i), \end{aligned}$$

así

$$\begin{aligned} \sigma_{x^{i+2}}^2 &\leq [2\rho_{x^{i+1}}\rho_{x^i}(\bar{y}^i - \bar{x}^i)]^2 \\ &\leq 4\rho_{x^{i+1}}^2\rho_{x^i}^2(\bar{y}^i - \bar{x}^i)^2 \\ &\leq 4\rho_{x^{i+1}}^2\rho_{y^i}\rho_{x^i}(\bar{y}^i - \bar{x}^i)^2. \end{aligned}$$

De la igualdad (3.2) se deduce que $\sigma_{z^i}^2 \geq \rho_{y^i}\rho_{x^i}(\bar{y}^i - \bar{x}^i)^2$ y entonces

$$\begin{aligned} \sigma_{x^{i+2}}^2 &\leq 4\rho_{x^{i+1}}^2\sigma_{z^i}^2 \\ &\leq \frac{1}{16}\sigma_{z^i}^2, \end{aligned}$$

donde se ha usado que la iteración $(i+1)$ también es mala.

3. Como la iteración $(i+1)$ y la iteración i son malas

$$\bar{y}^{i+2} - \bar{x}^{i+2} \leq 2\rho_{x^{i+1}}(\bar{y}^{i+1} - \bar{x}^{i+1}) \leq 4\rho_{x^{i+1}}\rho_{x^i}(\bar{y}^i - \bar{x}^i),$$

entonces

$$\begin{aligned} (\bar{y}^{i+2} - \bar{x}^{i+2})^2 &\leq 16\rho_{x^{i+1}}^2\rho_{x^i}^2(\bar{y}^i - \bar{x}^i)^2 \\ &\leq \frac{1}{4}\sigma_{z^i}^2. \end{aligned}$$

Entonces

$$\sigma_{z^{i+2}}^2 \leq \rho_{x^{i+2}}\frac{1}{16}\sigma_{z^i}^2 + \rho_{y^{i+2}}\sigma_{z^{i+2}}^2 + \rho_{x^{i+2}}\rho_{y^{i+2}}\frac{1}{4}\sigma_{z^i}^2$$

de lo cual se deduce

$$\sigma_{z^{i+2}}^2 \leq \frac{5}{16}\sigma_{z^i}^2.$$

Entonces

$$\varphi_{i+2} = |z^{i+2}|\sigma_{z^{i+2}}^2 \leq \frac{5}{16}|z^i|\sigma_{z^i}^2 = \frac{5}{16}\varphi_i.$$

Por lo tanto para todo i se cumple que $\varphi_{i+2} \leq \frac{7}{8}\varphi_i$ y en consecuencia con $\mathcal{O}(\log n)$ iteraciones de la dinámica se eliminan todos los elementos de la secuencia. \square

Por lo argumentado anteriormente, la dinámica de grafos aplicada a la unión disjunta de cliques se comporta de la misma manera que lo haría la dinámica sobre secuencias aplicada a la secuencia de grados del grafo. Entonces

Corolario 3.1 Para todo grafo G de n vértices que sea unión disjunta de cliques, $t(G) = \mathcal{O}(\log n)$.

Observación Notar que en la demostración del tiempo sobre la dinámica aplicada a secuencias, el valor $+1$ nunca fue considerado en el análisis. En otras palabras, si ahora se considera la dinámica sobre secuencias que en cada iteración elimina los elementos con valor a lo más el promedio, entonces el tiempo de esta dinámica también será $\mathcal{O}(\log n)$.

Capítulo 4

Otros resultados en el modelo BCAST

En este capítulo se estudian otros problemas de grafos en el modelo BCAST. Se comienza estudiando el problema de encontrar un conjunto independiente maximal con un vértice fijo x . Se denotará a dicho problema como MIS_x . Se demuestra que el problema MIS_x resulta difícil en 1 ronda, usando una reducción de detección de cliques. Se construye un protocolo aleatorio que en $\mathcal{O}(\log n)$ rondas, con mensaje de largo $\mathcal{O}(\log n)$ encuentra un el independiente.

A continuación, se estudia el problema de calcular el número cromático de un grafo, dicho problema se denota por CHROM. Utilizando una técnica similar a la utilizada en el capítulo 2, se demuestra que CHROM es difícil en 1 ronda.

Finalmente, se estudia el problema de identificar si el grafo contiene un conjunto dominante de tamaño k , dicho problema se denota por k -DOMSET. Se demuestra que que 1 ronda el problema k -DOMSET es difícil. Este resultado se extiende a los conjuntos ℓ -dominantes, denotando en este caso al problema como (k, ℓ) -DOMSET.

4.1. El problema del MIS_x

4.1.1. MIS_x en modelo determinista

En esta subsección se demostrará la dificultad en el problema determinista. Se mostrará que en caso de poder resolver el problema MIS_x , entonces se pueden reconstruir todos los grafos de n vértices. El lema clave para la demostración de la dificultad es el siguiente

Lema 4.1 Sea \mathcal{P} un protocolo determinista de 1 ronda en BCAST que reconstruye su entrada, es decir, al final de la ronda todos los vértices conocen el grafo de entrada. Entonces $b_{\mathcal{P}} = \Omega(n)$.

Demostración. Suponga que no, sea $c = 1/5$ y n adecuado tal que $b = b_{\mathcal{P}} < n/5$. El número

de bits escritos en la pizarra al final de la ronda son a lo más

$$bn < \frac{n^2}{5},$$

y por lo tanto a lo más $2^{n^2/5}$ grafos distintos se pueden codificar con dicha cantidad de bits. Notando que hay $2^{\binom{n}{2}}$ grafos de n vértices en total, se encuentra la contradicción. \square

Teorema 4.1 Cualquier protocolo determinista \mathcal{P} de 1 ronda en BCAST que resuelva MIS_x cumple $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Sea \mathcal{P} un protocolo determinista de 1 ronda que resuelve MIS_x con mensaje de largo b . La idea es construir un protocolo determinista \mathcal{P}' de 1 ronda que sea capaz de reconstruir su entrada. Dado G grafo de entrada cada vértice simulará estar en el grafo G' que corresponderá al grafo G más un vértice extra llamado simplemente $n + 1$. El siguiente protocolo muestra como se trabajará en dicho grafo

