Felix Yowtang Liu

Um Algoritmo de Teste de Propriedades Aplicado ao Problema da Clique

Trabalho apresentado como requisito parcial à conclusão do Curso de Bacharelado em Ciência da Computação, setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Renato José da Silva Carmo.

Curitiba PR
2017

Sumário

Notação Usada

Sejam $a, b, n \in \mathbb{Z}$ inteiros tais que $a \leq b$ e $n \geq 1$. Seja também x um vetor.

$[a..b]$	denota o conjunto dos inteiros entre a e b inclusive, isto é, $\{z \in \mathbb{Z} : a \leq z \leq b\}$.
$[n]$	denota o conjunto $[1..n]$.
x_n	denota o conteúdo da n-ésima posição de x.
$\lvert x \rvert$	denota o tamanho (i.e. quantidade de posições) de x.

A notação usada para grafos é apresentada na Seção 3.1.

Capítulo 1

Introdução

À medida que a capacidade de processamento e armazenamento aumentam, a noção de eficiência de um algoritmo tende a se tornar mais restrita. Antes do computador digital, computações em tempo finito eram consideradas soluções satisfatórias. A expansão dos recursos computacionais disponíveis permitiu que muitos problemas emergissem e os primeiros problemas realmente difíceis se tornassem evidentes, levando à definição da computação em tempo polinomial. E quando mesmo o tempo polinomial já não bastava, esforços foram feitos em torno da computação paralela (Papadimitriou, 1994).

Avanços tecnológicos recentes exigem o processamento de quantidades cada vez maiores de dados em tempos cada vez menores. Há situações práticas em que a entrada é grande demais para que uma solução em tempo linear seja satisfatória, como é o caso de alguns bancos de dados modernos. Em outras situações, o acesso à entrada é custoso – por exemplo, via rede, em algum servidor externo. Ou ainda, é possível que não exista sequer uma representação explícita da entrada, mas um oráculo capaz de calcular seus valores pontuais esteja disponível. No entanto, a leitura completa da entrada ainda é passo imprescindível para vários algoritmos (Fischer, 2004).

Teste de propriedades consiste no estudo de algoritmos construídos para inspecionar uma pequena parte da entrada e a partir disso inferir se a entrada toda apresenta certa propriedade global ou se está *longe* de apresentá-la. Em geral, assume-se que o algoritmo está dotado de acesso aleatório à entrada: que sabe-se o tamanho da entrada e há um oráculo capaz de informar o valor de uma dada posição da entrada. Espera-se, com isso, obter algoritmos que computem suas respostas em tempo *sublinear* ou, ainda, *constante independente do tamanho da entrada*.

Dentre os problemas ainda sem resposta satisfatória – mesmo polinomial – está o problema da clique: um problema de decisão NP-completo que figura entre os 21 problemas de Karp (Karp, 1972). Sua versão de otimização – encontrar uma clique máxima – é NP-difícil e figura entre os problemas combinatórios mais estudados, com aplicações práticas em várias outras áreas do conhecimento.

Em Goldreich, Goldwasser e Ron (1998), os autores propõem uma solução em tempo independente do tamanho do grafo para o problema da ρ-CLIQUE , que pode ser usada para resolver o problema de decisão. No entanto, muitos aspectos da prova apresentada são abreviados ou omitidos e algumas constantes são apresentadas sob notação assintótica.

O presente trabalho se propõe a apresentar o teste de propriedades e estudar o algoritmo proposto em Goldreich et al. (1998) para a clique, apresentando os passos sugeridos no trabalho original de forma mais detalhada e explicitando os valores das constantes.

No Capítulo 2, apresentamos o teste de propriedades com um breve histórico das pesquisas sobre o assunto (Seção 2.1), apontamos suas características principais (Seção 2.2) e explicitamos algumas situações em que a aplicação do teste de propriedades pode ser interessante (Seção 2.3). Concluímos o capítulo estudando um *toy problem* como ilustração (Seção 2.4).

No Capítulo 3, estabelecemos brevemente o formalismo sobre grafos necessário para o presente trabalho (Seção 3.1) e apresentamos o *modelo denso* para grafos (Seção 3.2), que é usado para o estudo de teste de propriedades em grafos e será usado no presente trabalho. Enfim, apresentamos brevemente o problema da clique e definimos o problema da ρ-CLIQUE (Seção 3.3) e discorremos acerca do testador apresentado em Goldreich et al. (1998) (Seção 3.4 e Seção 3.5), descrevendo-o e demonstrando a sua corretude.

Capítulo 2

Teste de Propriedades

Teste de propriedades é o estudo da seguinte classe de problemas:

> Seja P uma propriedade definida sobre um domínio D e seja $d \in D$ um objeto dado. Dada a possibilidade de acesso a informações locais sobre d, a tarefa consiste em, pela da inspeção de uma pequena fração de d, determinar (com alta probabilidade) se d satisfaz a propriedade P ou se d está *longe* de satisfazer P.

Por exemplo, se o objeto é uma função f, uma propriedade que pode ser testada é sua linearidade. Nesse caso, o testador pode realizar consultas como "Qual o valor de $f(x)$?" e, através de uma amostra no domínio de f, inferir se a função é linear ou está longe de ser linear.

Ou ainda, quando o objeto é um grafo G, uma propriedade a se testar é se G é bipartido. Neste caso, as consultas poderiam ser a respeito da vizinhança de um dado vértice ou da presença de arestas entre pares de vértices.

Neste capítulo discorremos mais a respeito do teste de propriedades. Apresentamos um breve histórico das pesquisas sobre o assunto na Seção 2.1, seguido de algumas características de destaque na Seção 2.2. Enfim, apontamos alguns cenários em que o uso do teste de propriedades pode ser benéfico na Seção 2.3 e terminamos na Seção 2.4 com um exemplo de problema que pode ser resolvido com teste de propriedades.

2.1 Um breve histórico

Teste de propriedades foi originado no contexto de verificação de corretude de programas. O conceito aparece (implicitamente) pela primeira vez em Blum, Luby e Rubinfeld (1990), no qual os autores demonstram que é possível testar se a função computada por um dado programa é linear após examinar os valores que ela assume para alguns pontos amostrados no seu domínio.

Rubinfeld e Sudan (1996) é um estudo feito em continuidade ao trabalho iniciado em Blum et al. (1990). Nesse trabalho, o teste de propriedades é definido explicitamente pela primeira vez, com a intenção de estudar outras propriedades algébricas de funções que também poderiam ser testadas.

O estudo de teste de propriedades num contexto mais geral foi iniciado em Goldreich et al. (1998), quando os autores apresentaram o teste de propriedades como um modelo geral aplicável a diversos outros tipos de objetos e então voltaram a sua atenção à aplicação do teste de propriedades em grafos, que até então ainda não haviam sido considerados.

Posteriormente, em Alon e Shapira (2008) foi demonstrado que todas as propriedades hereditárias de grafos são testáveis em tempo constante (uma propriedade é hereditária quando é satisfeita por um grafo e todos os seus subgrafos); e em Alon, Fischer, Newman e Shapira (2009) foi fornecida uma caracterização completa das propriedades de grafos testáveis com número constante de consultas, a qual envolve conceitos sobre regularidade em grafos.

2.2 Características principais

Um algoritmo capaz de resolver um problema de teste de propriedades é chamado de *testador de propriedades*, e no presente trabalho eles também serão tratados simplesmente por "testadores".