Protocolo 4: Reconstruye el grafo de entrada

- 1 Todos los vértices v escriben 2 mensajes m_v y m'_v usando \mathcal{P} en la pizarra, el mensaje m_v es el mensaje que escribiría si su vecindad es $N_G(v)$ y m'_v es el mensaje que escribiría si su vecindad es $N_G(v) \cup \{n + 1\}$
 - 2 El referee puede saber si la arista uv pertenece al grafo G usando el protocolo \mathcal{P} en los mensajes $\{m_u, m_v\} \cup \{m'_r\}_{r \neq u, v} \cup \{m\}$ donde m es el mensaje de $n + 1$ siendo su vecindad $V \setminus \{u, v\}$
 - 3 Si el independiente encontrado por \mathcal{P} es $\{u, v, n + 1\}$ entonces el referee dice que u y v no están conectados y en caso contrario dice que si lo están
 - 4 Repitiendo esto con todos los pares u, v se logra reconstruir el grafo
-

Para ver la correctitud de \mathcal{P}' basta notar que al tomar G_{uv} grafo con vértices $V \cup \{n + 1\}$ y aristas $E \cup \{s(n + 1)\}_{s \neq u, v}$, tiene como independiente maximal a $\{u, v, n + 1\}$ si y solo si $uv \notin E$. Con esto el referee (o sea todos) pueden reconstruir G ciclando sobre todos los pares de vértices u, v del grafo. Por lo tanto, por lema 4.1 el largo de mensaje $b_{\mathcal{P}'}$ debe ser $\Omega(n)$. Notando que $b_{\mathcal{P}'} \leq b(n) + b(n + 1) \leq 2b(n + 1)$ se concluye que $b(n + 1) = \Omega(n)$ que implica lo que se quería probar. \square

4.1.2. MIS_x en modelo aleatorio

En esta subsección se demostrará que aún añadiéndole aleatoriedad a los protocolos en BCAST, el largo de mensaje para resolver MIS_x debe ser $\Omega(n)$. El espíritu de la demostración es similar a la del caso determinista. En [20] se define el problema H -SUBGRAPH que consiste en reconocer si el grafo de entrada contiene o no al grafo H como subgrafo. Los autores prueban que cuando H es un grafo no bipartito, entonces cualquier protocolo que resuelva H -SUBGRAPH tiene largo de mensaje $b(n) = \Omega(n)$. Se utilizará este resultado para probar la dificultad del problema MIS_x .

Teorema 4.2 Cualquier protocolo aleatorio \mathcal{P} de 1 ronda en BCAST que resuelva MIS_x cumple $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Sea \mathcal{P} un protocolo aleatorio de 1 ronda en BCAST que resuelve MIS_x con mensaje de largo $b = b(n)$. Se construirá un protocolo aleatorio que resuelve K_3 -SUBGRAPH que implicará la cota inferior en b . La idea resulta similar a la demostración del teorema anterior. Se construye un grafo G' a partir del grafo de entrada G . A G se le agrega un vértice s , distinto de los vértices de G , y este nuevo vértice s es unido a 3 vértices u, v, w fijos. Entonces G tiene como triángulo a u, v, w si y solo si $\overline{G'}$ tiene como independiente maximal que contiene a al vértice s al conjunto $\{s, u, v, w\}$.

Protocolo 5: Reconoce si G contiene a K_3

- 1 Todos los vértices v escriben 2 mensajes m_v y m'_v realizados con el protocolo \mathcal{P} . El mensaje m_v corresponde al mensaje con $n + 1$ vértices y vecindad $\overline{N_G(v)}$. El mensaje m'_v corresponde al mensaje con $n + 1$ vértices y vecindad $\overline{N_G(v) \cup \{s\}}$
 - 2 El referee verifica si $\{u, v, w, x\}$ es la respuesta de \mathcal{P} simulando \mathcal{P} en los mensajes $\{m_u, m_v, m_w\} \cup \{m_r : r \neq u, v, w\} \cup \{m\}$ con m mensaje de s con vecindad $V \setminus \{u, v, w\}$
 - 3 Si el independiente maximal es $\{u, v, w, s\}$ todos saben que hay un K_3 en G
 - 4 Si no, itera sobre otro trío u, v, w y en caso de que ninguno dé positivo entonces no hay K_3 en G
-

La correctitud está dada en el párrafo anterior y realizando el mismo cálculo que en el teorema anterior se prueba que $b(n) = \Omega(n)$. □

4.1.3. Un protocolo aleatorizado en $\mathcal{O}(\log n)$ rondas en BCAST con mensajes de largo $\mathcal{O}(\log n)$

De las subsecciones anteriores se sabe que el problema MIS_x es difícil en 1 ronda. La pregunta natural que surge es si en varias rondas lo mismo sigue siendo cierto. Afortunadamente en [3] se construye un algoritmo en paralelo calcular conjuntos maximales en tiempo esperado $\mathcal{O}(\log n)$. El objetivo de esta subsección es mostrar como adaptar ese algoritmo al modelo BCAST para obtener $\mathcal{O}(\log n)$ rondas con mensajes de largo $\mathcal{O}(\log n)$.

El protocolo consiste de un protocolo principal \mathcal{P}_p que ejecuta $\mathcal{O}(\log n)$ veces el protocolo auxiliar \mathcal{P}_s . El protocolo \mathcal{P}_s recibe un grafo de entrada H y entrega como salida un conjunto independiente S . El protocolo \mathcal{P}_p iterativamente ejecuta el protocolo \mathcal{P}_s e incrementa el conjunto independiente hasta llegar a un independiente maximal.

Protocolo 6: $\mathcal{P}_s(H)$

```
1 for  $v \in H$  do
2   if  $d_H(v) = 0$  then
3     └ escribe en la pizarra 1
4   else
5     └ escribe en la pizarra 1 con probabilidad  $1/d_H(v)$  y si no escribe 0
6 for  $uv \in H$  do
7   if  $u$  y  $v$  están marcados con 1 then
8     └ cambiar el 1 de  $u$  con probabilidad  $d_H(v)/(d_H(u) + d_H(v))$  y si no cambiar el 1 de
9       └  $v$ 
9  $S \leftarrow$  vértices que quedan con un 1
```

Es claro que \mathcal{P}_s entrega un conjunto independiente. El protocolo principal es

Protocolo 7: $\mathcal{P}_p(G)$

```
1  $I \leftarrow \{x\}, V(H) \leftarrow V$ 
2 while  $V(H) \neq \emptyset$  do
3   └  $S \leftarrow \mathcal{P}_s(H)$ 
4   └  $V(H) \leftarrow V(H) \setminus (S \cup N_H(S))$ 
5   └  $I \leftarrow I \cup S$ 
```

Primero notar que el protocolo efectivamente se puede realizar en el modelo BCAST. En efecto, en cada iteración de \mathcal{P}_s se sabe quiénes son los vértices que se agregan a conjunto I , más aún, todos los vértices conocen el conjunto I y al conjunto S en todo momento. Por lo tanto, aquellos vértices que están en $N_H(S)$ y que dejan de participar en la siguiente iteración les comunican a todo el mundo que ellos dejan de participar en la siguiente iteración mediante un bit.

Se reproducirá la demostración de [3] que muestra que el tiempo esperado para que $V(H)$ se vacíe es $\mathcal{O}(\log n)$. Para ello se demostrará que en cada iteración de $\mathcal{P}_s(H)$ se borran en esperanza $\Omega(m_H)$ aristas.

Sea $H = (V, E)$, se abreviará d_H como d . Un vértice $v \in V$ se dice *malo* si

$$|\{u \in N(v) : d(u) > d(v)\}| \geq \frac{2}{3}d(v),$$

es decir, a lo menos dos tercios de los vecinos de v tienen grado mayor que el. Un vértice se dirá *bueno* si no es malo. Una arista uv se dirá *mala* si sus extremos son vértices malos y se dirá *bueno* en caso contrario.

Lema 4.2 Sea $u \in V$ bueno de grado $d > 0$. Entonces

$$\mathbb{P}(\exists v \in N(u) : v \text{ escribe 1 durante } \mathcal{P}_s(H)) \geq 1 - e^{-1/3}.$$

Demostración. Sean u_1, \dots, u_k los vecinos de u tales que $d(u_j) \leq d$. Como u es bueno entonces $k \geq d/3$ y así

$$\begin{aligned} \mathbb{P}(\exists u \in N(v) : u \text{ escribe } 1 \text{ durante } \mathcal{P}_s(H)) &\geq \mathbb{P}(\exists u_i \text{ tal que escribe } 1) \\ &\geq 1 - \mathbb{P}(\forall u_i \text{ no escribe } 1) \\ &= 1 - \prod_{i=1}^k (1 - 1/d(u_j)) \\ &\geq 1 - (1 - 1/d)^k \\ &\geq 1 - e^{-1/3} \end{aligned}$$

donde se ha usado que cada u_i escribe en la pizarra 1 de manera independiente del resto. \square

Lema 4.3 Si $u \in V$ tiene grado $d > 0$. Entonces, en la ejecución de $\mathcal{P}_s(H)$,

$$\mathbb{P}(u \in S \mid u \text{ escribió } 1) \geq 1/e.$$

Demostración. Sean u_1, \dots, u_d los vecinos de u con $d_j = d(u_j)$. Sea A_j el evento u_j fue marcado con 1 y el 1 de u se borra al evaluar la aristas uu_j . Se obtiene que $\{A_j\}_{j=1}^d$ son independientes. Entonces

$$\begin{aligned} \mathbb{P}(u \in S \mid u \text{ escribió } 1) &= \mathbb{P}(\text{ningún } A_j \text{ ocurre} \mid u \text{ escribió } 1) \\ &= \prod_{j=1}^d (1 - \mathbb{P}(A_j)) \\ &= \prod_{j=1}^d \left(1 - \frac{1}{d_j} \frac{d_j}{d + d_j}\right) \\ &\geq \left(1 - \frac{1}{d + 1}\right)^d \\ &\geq e^{-1}. \end{aligned}$$

\square

Corolario 4.1 Sea $u \in V$ bueno de grado $d > 0$. Entonces su probabilidad de ser borrado en la ejecución de $\mathcal{P}_s(H)$ es a lo menos $(1 - e^{-1/3})/e$.

Demostración. Sean u_1, \dots, u_d los vecinos de u . Entonces

$$\begin{aligned} \mathbb{P}(u \text{ es borrado}) &\geq \mathbb{P}(\exists u_j \in S) \\ &= \sum_{j=1}^d \mathbb{P}(u_j \text{ es el primero marcado}) \mathbb{P}(u_j \in S \mid u_j \text{ es el primero marcado}) \\ &\geq \sum_{j=1}^d \mathbb{P}(u_j \text{ es el primero marcado}) \mathbb{P}(u_j \in S \mid u_j \text{ fue marcado}) \end{aligned}$$

$$\begin{aligned}
&\geq \frac{1}{e} \sum_{j=1}^d \mathbb{P}(u_j \text{ es el primero marcado}) \\
&= \frac{1}{e} \mathbb{P}(\exists u_j \text{ fue marcado}) \geq \frac{1}{e} (1 - e^{-1/3}). \quad \square
\end{aligned}$$

Lema 4.4 Sea $H = (V, E)$. El número de aristas buenas de H es a lo menos $|E(H)|/2$.

Demostración. Ver Lema 4.4 de [3]. □

Corolario 4.2 El número esperado de aristas borradas en una iteración de \mathcal{P}_s es a lo menos $(1/2e)(1 - e^{-1/3})|E(H)|$. Por lo tanto el número esperado de iteraciones de \mathcal{P}_p es $\mathcal{O}(\log n)$.