Um testador tem acesso a informações parciais do objeto dado. Tal acesso pode lhe ser concedido através de um oráculo que lhe permita realizar consultas locais a respeito do objeto ou ainda pela apresentação de amostras do objeto seguindo uma dada distribuição.

A restrição de acesso à entrada imposta pela definição do teste de propriedades tem como consequência a emergência de algumas das características de destaque dos testadores, as quais são abordadas nesta seção.

2.2.1 Algoritmos aleatorizados

Uma característica necessária imediatamente aparente é a de que os testadores precisam ser algoritmos aleatorizados, dado que a entrada não pode ser lida completamente. Exceto em casos degenerados (em que grande parte da entrada não importa para o problema em questão), a intuição é a de que cada componente da entrada deve ser passível de amostragem para que o algoritmo possa inferir uma característica do todo. (Fischer, 2004)

Dessa forma, os testadores podem apresentar uma pequena probabilidade de erro. Em geral, testadores apresentam *erro bilateral*: há probabilidade positiva de avaliar incorretamente se a entrada possui a propriedade e também se ela está longe de tê-la. No entanto, existem testadores com *erro unilateral*, os quais sempre avaliam corretamente quando a entrada apresenta a propriedade, mas podem errar quando ela está longe de ter (e.g. o testador de k-colorabilidade apresentado em Goldreich et al. (1998)).

Outra classificação decorrente desta característica é a que considera a forma como consultas influenciam na execução do algoritmo. Um testador é considerado *não adaptativo* quando todas as consultas são feitas uma única vez ao começo do algoritmo e o oráculo não é mais consultado durante o resto da execução. Nesse caso, o resultado depende unicamente da amostra obtida através dessas consultas. Por outro lado, um testador *adaptativo* pode consultar o oráculo durante a sua execução, o que lhe confere liberdade para tomar decisões considerando o resultado de consultas feitas até então.

2.2.2 Decisão aproximada

Há casos, no entanto, em que a presença de aleatoriedade no algoritmo não basta para que ele apresente resultados precisos. Considere, por exemplo, o problema de determinar se todas as posições de um vetor dado são zero. Não há algoritmo que leia pequena parte da entrada capaz de distinguir com alta probabilidade entre uma entrada que satisfaz a propriedade e uma que contém um único '1' em alguma posição.

Fischer (2004) aponta para o fato de que há conjecturas e resultados afirmando que a maior parte das propriedades é "evasiva", exigindo um número linear de consultas para decidi-las mesmo com o uso de algoritmos probabilísticos.

O problema de determinar se uma dada entrada satisfaz ou não uma certa propriedade é um problema de decisão, que será chamado no presente trabalho de *decisão exata*.

Definição 2.1. (Decisão Exata) Seja P uma propriedade definida sobre um domínio D. Podemos definir o problema de decisão exata associado a P da seguinte maneira:

Decisão Exata	
Instância :	um objeto $x \in D$
Resposta :	sim, se x satisfaz P; ou
	não, se x não satisfaz P.

Os objetos para os quais a resposta é sim são também chamados de *instâncias positivas* e objetos para os quais a resposta é não dá-se o nome de *instâncias negativas*. Dizemos que um algoritmo *aceita* uma instância quando ele devolve sim para ela e dizemos que ele *rejeita* a instância quando a resposta dada é não.

Um problema de decisão exata também pode ser definido em termos do conjunto C_P dos objetos que satisfazem P: todo $x \in C_P$ deve ser aceito e todo $x \notin C_P$ deve ser rejeitado.

Dessa forma, é possível definir uma noção de distância em relação à propriedade:

Definição 2.2. (ϵ-longe e ϵ-perto) Dado um parâmetro de distância $\epsilon \in [0, 1]$ e uma métrica $d : D \times D \to [0, 1]$, um objeto $x \in D$ é dito *ϵ-longe* de P se $\min\{d(x, y), y \in C_P\} > \epsilon$; e, analogamente, *$\epsilon$-perto* de P se $\min\{d(x, y), y \in C_P\} \leq \epsilon$.

É necessário destacar que a métrica varia conforme a classe de objetos, a sua representação e a propriedade que está sendo considerada. Por exemplo, para duas *strings*, uma métrica possível é a razão entre a distância de Hamming (quantidade de posições em que as *strings* diferem) e o tamanho da maior *string*. Goldreich (2011) apresenta três modelos de representação distintos para grafos, os quais possuem uma métrica distinta cada.

Espera-se que a facilidade de determinar se uma instância não satisfaz uma propriedade seja crescente em relação à sua distância à propriedade: entradas com maior diferença apresentam maior probabilidade de oferecer evidências nesse quesito.

Considerando essas observações, pode-se propor o seguinte relaxamento do problema de decisão exata: ao invés de demandar a rejeição de toda instância que não satisfaz a propriedade, somente exigir que sejam rejeitadas as entradas que estejam *longe* de satisfazê-la. Assim, possíveis instâncias difíceis são ignoradas, o que pode reduzir a complexidade necessária para a resolução do problema.

Dito isso, pode-se formular uma versão aproximada do problema de decisão exata, que será chamada no presente trabalho de *decisão aproximada*.

Definição 2.3. (Decisão Aproximada) Sejam P uma propriedade definida sobre um domínio D, $\epsilon \in [0, 1]$ um parâmetro de distância e $d : D \times D \to [0, 1]$ uma métrica. O problema de decisão aproximada associado a P é o seguinte:

Decisão Aproximada	
Instância :	um objeto $x \in D$
Resposta :	sim, se x satisfaz P, ou
	não, se x está ϵ-longe de satisfazer P.

Definição 2.4. (ϵ-testador) Seja P uma propriedade. Um ϵ-testador para P é um algoritmo (possivelmente aleatorizado) A que resolve o problema de decisão aproximada associado a P com parâmetro de distância ϵ.

O comportamento esperado de um ϵ-testador para P quando a entrada é ϵ-próxima de satisfazer P (mas não satisfaz P) é arbitrário. Por isso, a decisão aproximada deve ser considerada um problema de promessa (*promise problem*), uma vez que seu comportamento não é definido para todo o domínio e, assim, deve presumir que todo objeto dado provém de alguma região para a qual há comportamento definido.

2.2.3 Troca de precisão por eficiência

Segue das características mencionadas acima que o teste de propriedades troca precisão por eficiência. O parâmetro de distância ϵ costuma estar atrelado ao tamanho da amostragem realizada por um testador, o que está relacionado com a quantidade de consultas realizadas (indicativo do tempo de execução necessário) e com a precisão do algoritmo (devido à incerteza introduzida com a redução da amostra). Assim, espera-se que ambos o tempo de execução e a precisão sejam decrescentes em função de ϵ.

2.2.4 Foco em complexidade de consultas

Como os testadores relacionam-se com a entrada através de consultas, pesquisas na área de teste de propriedades tendem a voltar o seu foco à *complexidade de consultas* dos testadores. De certa forma, esse foco fornece uma estimativa inicial da complexidade de tempo de um dado testador; e a obtenção do limitante inferior para a complexidade de consultas implica um limitante inferior para a complexidade de tempo. Em geral, espera-se que a complexidade de tempo seja no máximo exponencial em função da complexidade de consultas; e em muitos casos ela é polinomial em relação à complexidade de consultas.

2.3 Benefícios potenciais

Nesta seção, apresentamos algumas situações nas quais a troca de precisão por eficiência feita pelos algoritmos de teste de propriedades pode ser benéfica. Naturalmente, como os benefícios dependem das especificidades das aplicações, limitamo-nos à descrição de alguns cenários gerais de interesse.