Demostración. Sea X el número de aristas borradas en \mathcal{P}_s . Por el lema anterior el número de aristas buenas es a lo menos $\frac{1}{2}|E(H)|$. Por corolario anterior la probabilidad de que una de esas aristas sea borrada es a lo menos $\frac{1}{e}(1 - e^{-1/3})$ y así

$$\begin{aligned}
\mathbb{E}[X] &\geq \mathbb{E}[\#\text{Aristas buenas borradas}] \\
&\geq \frac{1}{2e}(1 - e^{-1/3})|E(H)|.
\end{aligned}$$

Ahora bien, $\mathbb{P}\left[X \geq \frac{1}{100}|E(H)|\right] \geq 1/100$ pues en caso contrario

$$\begin{aligned}
\mathbb{E}[X] &= \mathbb{E}\left[X \mathbf{1}_{\{X \geq \frac{1}{100}|E|\}}\right] + \mathbb{E}\left[X \mathbf{1}_{\{X < \frac{1}{100}|E|\}}\right] \\
&\leq |E(H)| \mathbb{P}\left[X \geq \frac{1}{100}|E|\right] + \frac{1}{100}|E(H)| \mathbb{P}\left[X < \frac{1}{100}|E|\right] \\
&\leq \frac{2}{100}|E(H)|,
\end{aligned}$$

que contradice el resultado anterior. Esto quiere el número esperado de veces que se debe utilizar \mathcal{P}_s para borrar a lo menos $\frac{1}{100}|E|$ aristas es constante y por lo tanto en $\mathcal{O}(\log |E|) = \mathcal{O}(\log n)$ iteraciones de \mathcal{P}_s el grafo se vacía. □

Para concluir el resultado general basta notar que cualquier protocolo con número esperado de rondas $T(n)$ puede transformarse en un protocolo aleatorio con error ε y número de rondas $\frac{1}{\varepsilon}T(n)$ manteniendo el mismo largo de mensaje. En efecto, el nuevo protocolo simulará el protocolo anterior hasta que realice $\frac{1}{\varepsilon}T(n)$ rondas, y si hasta ese momento no se ha concluido se termina de manera anticipada. Si X es el número de rondas del protocolo original, la probabilidad de que el nuevo protocolo tome más de $\frac{1}{\varepsilon}T(n)$ rondas es

$$\mathbb{P}\left[X > \frac{1}{\varepsilon}T(n)\right] \leq \varepsilon,$$

por cota de Markov.

4.2. Colorabilidad

En esta sección se prueba que en modelo BCAST el problema de calcular el número cromático $\chi(G)$ de un grafo es difícil en 1 ronda, se denota por CHROM a dicho problema. Un grafo se dice k -cromático si puede ser coloreado con k colores o de manera equivalente tiene número cromático $\chi(G)$ a lo más k . Sea G grafo de n vértices y sea $F \subseteq G$, durante esta sección se denotará por G_F al grafo $G_F = G \cup (F * K_n)$.

Lema 4.5 G es libre de K_k si y sólo si para todo $F \subseteq G$ con $|F| = k$ el grafo G_F es $(n + k - 1)$ -cromático.

Demostración.

(\implies) Sea $F \subseteq G$ de tamaño k . Sea el siguiente $(n + k - 1)$ -coloreo de G_F :

- K_n es obligatoriamente pintado con n colores: $1, \dots, n$
- F es pintado con $k - 1$ colores: $n + 1, \dots, n + k - 1$
- $G \setminus F$ es pintado con $n - k$ colores: $1, \dots, n - k$

es claro que el coloreo propuesto es válido y por lo tanto G_F es $(n + k - 1)$ -cromático.

(\impliedby) Si $K_k \subseteq G$, sea $F = K_k$ y entonces claramente $\chi(G_F) \geq \chi(K_{k+n}) = n + k$ y por lo tanto G_F no es $(n + k - 1)$ -cromático. \square

El lema anterior entrega una forma de poder identificar la contención de K_k , usando el resultado de [13] para cliques se puede concluir la dificultad en 1 ronda.

Teorema 4.3 Para $k \geq 4$, todo protocolo \mathcal{P} aleatorio que resuelva CHROM en el modelo BCAST en 1 ronda requiere $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Sea \mathcal{P} un protocolo aleatorio resuelve CHROM en 1 ronda y tiene largo de mensaje $b_{\mathcal{P}} = b_{\mathcal{P}}(n)$. El siguiente protocolo resuelve contención de K_k

Protocolo 8: Reconoce si G contiene a K_k

- 1 Todos los vértices v envían dos mensajes: m_v y m'_v usando \mathcal{P} , ambos mensajes tomados en el grafo con $2n$ vértices. El mensaje m_v corresponde al mensaje con vecindad $N_G(v)$ y el mensaje m'_v corresponde al mensaje con vecindad $N_G(v) \cup K_n$
 - 2 El referee prueba sobre todos los $F \subseteq G$ de tamaño k si el grafo G_F es $(n + k - 1)$ -cromático. Para ello utiliza los mensajes $\{m_v\}_{v \in G \setminus F} \cup \{m'_v\}_{v \in F} \cup \{M_1, \dots, M_n\}$ donde los mensajes M_i son los mensajes de los vértices en K_n con vecindad F
 - 3 Si todos los F responden que G_F es $(n + k - 1)$ -cromático, entonces no hay K_k
 - 4 Si para algún F se responde que G_F no es $(n + k - 1)$ -cromático, entonces hay K_k
-

la correctitud proviene del lema 4.5. Para el análisis de la complejidad basta notar que

$$\Omega(n) = b_{\mathcal{P}'}(n) \leq 2b_{\mathcal{P}}(2n),$$

lo que concluye el resultado. □

4.3. Conjuntos dominantes

Un conjunto de vértices D se dice *conjunto dominante* si todos los vértices del grafo están en D o tienen algún vecino en D , en otras palabras todos los vértices están a distancia a lo más 1 del conjunto D . Un vértice $u \in D$ se dice que *domina* a sus vecinos y a si mismo. El problema k -DOMSET en BCAST consiste en saber si el grafo de entrada contiene un conjunto dominante de tamaño k . En esta sección se demostrará que en 1 ronda el problema es difícil.

Sea G un grafo de n vértices y sean $u, v \in G$. Para $k \geq 2$ se define el grafo $G_{u,v}^k$ como el grafo con vértices $V_{u,v}^k = V(G) \cup \{x_1, \dots, x_{k-1}\} \cup \{s, t\}$ y aristas $E_{u,v}^k = E(G) \cup \{ut, x_1s\} \cup \{wx_j : j = 1, \dots, k-1, w \in G, w \neq u, v\}$, ver figura 4.1.