Acesso completo ao objeto é intratável Seja devido ao tamanho do objeto ou pelo custo de acesso, existem situações em que o acesso completo ao objeto torna-se intratável. Nesse caso, não há alternativa a não ser fazer uso de algoritmos com número de consultas sublinear.

Objetos ou apresentam a propriedade, ou estão longe de tê-la Aplicações nas quais sabe-se *a priori* que todos os objetos ou são perfeitos (possuem a propriedade), ou são "muito ruins" (longe), não precisam tratar objetos "quase bons" (perto).

Objetos perto de ter a propriedade são satisfatórios Devido à natureza dos testadores, um objeto aceito pode possuir a propriedade ou estar perto de satisfazê-la, uma vez que o comportamento para objetos ϵ-perto é arbitrário. É possível que para a aplicação objetos que quase satisfaçam a propriedade sejam tão bons quanto os que satisfaçam; ou que o custo para modificá-los de forma a satisfazer a propriedade seja razoável.

Como passo preliminar para outro algoritmo Nos testadores, objetos longe de apresentar a propriedade são rejeitados e objetos perto de apresentar a propriedade *podem* ser rejeitados. Quando a quantidade de objetos longe de apresentar a propriedade não é pouca, essa característica pode ser usada para eliminar rapidamente muitos objetos que não têm dada propriedade.

2.4 Exemplo: O problema da maioria

Para ilustrar a aplicação do teste de propriedades a um problema, vamos considerar o problema da maioria:

Seja $\mathsf{MAJ} := \{x : \sum_{i=1}^{|x|} x_i > |x|/2\}$ o conjunto dos vetores binários em que há mais bits 1 que bits 0. Enquanto o problema de decisão exata para MAJ corresponde ao problema de decidir se um vetor x pertence ou não a MAJ, no contexto dos algoritmos de decisão aproximada o problema da maioria consiste em decidir se um vetor x de n bits está em MAJ ou se ele é ϵ-longe de MAJ. Nesse caso, a métrica é a de Hamming: a distância entre dois vetores é a fração de posições em que eles diferem em relação ao tamanho do maior deles.

Nesta seção, serão demostradas as seguintes afirmações:

- Existe um ϵ-testador que resolve o problema de decisão aproximada relacionado a MAJ com tempo $\Theta(\frac{1}{\epsilon^2})$ (Página 8);
- Não há algoritmo em tempo sublinear que resolva a sua versão de decisão exata (Página 11); e
- A aleatorização é uma característica essencial do testador (Página 14).

2.4.1 Um ϵ-testador para MAJ

Seja x uma entrada com $n = |x|$ posições.

Considere o seguinte testador: o algoritmo consulta a entrada x em $m = c/\epsilon^2$, $c \geq 3$, posições sorteadas independentemente com probabilidade uniforme, denotadas $i_1, \ldots, i_m$, e aceita a entrada se e somente se a média dos bits sorteados (i.e. $\sum_{j \in [m]} x_{i_j}/m$) for maior que $(1-\epsilon)/2$.

Mostraremos que este testador aceita com alta probabilidade toda entrada em MAJ e rejeita com alta probabilidade toda entrada ϵ-longe de MAJ.

2.4.2 Corretude do ϵ-testador

Teorema 2.5. *Seja A o ϵ-testador descrito na Subseção 2.4.1 e x uma entrada para A. Então*

$$\begin{aligned} x \in \mathsf{MAJ} \quad &\Longrightarrow \quad \mathbb{P}(A(x) = \textsf{sim}) \geq 2/3; \text{ e} \\ x \text{ é } \epsilon\text{-longe de } \mathsf{MAJ} \quad &\Longrightarrow \quad \mathbb{P}(A(x) = \textsf{não}) \geq 2/3. \end{aligned}$$

Prova Sejam $X_i, i \in [m]$, as variáveis aleatórias definidas da seguinte forma:

$$X_i = \begin{cases} 0, \text{ se } x_i = 0; \text{ ou} \\ 1, \text{ se } x_i = 1. \end{cases}$$

E seja $S = m^{-1} \sum_{i=1}^{m} X_i$.

Seja também $M = n^{-1} \sum_{i=1}^{n} x_i$ a média total da entrada.

Esta prova será feita em duas partes. Vamos primeiro mostrar que a probabilidade da média da amostra se distanciar no máximo $\epsilon/2$ da média total da entrada é maior que 2/3, ou seja,

$$\mathbb{P}\left(|S - M| \leq \frac{\epsilon}{2}\right) \geq \frac{2}{3}.$$

Esse fato será usado ao fim da prova para mostrar que com probabilidade 2/3:

1. Quando $x \in \mathsf{MAJ}$, a média da amostra é maior que $\frac{1-\epsilon}{2}$ e o algoritmo aceita a entrada; e
2. Quando x é ϵ-longe de MAJ, a média da amostra é menor que $\frac{1-\epsilon}{2}$ e o algoritmo rejeita a entrada.

Note que a probabilidade p de um bit do vetor x sorteado com probabilidade uniforme ter valor 1 é

$$p = \frac{\sum_{i=1}^{n} x_i}{n} = M$$

Assim, a variância de S é dada por:

$$\begin{aligned}
\text{Var}\,(S) &= \text{Var}\left(\frac{\sum_{i=1}^{m} X_i}{m}\right) \\
&= \frac{\text{Var}\,(\sum_{i=1}^{m} X_i)}{m^2} && \text{(propriedade da variância)} \\
&= \frac{\sum_{i=1}^{m} \text{Var}\,(X_i)}{m^2} && \text{(variância da soma de VA's independentes)} \\
&= \frac{\sum_{i=1}^{m} p(1-p)}{m^2} && \text{(variância de VA indicadora)} \\
&= \frac{mp(1-p)}{m^2} \\
&= \frac{p(1-p)}{m}.
\end{aligned}$$

E a esperança de S é dada por:

$$\begin{aligned}
\mathbb{E}\,[S] &= \frac{\sum_{i=1}^{m} \mathbb{E}\,[X_i]}{m} \\
&= \frac{\sum_{i=1}^{m} p}{m} \\
&= \frac{mp}{m} \\
&= p = M.
\end{aligned}$$

Pela desigualdade de Chebyshev (Apêndice A), para toda constante positiva $a \in \mathbb{R}$,

$$\mathbb{P}\,(|S - \mathbb{E}\,[S]\,| \geq a) \leq \frac{\text{Var}\,(S)}{a^2}.$$

Assim,

$$\mathbb{P}\left(|S - M| \geq a\right) \leq \frac{p(1-p)}{ma^2}.$$

Tomando $a = \epsilon/2$, temos

$$\mathbb{P}\left(|S - M| \geq \frac{\epsilon}{2}\right) \leq \frac{4p(1-p)}{m\epsilon^2}.$$

Para se obter $\frac{4p(1-p)}{m\epsilon^2} \leq 1/3$, basta observar que

$$p(1-p) \leq \frac{1}{4}; \text{ e}$$
$$\frac{4p(1-p)}{3} \leq \frac{1}{3}.$$

Do que concluímos que tomando $m \geq 3/\epsilon^2$ obtemos

$$\mathbb{P}\left(|S - M| \geq \frac{\epsilon}{2}\right) \leq \frac{1}{3}; \text{ e}$$
$$\mathbb{P}\left(|S - M| \leq \frac{\epsilon}{2}\right) \geq \frac{2}{3},$$

provando que a probabilidade da média da amostra se distanciar no máximo $\epsilon/2$ da média total da entrada é pelo menos 2/3, o que conclui a primeira parte da prova.

Vamos mostrar agora que, com probabilidade pelo menos 2/3:

1. Quando $x \in \mathsf{MAJ}$, o algoritmo aceita a entrada; e

2. Quando x é ϵ–longe de MAJ, o algoritmo rejeita a entrada.

Mostramos que, com probabilidade pelo menos 2/3, vale que

$$|S - M| \leq \frac{\epsilon}{2}.$$

Ou seja, que

$$-\frac{\epsilon}{2} \leq S - M \leq \frac{\epsilon}{2}.$$

Em outras palavras,

$$M - \frac{\epsilon}{2} \leq S \leq \frac{\epsilon}{2} + M.$$

Quando $x \in \mathsf{MAJ}$, temos que $M > 1/2$. Nesse caso,

$$\frac{1}{2} < M; \text{ e}$$
$$\frac{1-\epsilon}{2} < M - \frac{\epsilon}{2}.$$

Como temos que com probabilidade pelo menos 2/3

$$M - \frac{\epsilon}{2} \leq S,$$

então com probabilidade pelo menos 2/3,

$$\frac{1-\epsilon}{2} < S;$$

e concluímos que quando $x \in \mathsf{MAJ}$, o algoritmo aceita a entrada com probabilidade 2/3.

Quando x é ϵ-longe de MAJ, mais que ϵn dos bits de x precisam ser alterados para que $x \in \mathsf{MAJ}$. Como os bits que precisam ser alterados têm valor 0 em x, temos que $M < 1/2 - \epsilon$. Nesse caso,

$$M < \frac{1}{2} - \epsilon; \text{ e}$$
$$\frac{\epsilon}{2} + M < \frac{\epsilon}{2} + \frac{1}{2} - \epsilon = \frac{1-\epsilon}{2}.$$

Como temos com probabilidade 2/3 que

$$S \leq \frac{\epsilon}{2} + M,$$

então com probabilidade 2/3,

$$S < \frac{1-\epsilon}{2};$$

do que concluímos que quando x é ϵ-longe de MAJ, o algoritmo rejeita a entrada com probabilidade ao menos 2/3. □

2.4.3 Limite mínimo para a decisão exata de MAJ

Teorema 2.6. *Qualquer algoritmo aleatorizado que resolva o problema de decisão exata para* MAJ *deve fazer* $\Omega(n)$ *consultas.*

Prova Esta será uma prova por contradição: vamos mostrar que se um algoritmo aleatorizado A que resolve o problema de decisão exata para MAJ fizer um número sublinear de consultas, então existe uma instância para a qual ele erra com alta probabilidade, o que não poderia acontecer.

Para cada $n \in \mathbb{N}$, consideremos dois vetores binários X_n, Z_n tais que:

- X_n é sorteado uniformemente entre os vetores de tamanho n com $\left\lfloor \frac{n}{2} \right\rfloor + 1$ bits 1; e
- Z_n é sorteado uniformemente entre os vetores de tamanho n com $\left\lfloor \frac{n}{2} \right\rfloor$ bits 1.

Disso temos que $\mathbb{P}(X_n \in \mathsf{MAJ}) = 1$ e $\mathbb{P}(Z_n \in \mathsf{MAJ}) = 0$.

Vamos primeiro provar que se A consultar a entrada em q posições, então $|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| \leq \frac{q}{n}$. Usaremos isso para mostrar que se for realizado um número sublinear de consultas, então A terá alta probabilidade de errar a resposta ou para X_n, ou para Z_n.

É instrutivo visualizar a construção de X_n e de Z_n através do seguinte processo:

1. Primeiro, um $i \in [n]$ é sorteado uniformemente.
2. Depois, um vetor y de n posições é sorteado uniformemente entre os vetores com exatamente $\left\lfloor \frac{n}{2} \right\rfloor$ bits 0 cuja posição i vale 0.
3. Então $X_n \leftarrow y$ e $Z_n \leftarrow y$.
4. Ao fim, $X_n[i] \leftarrow 1$.

Note que se a posição i não for consultada, o comportamento de A é o mesmo para X_n e Z_n. Assim, A só pode apresentar resultados diferentes quando ao menos uma das q consultas é à posição i.

Seja T então o conjunto de todas as sequências de q consultas feitas sobre uma entrada de tamanho n. Vamos destacar os seguintes subconjuntos:

- $U \subset T$, as sequências sem i que fazem A aceitar a entrada;
- $D_{X_n} \subseteq T$, as sequências com i que fazem A aceitar X_n; e
- $D_{Z_n} \subseteq T$, as sequências com i que fazem A aceitar Z_n.

Assim, podemos dizer que

$$\begin{aligned}
|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| &= \left| \frac{|U| + |D_{X_n}|}{|T|} - \frac{|U| + |D_{Z_n}|}{|T|} \right| \\
&= \left| \frac{|U| + |D_{X_n}| - |U| - |D_{Z_n}|}{|T|} \right| \\
&= \left| \frac{|D_{X_n}| - |D_{Z_n}|}{|T|} \right|.
\end{aligned}$$

Ou seja, que a diferença entre a probabilidade de A aceitar X_n e a de A aceitar Z_n só depende das sequências em que i está presente.

Seja então $D \subseteq T$ o conjunto das sequências com i. Naturalmente, $0 \leq |D_{X_n}| \leq |D|$ e $0 \leq |D_{Z_n}| \leq |D|$. Assim, $\max\{|D_{X_n}| - |D_{Z_n}|\} = |D|$.

Logo,

$$\begin{aligned}
|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| &= \left| \frac{|D_{X_n}| - |D_{Z_n}|}{|T|} \right| \\
&\leq \left| \frac{|D|}{|T|} \right|.
\end{aligned}$$

O número de sequências possíveis é $|T| = n^q$ e o número de sequências sem i é dado por $|T \setminus D| = (n-1)^q$. Assim, temos $|D| = |T| - |T \setminus D| = n^q - (n-1)^q$.

$$\begin{aligned}\left|\frac{|D|}{|T|}\right| &= \left|\frac{n^q - (n-1)^q}{n^q}\right| \\ &= \left|\frac{n^q}{n^q} - \frac{(n-1)^q}{n^q}\right| \\ &= 1 - \left(\frac{n-1}{n}\right)^q;\end{aligned}$$

E então

$$|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| \leq 1 - \left(\frac{n-1}{n}\right)^q.$$

Para provar que $|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| \leq \frac{q}{n}$, basta observar que $1 - \left(\frac{n-1}{n}\right)^q \leq \frac{q}{n}$.

Para todo $n \geq 1$, com $q = 1$ temos

$$1 - \left(\frac{n-1}{n}\right)^q = \frac{1}{n} = \frac{q}{n};$$

E para todo $k > 1$, se $1 - \left(\frac{n-1}{n}\right)^k \leq \frac{k}{n}$, então $1 - \left(\frac{n-1}{n}\right)^{k+1} \leq \frac{k+1}{n}$, porque

$$\begin{aligned}1 - \left(\frac{n-1}{n}\right)^{k+1} &= 1 - \left(\frac{n-1}{n}\right)^k + \left(\frac{n-1}{n}\right)^k - \left(\frac{n-1}{n}\right)^{k+1} \\ &= 1 - \left(\frac{n-1}{n}\right)^k + \left(\frac{n-1}{n}\right)^k \left(1 - \frac{n-1}{n}\right) \\ &\leq \frac{k}{n} + \left(\frac{n-1}{n}\right)^k \frac{1}{n} \leq \frac{k+1}{n}.\end{aligned}$$

Provando que se A consultar a entrada em q posições, então $|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| \leq \frac{q}{n}$.