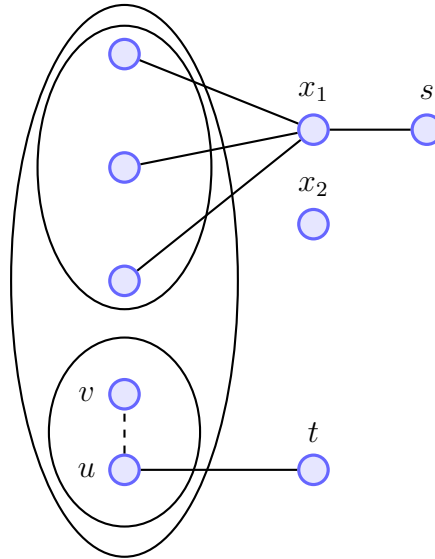


Figura 4.1: Ejemplo de $G_{u,v}^k$ con $k = 3$, el grafo G contiene a la arista uv si y solo si $G_{u,v}^k$ contiene un conjunto dominante de tamaño 3, a saber $\{x_1, x_2, u\}$.

Lema 4.6 Sea G con $n \geq 3$ vértices y sean $u, v \in G$, entonces G contiene a la arista uv si y sólo si $G_{u,v}^k$ tiene un conjunto dominante de tamaño k .

Demostración. Se realizará la demostración para $k = 2$ pues el caso general resulta análogo.

(\implies) Si $uv \in G$, entonces $D = \{x_1, u\}$ es un conjunto dominante de $G_{u,v}^2$, en efecto, x_1 domina a todos los vértices salvo u, v y t y el vértice u domina a v y t .

(\Leftarrow) Sea D un conjunto dominante de $G_{u,v}^2$ de tamaño mínimo. Como los vértices s y t deben estar dominados por vértices en D , necesariamente $D \cap \{x_1, s\} \neq \emptyset$ y $D \cap \{u, t\} \neq \emptyset$. Suponga que $uv \notin G$, entonces los vértices x_1, s, u y t no dominan al vértice v y por lo tanto D debe tener un vértice v' que domine a v lo cual implica que D tiene al menos tamaño 3. \square

El lema anterior entrega una estrategia para identificar aristas en G , esto permitirá al referee reconstruir el grafo lo cual por lema 4.1 se sabe que es difícil.

Teorema 4.4 Para $k \geq 2$, todo protocolo \mathcal{P} determinista que resuelva k -DOMSET en el modelo BCAST en 1 ronda requiere $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Nuevamente se realizará la demostración para $k = 2$, el caso general es análogo. Sea \mathcal{P} un protocolo determinista de 1 ronda en BCAST que resuelve k -DOMSET, el siguiente protocolo resuelve reconstrucción de grafos:

Protocolo 9: Reconstruye el grafo de entrada

- 1 Todos los vértices v escriben 3 mensajes en la pizarra m_v, m'_v, m''_v usando \mathcal{P} . Los 3 mensajes son considerados sobre el grafo con $n + 3$ vértices, el mensaje m_v es el mensaje que escribiría si su vecindad es $N_G(v)$, m'_v es el mensaje que escribiría si su vecindad es $N_G(v) \cup \{x_1\}$ y m''_v es el mensaje que escribiría si su vecindad es $N(v) \cup \{t\}$
 - 2 Para $u, v \in G$ el referee usa el protocolo \mathcal{P} para identificar si $G_{u,v}^2$ contiene un conjunto dominante de tamaño 2 o no. Para ello utiliza los mensajes $\{m'_w\}_{w \neq u,v} \cup \{m_v, m''_u\} \cup \{M_t, M_{x_1}, M_s\}$ donde los mensajes M_t es el mensaje de t con vecindad u , el mensaje M_{x_1} es el mensaje de x_1 con vecindad $G \cup \{s\} \setminus \{u, v\}$ y M_s es el mensaje de s con vecindad x_1
 - 3 Por el lema 4.6 el referee sabe que $uv \in G$ si y sólo si $G_{u,v}^2$ tiene un conjunto dominante de tamaño 2. Iterando sobre todos los $u, v \in G$ el referee reconstruye el grafo G
-

por lema 4.1 se sabe que $b_{\mathcal{P}}(n) = \Omega(n)$ y como $b_{\mathcal{P}'}(n) = \Theta(b_{\mathcal{P}}(n + 3))$ se concluye. \square

Observación Notar que el resultado sigue siendo válido para todo $k \leq n$.

El resultado anterior se puede extender al modelo con aleatoriedad, para ello se utilizará que reconocer cliques es difícil. Sea G grafo con $n \geq 5$ vértices y $k \geq 2$, para $F = (u_1, u_2, u_3, u_4)$ con $u_i \in G$ se define G_F^k como el grafo con vértices $V_F^k = V(G) \cup \{x_j\}_{j=1}^{k-1} \cup \{s, t\}$ y aristas $E_F^k = E(G) \cup \{u_1 t, x_1 s\} \cup \{w x_1 : w \in G, w \neq u_i\}$, ver figura 4.2.

Lema 4.7 Sea G con $n \geq 5$ vértices, G contiene a K_4 como subgrafo si y sólo si existe $\{u_1, u_2, u_3, u_4\} \subseteq G$ tal que todo ordenamiento F de $\{u_1, u_2, u_3, u_4\}$ hace que G_F^k tenga un conjunto dominante de tamaño k .

Demostración. La demostración se realizará para $k = 2$, el caso general resulta análogo.

(\Rightarrow) Sea $\{u_1, u_2, u_3, u_4\} \subseteq G$ tal que $G[u_1, u_2, u_3, u_4] \cong K_4$. Sea F un ordenamiento

cualquiera de $\{u_1, u_2, u_3, u_4\}$, entonces $D = \{x_1, u_1\}$ es conjunto dominante de G_F^2 , en efecto, u_1 domina a K_4 y t y x_1 domina a todo el resto del grafo.