Vamos mostrar agora que se o número de consultas realizadas por A for $o(n)$, então A terá alta probabilidade de errar a decisão ou para X_n, ou para Z_n.

Se q for $o(n)$, então como $|\mathbb{P}(A(X_n) = 1) - \mathbb{P}(A(Z_n) = 1)| \leq \frac{q}{n}$, a probabilidade de A devolver a mesma resposta para X_n e Z_n aumenta à medida que n aumenta. Isso significa que, caso ela acerte com alta probabilidade a resposta para um dos vetores, então ela também errará com alta probabilidade a resposta para o outro.

Logo, se A realiza um número sublinear de consultas, então A apresenta com alta probabilidade a resposta errada para alguma instância. Assim, A deve realizar $\Omega(n)$ consultas para responder corretamente todas as instâncias. $\square$

2.4.4 Necessidade de aleatorização

Teorema 2.7. *Qualquer algoritmo determinístico que distinga entre entradas em* MAJ *e entradas 0,5-longe de* MAJ *precisa realizar ao menos* $\frac{n}{2}$ *consultas, onde* n *é o tamanho da entrada.*

Prova Mostraremos que se A é um algoritmo determinístico que faz $q < \frac{n}{2}$ consultas e aceita todas as instâncias positivas, então ele também aceita o vetor z_0 que possui apenas zeros, o qual é 0,5-longe de MAJ.

Considere a execução de A em que todas as posições consultadas possuem valor 0, e chamemos o conjunto das posições consultadas de Q. Naturalmente, o vetor z_0 é consistente com essas respostas.

Por outro lado, considere o vetor x tal que em toda posição $i \in Q$ o seu valor é 0 e em todas as outras posições o seu valor é 1. x também é consistente com uma execução dessas. No entanto, note que como A realiza $q < \frac{n}{2}$ consultas, então x contém mais que $\frac{n}{2}$ bits 1. Ou seja, $x \in$ MAJ.

A age de maneira similar tanto para x, quanto para z_0. Como A aceita x, então A também aceita z_0.

Logo, qualquer algoritmo determinístico que faça menos que $\frac{n}{2}$ consultas falha em distinguir instâncias em MAJ e instâncias que estão 0,5-longe de MAJ. $\square$

Capítulo 3

Um testador para CLIQUE

Em Goldreich et al. (1998), os autores introduzem o problema da ρ-CLIQUE e apresentam um testador para a versão de decisão aproximada desse problema. Para isso, apresentam primeiro um algoritmo chamado *Encontrador de Quase-Clique* que, dado um grafo com uma clique de tamanho c, devolve um conjunto de c vértices perto de ser uma clique. Em seguida, é apresentado o *Testador de Grau da Clique*, que é uma modificação do encontrador.

Começaremos este capítulo estabelecendo o formalismo de que necessitamos para o estudo desse algoritmo (Seção 3.1) e apresentando o *modelo denso* para grafos, proposto em Goldreich (2011) para o estudo de teste de propriedades em grafos (Seção 3.2). A seguir, apresentamos brevemente o problema da clique e definimos o problema da ρ-CLIQUE (Seção 3.3). Enfim, discorremos acerca do testador apresentado em Goldreich et al. (1998) para a ρ-CLIQUE (Seção 3.4 e Seção 3.5).

3.1 Grafos

Dados um conjunto S e um inteiro k, o conjunto de subconjuntos de S de tamanho k é denotado $\binom{S}{k}$. É importante observar que $\left|\binom{S}{k}\right| = \binom{|S|}{k}$.

Um *grafo* G é um par $(V(G), E(G))$, onde $V(G)$ é um conjunto finito e $E(G) \subseteq \binom{V(G)}{2}$. Cada elemento de $V(G)$ é chamado um *vértice* de G e cada elemento de $E(G)$ é chamado de uma *aresta* de G. Quando $E(G) = \binom{V(G)}{2}$, dizemos que o grafo é *completo*.

Um vértice v é *vizinho* de u em G se $\{u, v\}$ é uma aresta de G. A *vizinhança* do vértice v em G é o conjunto de seus vizinhos em G e é denotado $\Gamma_G(v)$. A vizinhança do vértice v num conjunto de vértices S será denotado $\Gamma_S(v)$ e corresponde ao conjunto dos vizinhos de v em S (i.e. $\Gamma_S(v) := \Gamma_G(v) \cap S$).

O *grau* do vértice v em G é o número de vizinhos de v em G e será denotado $\deg(v)$. O grau de um vértice v em relação a um conjunto de vértices S será denotado $\deg_S(v)$ e corresponde à quantidade de vizinhos que v possui em S (i.e. $\deg_S(v) := |\Gamma_S(v)|$).

Um grafo H é dito *subgrafo* de G se $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Dado um conjunto de vértices $S \subseteq V(G)$, o *subgrafo induzido por S em G* é o grafo $G[S] := \left(S, E(G) \cap \binom{S}{2}\right)$.

Uma *clique* em G é um subconjunto de $V(G)$ que induz um subgrafo completo. Uma clique é *maximal* se não está contida em alguma outra clique e é *máxima* se o seu tamanho é o maior dentre os tamanhos das cliques em G. O tamanho de uma clique máxima de G é também chamado de *número de clique* do grafo e é denotado $\omega(G)$.

Dado um parâmetro de distância $\epsilon \in [0, 1]$, uma *ϵ-quase-clique* é um conjunto de vértices $\widetilde{C} \subseteq V(G)$ tal que $\widetilde{C}$ é ϵ-próxima de uma clique de tamanho $|\widetilde{C}|$. Uma ρ-clique é uma clique de tamanho $\rho|V(G)|$. Enfim, denotaremos por C_N^ρ os grafos de N vértices que consistem numa ρ-clique acompanhada de $(1-\rho)N$ vértices isolados.

Dois grafos G e H são isomorfos se existe uma bijeção $f : V(G) \to V(H)$ tal que para todo $u, v \in V(G)$ vale que

$$\{u, v\} \in E(G) \iff \{f(u), f(v)\} \in E(H).$$

3.2 Modelo Denso para Grafos

No estudo de teste de propriedades, a maneira com que objetos são representados determina que tipo de consultas podem ser realizadas e a medida de distância. Para o estudo de testadores, Goldreich propõe três modelos para a representação de grafos (Goldreich, 2011). Dentre esses modelos está o *modelo denso* para grafos (ou *modelo de matriz de adjacência*), usado para o estudo do testador que será apresentado adiante.

No modelo denso, um grafo G é representado por uma função booleana $g : V(G) \times V(G) \to \{0, 1\}$ tal que $g(u, v) = 1$ se e somente se u e v são vizinhos em G (i.e. $\{u, v\} \in E(G)$).

Sejam G, H dois grafos e g, h as respectivas funções booleanas que as representam no modelo denso. Caso $|V(G)| = |V(H)|$, definimos a distância $d(G, H)$ entre os grafos G e H como sendo a fração mínima de g e h que deve ser alterada para que G e H sejam isomorfos.

Se $|V(G)| > |V(H)|$, então consideramos o grafo H' que corresponde ao grafo H aumentado em $|V(G)| - |V(H)|$ vértices isolados e definimos $d(G, H) = d(G, H')$. Naturalmente, é sempre verdade que $d(G, H) = d(H, G)$.

3.3 O problema da clique

O problema de decisão exata da CLIQUE é definido da seguinte forma:

Clique (CLIQUE)	
Instância :	um grafo G e um inteiro k.
Pergunta :	G tem clique de tamanho k?

Sua versão de decisão é um dos problemas apresentados em Karp (1972) e trata-se de um problema $\mathcal{NP}$-completo (Garey e Johnson, 1979), do que conclui-se que sua versão de otimização é $\mathcal{NP}$-difícil. O problema da clique figura entre os problemas combinatórios mais estudados e possui muitas aplicações práticas em outras áreas do conhecimento. Wu e Hao (2015) é uma resenha sobre ele e apresenta vários apontamentos nesse quesito.

Dado $\rho \in (0, 1]$, o problema de decisão exata da ρ-Clique é definido da seguinte maneira:

ρ-Clique (ρ-CLIQUE)

Instância : Um grafo G

Pergunta : A maior clique de G tem tamanho pelo menos $\rho|V(G)|$?

O problema da ρ-CLIQUE pode ser definido em termos do conjunto C_ρ dos grafos G cuja maior clique tem tamanho pelo menos $\rho|V(G)|$ (i.e. $C_\rho := \{G : \omega(G) \geq \rho|V(G)|\}$). Note que ρ-CLIQUE e CLIQUE estão relacionados no sentido que quando $\rho = \frac{k}{V(G)}$ elas são problemas equivalentes.

3.4 Encontrador de Quase-Clique

Algoritmo 1: Encontrador de Quase-Clique (ENCONTRAρ)

Entrada : Um grafo G de N vértices com clique de tamanho ρN

Saída : Uma ϵ-quase-clique $\widetilde{C}$ de tamanho ρN

$t \leftarrow \left\lceil c_t \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2} \right\rceil$

$r \leftarrow \left\lceil c_r \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2} \right\rceil$

$U \leftarrow t$ vértices de $V(G)$ selecionados uniformemente

$W \leftarrow r$ vértices de $V(G)$ selecionados uniformemente

para cada *$U' \subseteq U$ de tamanho $\lceil t\frac{\rho}{2} \rceil$* **faça**

  $T(U') \leftarrow \{v \in V(G) : \Gamma_G(v) \cap U' = U'\}$

  $W(U') \leftarrow \{w \in W : \Gamma_G(w) \cap U' = U'\}$

  $C(U') \leftarrow \rho N$ vértices de $T(U')$ com maior $|\Gamma_G(v) \cap W(U')|$

$\widetilde{C} \leftarrow C(U')$ computado que minimiza $d(C(U'), C_N^\rho)$

devolva $\widetilde{C}$

Na construção do testador para ρ-CLIQUE em Goldreich et al. (1998), os autores apresentam o Encontrador de Quase-Clique (descrito no Algoritmo 1): um algoritmo que, dado um grafo G de N vértices com uma clique C de tamanho ρN, encontra com probabilidade pelo menos $1 - \delta$ um conjunto de vértices $\widetilde{C} \subseteq V(G)$ tal que $|\widetilde{C}| = \rho N$ e $\widetilde{C}$ é uma ϵ-quase-clique, onde $\delta \in [0, \frac{1}{2})$ é a probabilidade de erro do algoritmo.

Durante a prova da corretude do encontrador que apresentaremos em Subseção 3.4.2, assumiremos que $\epsilon < \rho^2$, uma vez que a distância máxima que um grafo arbitrário pode estar de ter uma clique de tamanho ρN é ρ^2.

Nesta seção, explicaremos a intuição por trás deste algoritmo (Subseção 3.4.1) e forneceremos os apontamentos suficientes para a prova de sua corretude (Subseção 3.4.2), tomando como referência o que foi apresentado em (Goldreich et al., 1998, p. 694 - 699) e explicitando etapas da prova não detalhadas no trabalho original. Enfim, discutimos que valores seriam adequados para as constantes c_t e c_r do algoritmo (Subseção 3.4.3).

3.4.1 Intuição do algoritmo

Antes de apresentar a intuição do algoritmo, vamos definir o seguinte conceito, que será usado diversas vezes durante a discussão do encontrador e do testador.

Definição 3.1. (Vizinho de todo X) Seja G um grafo e $X \subseteq V(G)$ um subconjunto de seus vértices. Dizemos que um vértice $v \in V(G)$ é *vizinho de todo X* se e somente se ele for vizinho de todos os vértices em X. Ou seja,

$$v \text{ é vizinho de todo } X \iff \Gamma_X(v) = X \setminus \{v\}.$$

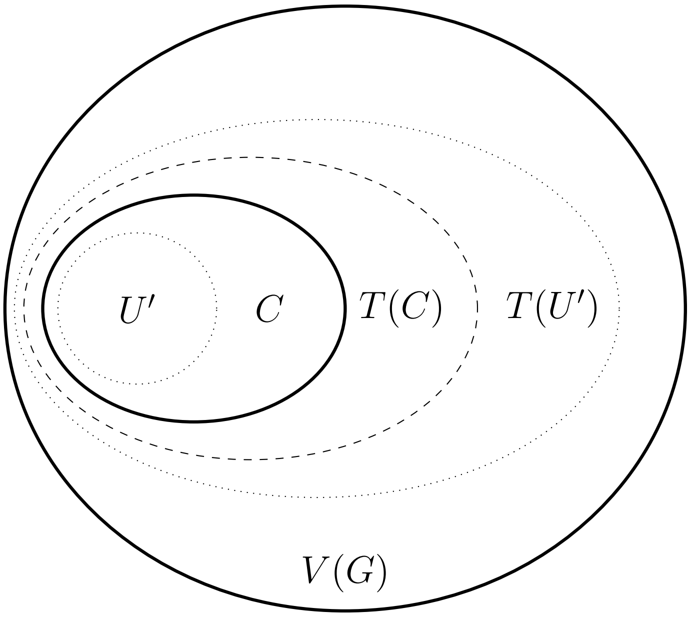

Figura 3.1: Relaxamento considerado pelo algoritmo.

Note que se $v \in X$, basta que ele seja vizinho de todos os outros vértices em X. Denotaremos por $T(X)$, o conjunto de todos os vértices de $V(G)$ que são vizinhos de todo X, isto é,

$$T(X) := \{v \in V(G) : \Gamma_G(v) \cap X = X \setminus \{v\}\}.$$

Para entender o algoritmo ENCONTRAρ , considere o seguinte teorema:

Teorema 3.2. *Seja G um grafo de N vértices que possui uma clique C. Então o conjunto com os $|C|$ vértices de maior grau em $G[T(C)]$ forma uma clique.*

Prova Cada $c \in C$ estará em $T(C)$ porque c é vizinho de todo vértice em $C \setminus \{c\}$. Além disso, como cada $v \in T(C)$ é vizinho de todos os vértices de C, então c será vizinho de todos os outros $|T(C)| - 1$ vértices de $T(C)$; e terá o maior grau que um vértice em $T(C)$ pode ter.