(\Leftarrow) Suponga que $K_4 \not\subseteq G$, por lo tanto existe F un ordenamiento de 4 vértices u_1, u_2, u_3, u_4 tal que u_1 no está conectado a u_2 . Sea D el conjunto dominante mínimo de G_F^2 , nuevamente como D debe dominar a los vértices x_1, s_1, u_1 y t , se debe cumplir que $D \cap \{x_1, s_1\} \neq \emptyset$ y $D \cap \{u_1, t\} \neq \emptyset$. Como x_1, s_1, u_1 y t no dominan a u_2 se debe cumplir que existe w distinto a los 4 vértices anteriores que domina a u_2 y por lo tanto $|D| \geq 3$. \square

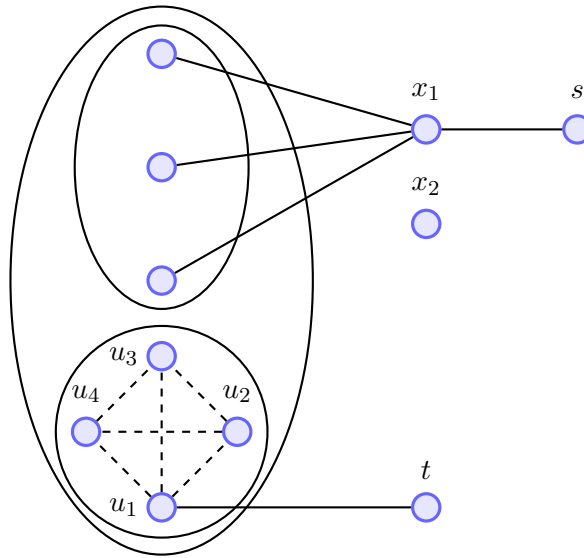


Figura 4.2: Ejemplo de G_F^k con $k = 3$, el grafo G contiene al completo K_4 si y solo si G_F^k contiene un conjunto dominante de tamaño 3 para cualquier ordenamiento de los vértices de K_4 .

El lema 4.7 entrega la estrategia para identificar cliques de tamaño 4 en grafos con al menos 5 vértices.

Teorema 4.5 Para $k \geq 4$, todo protocolo \mathcal{P} aleatorio que resuelva k -DOMSET en el modelo BCAST en 1 ronda requiere $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Por simplicidad se realizará la demostración para $k = 2$, el caso general es análogo. Sea \mathcal{P} un protocolo aleatorio de 1 ronda en BCAST que resuelve k -DOMSET, el siguiente protocolo resuelve detección de cliques de tamaño 4:

Como $b_{\mathcal{P}}(n) = \Omega(n)$ y $b_{\mathcal{P}'}(n) = \Theta(b_{\mathcal{P}}(n+3))$ se concluye. \square

Protocolo 10: Reconoce K_4

- 1 Todos los vértices v escriben 3 mensajes en la pizarra m_v, m'_v, m''_v usando \mathcal{P} . Los 3 mensajes son considerados sobre el grafo con $n + 3$ vértices, el mensaje m_v es el mensaje que escribiría si su vecindad es $N_G(v)$, m'_v es el mensaje que escribiría si su vecindad es $N_G(v) \cup \{x_1\}$ y m''_v es el mensaje que escribiría si su vecindad es $N(v) \cup \{t\}$
 - 2 Para $u_1, u_2, u_3, u_4 \in G$ y un ordenamiento F de estos vértices el referee utiliza el protocolo \mathcal{P} para identificar si G_F^2 contiene un conjunto dominante de tamaño 2 o no. Para ello utiliza los mensajes $\{m'_w\}_{w \neq u_1, u_2, u_3, u_4} \cup \{m''_{u_1}, m_{u_2}, m_{u_3}, m_{u_4}\} \cup \{M_t, M_{x_1}, M_s\}$ donde el mensaje M_t es el mensaje de t con vecindad u_1 , el mensaje M_{x_1} es el mensaje de x_1 con vecindad $G \cup \{s\} \setminus \{u_1, u_2, u_3, u_4\}$ y M_s es el mensaje de s con vecindad x_1
 - 3 Por el lema 4.7 el referee sabe que $G[u_1, u_2, u_3, u_4] \cong K_4$ si y sólo si G_F^2 tiene un conjunto dominante de tamaño 2, para todo ordenamiento F de u_1, u_2, u_3, u_4 . Iterando sobre todos los $u_1, u_2, u_3, u_4 \in G$ el referee identifica si G contiene a K_4 o no
-

4.3.1. Conjuntos ℓ -dominantes

Un conjunto de vértices D se dice ℓ -dominante si todo vértice está a distancia a lo más ℓ de D , el caso $\ell = 1$ consiste en el conjunto dominante clásico. El problema (k, ℓ) -DOMSET en BCAST consiste en identificar si el grafo de entrada tiene un conjunto de vértices ℓ -dominante. En esta subsección se demostrará que el problema (k, ℓ) -DOMSET es difícil en 1 ronda.

Sea G grafo de n vértices. para $k \geq 2$ sea $G_F^{k, \ell}$ el grafo obtenido a partir del grafo G_F^k expandiendo las aristas x_1s y wx_1 , $w \neq u_i$, en caminos de largo $\ell - 1$ y la arista u_1t por un camino de largo $2\ell - 1$, se denota por t' al vértice vecino de u_1 en este último camino (ver figura 4.3).

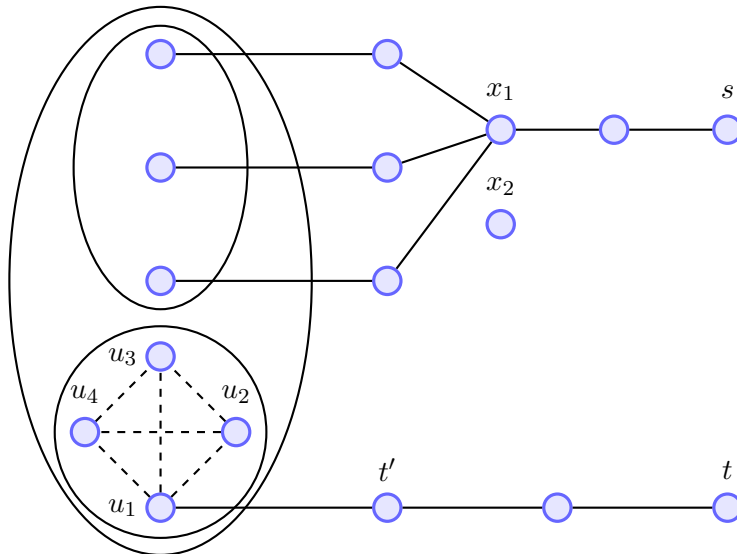


Figura 4.3: Ejemplo de $G_F^{k, \ell}$ con $k = 3$, el grafo G contiene K_4 si y solo si para todo F el grafo $G_F^{k, \ell}$ contiene un conjunto 2-dominante de tamaño 3, a saber $\{x_1, x_2, t'\}$.

Lema 4.8 Sea G con $n \geq 5$ vértices, G contiene a K_4 como subgrafo si y sólo si existe $\{u_1, u_2, u_3, u_4\} \subseteq G$ tal que todo ordenamiento F de $\{u_1, u_2, u_3, u_4\}$ hace que $G_F^{k,\ell}$ tenga un conjunto ℓ -dominante de tamaño k .