Como é possível que existam outros vértices $t \in T(C)$ que não pertencem à clique, podemos afirmar que há pelo menos $|C|$ vértices em $T(C)$ de grau $|T(C)| - 1$ (podem existir mais). Assim, ao ordenar os vértices de $T(C)$ pelo seu grau em $G[T(C)]$, temos garantidamente que os primeiros $|C|$ vértices são vizinhos de todo o $T(C)$ e, portanto, são vizinhos entre si – ou seja, uma clique. $\square$

O algoritmo faz uso do seguinte relaxamento: considere uma amostra $U' \subset C$. Todo vértice vizinho de todo C é também vizinho de todo U'. No entanto, podem existir vértices vizinhos de todo U' que não sejam de todo C (Figura 3.1). Caso U' seja grande o suficiente, há alta probabilidade de que quase todo vértice em $T(U')$ seja vizinho de quase todo C. Nesse caso, os $|C|$ vértices de maior grau em $G[T(U')]$ estarão próximos de ser uma clique.

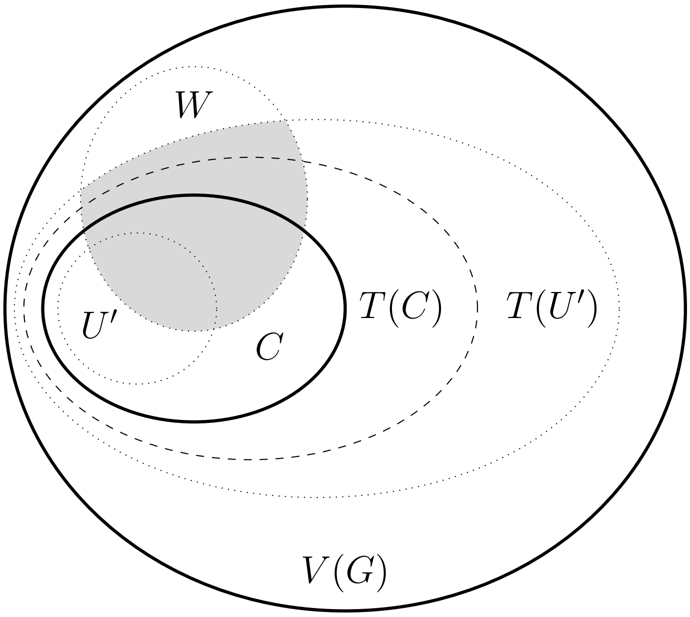

Figura 3.2: Cenário considerado pelo algoritmo com estimativa de grau. A região sombreada corresponde a $W(U')$.

Essa amostra U' é encontrada selecionando uniformemente um conjunto de vértices $U \subset V(G)$ de tamanho suficiente e considerando todos os seus subconjuntos U' de tamanho $|U|\frac{\rho}{2}$. Mostraremos posteriormente que há alta probabilidade de $|U \cap C| \geq |U|\frac{\rho}{2}$ e, portanto, um desses U' se enquadrará nas condições do parágrafo anterior.

Uma última alteração feita é a de estimar o grau de cada $v \in T(U')$, ao invés de computá-lo exatamente. Isso, apesar de não ser necessário pra eficiência do algoritmo, simplifica a análise do testador que será apresentado posteriormente.

A estimativa é feita ao selecionar uniformemente outro conjunto $W \subset V(G)$. Para cada U', fazemos $W(U')$ ser o conjunto dos vértices de W vizinhos de todo U' (isso pode ser visto como amostrar $T(U')$), como ilustrado pela Figura 3.2. Então ordenamos os vértices de $T(U')$ de acordo com o seu grau em $W(U')$, ao invés do seu grau em $G[T(U')]$.

3.4.2 Prova de corretude

Dada a intuição do algoritmo, a prova de sua corretude consiste em atestar a veracidade das seguintes afirmações:

- Um conjunto $U \subseteq V(G)$ de tamanho suficiente sorteado uniformemente tem alta probabilidade de conter uma boa amostra $U' \subseteq U$ da clique C. (Página 21)

- Quando $|T(U')| \geq |C|$, um conjunto $W \subseteq V(G)$ de tamanho suficiente sorteado uniformemente tem alta probabilidade de oferecer boas estimativas de grau para vértices em $T(U')$. (Página 26)

- Se U' é uma boa amostra de C e W oferece boas estimativas de grau para $T(U')$, então os $|C|$ vértices de maior grau segundo W em $T(U')$ formam uma quase-clique. (Página 28)

Uma boa amostra da clique

Antes de mais nada, é necessário definir mais precisamente o que faz de um conjunto uma *boa amostra* da clique, que chamaremos de conjunto ϵ_1-clique-representativo.

Definição 3.3. (Conjunto ϵ_1-clique-representativo) Dado um grafo G de N vértices com clique C de tamanho ρN e $\epsilon_1 \leq \rho$, um conjunto $U' \subseteq V(G)$ é ϵ_1-*clique-representativo* (em relação a C), se para a maior parte dos vértices v vizinhos de todo U', o vértice v é também vizinho de pelo menos $(\rho - \epsilon_1)N$ vértices na clique. Ou seja, que para ao menos $(1 - \epsilon_1)N$ vértices de G vale que

$$|\Gamma_C(v)| < (\rho - \epsilon_1)N \implies \Gamma_{U'}(v) \neq U'. \tag{3.1}$$

Dito isso, vamos provar (Corolário 3.6) que se um conjunto de vértices $U \subseteq V(G)$ de tamanho t for sorteado uniformemente, então há alta probabilidade de existir um $U' \subseteq U$ de tamanho $t' = \frac{\rho}{2}t$ que seja ϵ_1-clique-representativo.

Para isso, mostraremos que

1. (Teorema 3.4) Se U tiver tamanho suficiente t, então há alta probabilidade de $|U \cap C| \geq \frac{\rho}{2}t$; e
2. (Teorema 3.5) Um conjunto $U' \subset C$ de $t' = \frac{\rho}{2}t$ vértices sorteados uniformemente em C possui alta probabilidade de ser ϵ_1-clique-representativo.

Teorema 3.4. *Seja $t \geq c\left(\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}\right)$, para uma certa constante positiva c e seja $\epsilon_1 \in [0, \rho]$. Seja U um conjunto de t vértices sorteados uniformemente num grafo G de N vértices com clique C de tamanho $|C| = \rho N$. Então, com probabilidade pelo menos $1 - \frac{\delta}{4}$, temos $|U \cap C| \geq \frac{\rho}{2}t$.*

Prova Sem perda de generalidade, consideremos $U = \{u_1, u_2, \dots, u_t\}$. Sejam $U_i : i \in [t]$, variáveis aleatórias definidas da seguinte forma:

$$U_i = \begin{cases} 1, \text{ se } u_i \in C; \text{ ou} \\ 0, \text{ se } u_i \notin C. \end{cases}$$

Observe que $\sum_{i=1}^{t} U_i = |U \cap C|$.