Demostración. La demostración es análoga al lema 4.7. □

Teorema 4.6 Para $k \geq 4$ y $\ell \geq 1$ fijo, todo protocolo \mathcal{P} aleatorio que resuelva (k, ℓ) -DOMSET en el modelo BCAST en 1 ronda requiere $b_{\mathcal{P}}(n) = \Omega(n)$.

Demostración. Por simplicidad se realizará el bosquejo para $k = 2$, el caso general es análogo. Sea \mathcal{P} un protocolo aleatorio de 1 ronda en BCAST que resuelve (k, ℓ) -DOMSET, el nuevo protocolo actúa de manera similar al protocolo del teorema 4.5, cada vértice v envía 5 mensajes $m_v, m'_v, m''_v, m'''_v, m''''_v$ cada uno correspondiente al mensaje que enviaría v si su vecindad fuese $N_G(v)$, o $N_G(v) \cup \{t'\}$ o $N_G(v) \cup \{x'\}$ siendo x' el primer vértice de algún camino que conecta x_1 con v . El referee puede realizar todos los calculos necesarios sobre todos los $F \subseteq G$ de tamaño 4 y concluir si G contiene o no un K_4 . Finalmente el largo de mensaje cumple una cota similar a la del teorema 4.5 y con eso se concluye lo enunciado. □

Observación En el resultado anterior se usa implícitamente que ℓ es constante. No se puede asegurar que el resultado siga siendo cierto cuando $\ell = \ell(n)$, pues el largo de los caminos incrementa el tamaño del grafo de la reducción en al menos $\Omega(\ell n)$ vértices.

Conclusión

En el presente trabajo se estudiaron tres temas que involucran teoría de grafos y computación distribuida. En primer lugar, se estudió el problema de calcular la degenerancia de grafos en el *Congested Clique*: UCAST y BCAST. A continuación, se estudió una dinámica sobre grafos que iterativamente elimina los vértices del grafo que cumplen con tener grado a lo más el grado promedio $+1$. Finalmente, se estudiaron diversos problemas de grafos en el modelo BCAST.

La primera parte de esta tesis correspondió al estudio del cálculo de la degenerancia en los modelos UCAST y BCAST. Comenzando con UCAST, se construyó un protocolo aleatorio que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia del grafo de entrada en $\mathcal{O}(\log n)$ rondas, con mensaje de largo $\mathcal{O}(\log n)$. En el modelo BCAST, se demostró que el problema de calcular la degenerancia, DEG, resulta difícil en 1 ronda. Específicamente, se demostró que cualquier protocolo que sea capaz de resolver DEG en 1 ronda, debe enviar un mensaje de largo $\Omega(n)$. En el mismo modelo, se diseñó un protocolo determinista que calcula una $(2 + \varepsilon)$ -aproximación de la degenerancia del grafo de entrada en $\mathcal{O}(\log n)$ rondas, con mensaje de largo $\mathcal{O}(\log n)$. Finalmente, se construyó un protocolo aleatorio que calcula una $(1 + \varepsilon)$ -aproximación de la degenerancia del grafo de entrada en $\mathcal{O}(\log n)$ rondas, con mensaje de largo $\mathcal{O}(\log n)$, cuando el grafo de entrada es α -denso.

La segunda parte de esta tesis correspondió al estudio de la dinámica sobre grafos. Esta dinámica, iterativamente elimina del grafo aquellos vértices que tienen grado a lo más el *grado promedio del grafo hasta el momento más uno*. Se definió el *tiempo de la dinámica* como el número mínimo de iteraciones para vaciar el grafo. Se conjeturó que dicho número es $\mathcal{O}(\log n)$ para todo grafo de n vértices. Se demostró por qué es necesario el término « $+1$ » al borrar los vértices durante cada iteración, dando como contraejemplo el grafo camino. La conjetura resultó cierta para los grafos acíclico o bosques, los grafos planares, los grafos con degenerancia acotada y para los grafos que son unión disjunta de cliques. En las primeras tres familias, se usó fuertemente que el grado promedio es acotado localmente. Se intentó probar que en cada iteración de la dinámica una fracción constante del grafo era eliminada, sin embargo, un contraejemplo sencillo demostró que pueden haber iteraciones en las cuales un número constante (independiente de n) del grafo es borrado. Para estudiar la dinámica en la unión disjunta de cliques, se desarrolló una nueva dinámica sobre secuencias de números fijos. La nueva dinámica sobre secuencia de números, iterativamente borra aquellos elementos que tienen valor a los el promedio más uno. Se demostró que para toda secuencia de n números, el tiempo de esta nueva dinámica es $\mathcal{O}(\log n)$ mediante un potencial que involucró el número de elementos y la varianza de la secuencia. Este resultado terminó por demostrar la veracidad

de la conjetura en la unión disjunta de cliques.

La parte final de la tesis, correspondió al estudio de diversos problemas de grafos en el modelo de computación distribuida **BCAST**. Primero, se estudió el problema de encontrar un conjunto independiente maximal con un vértice fijo. A este problema se le denotó MIS_x , donde x es el vértice fijo. Se demostró que el problema es difícil en 1 ronda. Se construyó un protocolo aleatorio que puede resolver MIS_x en $\mathcal{O}(\log n)$ rondas, con mensaje de largo $\mathcal{O}(\log n)$. Segundo, se estudió el problema de calcular el número cromático del grafo. A este problema se le denotó por **CHROM**. Se demostró, usando una técnica similar a la usada en **DEG**, que en 1 ronda el problema es difícil. Finalmente, se estudió el problema de conjuntos dominantes y su generalización a los conjuntos ℓ -dominantes. A dichos problemas se les denotaron k -**DOMSET** y (k, ℓ) -**DOMSET**, respectivamente. En ambos casos se demostró que en 1 ronda los problemas son difíciles.

Esta memoria logra resolver parcialmente ciertas inquietantes sobre el problema de la degenerancia **DEG**. Por un lado, prueba que en el modelo **BCAST** degenerancia no puede ser resuelto en 1 ronda sin mensajes largos. Pero por otro lado, no logra entregar pruebas explícitas de su posible dificultad en varias rondas. Como trabajo a futuro, y suponiendo que **DEG** es realmente difícil en varias rondas, se podría intentar buscar una construcción similar a la usada en [13] para resolver **Disjointness** usando **DEG**. Adicionalmente, en el modelo **BCAST** resta demostrar o refutar la dificultad de los problemas **CHROM**, k -**DOMSET** y (k, ℓ) -**DOMSET**.

Una gran pregunta que deja abierta este trabajo concierne a la dinámica propuesta, ¿Será cierta la conjetura o existe algún grafo para el cuál el tiempo de la dinámica sea $\omega(\log n)$? Una conjetura más débil es que el tiempo de la dinámica es $o(n)$, por ejemplo $\mathcal{O}(\log^k n)$ para algún $k > 1$. Las técnicas usadas no lograron dar luces en grafos generales. El principal problema reside, y como se explicó en el capítulo, en que el grado promedio no tiene ningún tipo de monotonía. Más aún, no resulta evidente cómo relacionar varias iteraciones consecutivas, pues el grado promedio no es monótono como si lo es en el caso de la dinámica sobre secuencias. Un trabajo a futuro en esta área podría ser buscar un refinamiento de la técnica usada para la demostración de la dinámica sobre secuencias para poder aplicarla en grafos.

Bibliografía

- [1] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proc. of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2012, pages 459–467, 2012.
- [2] Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: Sparsification, spanners, and subgraphs. In *Proc. of the 31st Symposium on Principles of Database Systems*, PODS 2012, pages 5–14, 2012.
- [3] Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *J. Algorithms*, 7(4):567–583, 1986.
- [4] Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2004.
- [5] Florent Becker, Martín Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, and Ioan Todinca. Adding a referee to an interconnection network: What can(not) be computed in one round. In *Proc. of the 25th IEEE International Parallel and Distributed Processing Symposium*, IPDPS 2011, pages 508–514, 2011.
- [6] Florent Becker, Martín Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, and Ioan Todinca. Adding a referee to an interconnection network: What can(not) be computed in one round. *CoRR*, abs/1009.4447, 2010.
- [7] Florent Becker, Ivan Rapaport, Pedro Montealegre, and Ioan Todinca. The simultaneous number-in-hand communication model for networks: private coins, public coins and determinism. *Proc. of SIROCCO 2014*, pages 83–95, 2014.
- [8] Allan Bickle. The k-cores of a graph. 2010.
- [9] Béla Bollobás. *Extremal graph theory, (volume 11 of London Mathematical Society Monographs)*. Academic Press, London, 1978.
- [10] Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. Algebraic methods in the congested clique. In *Proc. of the 2015 ACM Symposium on Principles of Distributed Computing*, PODC 2015, pages 143–152, 2015.
- [11] Reinhard Diestel. *Graph Theory*. Graduate Texts in Mathematics, Volume 173. Springer-Verlag, Heidelberg, 4th edition, July 2010.

- [12] Danny Dolev, Christoph Lenzen, and Shir Peled. “Tri, tri again”: Finding triangles and small subgraphs in a distributed setting. In *Proc. of the 26th International Symposium on Distributed Computing (DISC 2012)*, volume 7611 of *LNCS*, pages 195–209, 2012.
- [13] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *PODC*, pages 367–376. ACM, 2014.
- [14] Martin Farach-Colton and Meng-Tsung Tsai. Computing the degeneracy of large graphs. In Alberto Pardo and Alfredo Viola, editors, *LATIN*, volume 8392 of *Lecture Notes in Computer Science*, pages 250–260. Springer, 2014.
- [15] Martín Farach-Colton and Meng-Tsung Tsai. Tight approximations of degeneracy in large graphs. In *Latin American Symposium on Theoretical Informatics*, pages 429–440. Springer, 2016.
- [16] Eugene C Freuder. A sufficient condition for backtrack-free search. *Journal of the ACM (JACM)*, 29(1):24–32, 1982.
- [17] András Gyárfás and András Frank. How to orient the edges of a graph. In *Combinatorics, I, Colloq. Math. Soc. János Bolyai*, volume 18, pages 353–364, 1978.
- [18] James W Hegeman, Gopal Pandurangan, Sriram V Pemmaraju, Vivek B Sardeshmukh, and Michele Squizzato. Toward optimal bounds in the congested clique: Graph connectivity and mst. *Proc. of PODC 2015*, pages 91–100, 2015.
- [19] James W. Hegeman, Sriram V. Pemmaraju, and Vivek Sardeshmukh. Near-constant-time distributed algorithms on a congested clique. *Proc. of the 28th International Symposium on Distributed Computing, DISC 2014*, pages 514–530, 2014.
- [20] Jarkko Kari, Martín Matamala, Ivan Rapaport, and Ville Salo. Solving the induced subgraph problem in the randomized multiparty simultaneous messages model. In Christian Scheideler, editor, *SIROCCO*, volume 9439 of *Lecture Notes in Computer Science*, pages 370–384. Springer, 2015.
- [21] Lefteris M Kirousis and Dimitris M Thilikos. The linkage of a graph. *SIAM Journal on Computing*, 25(3):626–647, 1996.
- [22] Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.
- [23] Christoph Lenzen. Optimal deterministic routing and sorting on the congested clique. In *Proceedings of the 2013 ACM symposium on Principles of distributed computing*, pages 42–50. ACM, 2013.
- [24] Don R Lick and Arthur T White. k -degenerate graphs. *Canadian J. of Mathematics*, 22:1082–1096, 1970.
- [25] Zvi Lotker, Elan Pavlov, Boaz Patt-Shamir, and David Peleg. MST construction in $\mathcal{O}(\log \log n)$ communication rounds. In *Proc. of the 15th ACM Symposium on Parallel*

Algorithms and Architectures, SPAA 2003, pages 94–100, 2003.

- [26] David W. Matula and Leland L. Beck. Smallest-last ordering and clustering and graph coloring algorithms. *J. ACM*, 30(3):417–427, July 1983.
- [27] Shanmugavelayutham Muthukrishnan. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.
- [28] Boaz Patt-Shamir and Marat Teplitsky. The round complexity of distributed sorting. In *Proc. of the 2011 ACM Symposium on Principles of Distributed Computing*, PODC 2011, pages 249–256, 2011.
- [29] David Peleg. *Distributed Computing: A Locality-sensitive Approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.