Seja também p a probabilidade de um vértice de G sorteado uniformemente pertencer à clique C. Então

$$p = \frac{|C|}{|V(G)|} = \frac{\rho N}{N} = \rho.$$

Como cada U_i é uma variável de Bernoulli com probabilidade p, então a média m de suas esperanças é dada por

$$\begin{aligned} m &= \frac{\sum_{i=1}^{t} \mathbb{E}\left[U_i\right]}{t} \\ &= \frac{\sum_{i=1}^{t} p}{t} \\ &= \frac{tp}{t} = p = \rho. \end{aligned}$$

Por uma cota de Chernoff multiplicativa (Apêndice A), temos que $\forall \alpha \in [0, 1]$,

$$\begin{gathered} \mathbb{P}\left(t^{-1}\sum_{i=1}^{t} U_i < (1-\alpha)m\right) < e^{-\alpha^2 mt/2}; \text{ e} \\ \mathbb{P}\left(t^{-1}|U \cap C| < (1-\alpha)\rho\right) < e^{-\alpha^2 \rho t/2}; \text{ e} \\ \mathbb{P}\left(|U \cap C| < (1-\alpha)\rho t\right) < e^{-\alpha^2 \rho t/2}. \end{gathered}$$

Tomando $\alpha = \frac{1}{2}$, temos que

$$\mathbb{P}\left(|U \cap C| < \frac{1}{2}\rho t\right) < e^{-\rho t/8}.$$

Para se obter

$$\mathbb{P}\left(|U \cap C| < \frac{1}{2}\rho t\right) < e^{-\rho t/8} < \frac{\delta}{4},$$

é necessário que

$$\begin{aligned} e^{-\rho t/8} &< \frac{\delta}{4}; \text{ e} \\ \rho t/8 &> -\log\left(\frac{\delta}{4}\right); \text{ e} \\ t &> -8\frac{\log\left(\frac{\delta}{4}\right)}{\rho}. \end{aligned}$$

Por definição, $t \geq c\left(\frac{\log\left(1/(\epsilon_1\delta)\right)}{\epsilon_1\rho}\right)$, para algum c positivo. Assim, basta encontrar c tal que

$$c\frac{\log\left(1/(\epsilon_1\delta)\right)}{\epsilon_1\rho} > -8\frac{\log\left(\delta/4\right)}{\rho}.$$

Como $c\frac{\log\left(1/(\epsilon_1\delta)\right)}{\epsilon_1\rho}$ é decrescente em função de ϵ_1, que está restrito ao intervalo $[0, 1]$, então ela atinge seu valor mínimo em $\epsilon_1 = 1$. Ou seja,

$$c\frac{\log\left(1/(\epsilon_1\delta)\right)}{\epsilon_1\rho} \geq c\frac{\log\left(1/\delta\right)}{\rho}.$$

Dessa forma, para a conclusão que queremos, basta definir um c tal que

$$c\frac{\log(1/\delta)}{\rho} > -8\frac{\log(\delta/4)}{\rho}; \text{ e}$$
$$c\log(1/\delta) > 8\log(4/\delta); \text{ e}$$
$$c > 8\frac{\log(4/\delta)}{\log(1/\delta)}.$$

A função $8\frac{\log(4/\delta)}{\log(1/\delta)}$, por sua vez, é crescente. Como $\delta \in [0, \frac{1}{2})$, então basta que

$$c > 8\frac{\log 8}{\log 2} = 24.$$

Logo, ao tomar $c > 24$, temos que

$$e^{-\rho t/8} < \frac{\delta}{4}.$$

Então tomando $\alpha = \frac{1}{2}$ e $t > 24\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}$, temos

$$\mathbb{P}\left(|U \cap C| < \frac{\rho}{2}t\right) < e^{-\rho t/8} < \frac{\delta}{4}; \text{ e}$$
$$\mathbb{P}\left(|U \cap C| \geq \frac{\rho}{2}t\right) > 1 - \frac{\delta}{4}.$$

E concluímos que, com probabilidade maior que $1 - \delta/4$, o tamanho de $U \cap C$ é pelo menos $\frac{\rho}{2}t$. $\square$

Teorema 3.5. *Sejam $t \geq c\left(\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}\right)$, para uma certa constante positiva c, com $\epsilon_1 \in [0, \rho]$; e $t' = \frac{\rho}{2}t$. Seja G um grafo de N vértices com clique C de tamanho $|C| = \rho N$; e seja U' um conjunto de t' vértices sorteados uniformemente em C. Então, com probabilidade pelo menos $1 - \frac{\delta}{4}$, o conjunto U' é ϵ_1-clique-representativo.*

Prova Lembramos que para U' ser ϵ_1-clique-representativo, ao menos $(1 - \epsilon_1)N$ vértices de G devem satisfazer a Equação 3.1, ou seja, para a maior parte dos vértices v de G, se v é vizinho de todo U', então ele é também vizinho da maior parte da clique, isto é,

$$|\Gamma_C(v)| < (\rho - \epsilon_1)N \implies \Gamma_{U'}(v) \neq U'.$$

Naturalmente, todo $v \in C$ satisfaz a Equação 3.1. Vamos então verificar os casos em que $v \in V(G) \setminus C$.

Seja $X(v)$ o evento “v não satisfaz a Equação 3.1”. Note que para que $X(v)$ aconteça, v deve ser simultaneamente vizinho de menos que $(\rho - \epsilon_1)N$ vértices de C e vizinho de todo U'.

A probabilidade $g(v)$ de um vértice sorteado uniformemente em C ser vizinho de um vértice v é dada pela razão entre a quantidade de vizinhos de v em C e o tamanho de C, isto é,

$$g(v) = \frac{|\Gamma_C(v)|}{|C|}.$$

Quando $X(v)$ ocorre, temos $|\Gamma_C(v)| < (\rho - \epsilon_1)N$. Então,

$$g(v) < \frac{(\rho - \epsilon_1)N}{\rho N} = 1 - \frac{\epsilon_1}{\rho}.$$

E como $\epsilon_1 \leq \rho \leq 1$, então,

$$g(v) < 1 - \epsilon_1.$$

A probabilidade de $X(v)$ para um $v \in V(G)\backslash C$, então, é a de se sortear uniformemente t' vértices em C que sejam vizinhos de v considerando que $|\Gamma_C(v)| < (\rho - \epsilon_1)N$. Ou seja,

$$\mathbb{P}\left(X(v)\right) = g(v)^{t'} < (1 - \epsilon_1)^{t'}.$$

Como $t' = \frac{\rho}{2}t$ e $t \geq c\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}$, para uma certa constante c positiva, então

$$\mathbb{P}\left(X(v)\right) < (1 - \epsilon_1)^{\frac{c}{2}\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1}}.$$

Aplicando algumas identidades logarítmicas, temos

$$(1 - \epsilon_1)^{\frac{c}{2}\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1}} = (\epsilon_1\delta)^{\frac{c}{2}\frac{\log(1/(1-\epsilon_1))}{\epsilon_1}}.$$

Note que tomando $c' = \frac{\log(1/(1-\epsilon_1))}{\epsilon_1}$, temos que $c' > 1$ para todo $\epsilon_1 \in [0, 1]$. Assim,

$$\mathbb{P}\left(X(v)\right) < (\epsilon_1\delta)^{\frac{c}{2}c'}.$$

E portanto

$$\mathbb{P}\left(X(v)\right) < \epsilon_1\delta(\epsilon_1\delta)^{\frac{c}{2}c'-1}.$$

Como $\epsilon_1 \in [0, 1]$ e $\delta \in [0, \frac{1}{2})$, então

$$0 \leq \epsilon_1\delta < \frac{1}{2}$$

E assim, tomando $c > 24$,

$$(\epsilon_1 \delta)^{\frac{c}{2}c'-1} < \left(\frac{1}{2}\right)^{11} < \frac{1}{4}.$$

Enfim, temos que

$$\mathbb{P}\left(X(v)\right) < \frac{\epsilon_1 \delta}{4}.$$

Sejam $X_v : v \in V(G) \setminus C$, as variáveis aleatórias assim definidas:

$$X_v = \begin{cases} 1, & \text{se } X(v)\text{; ou} \\ 0, & \text{caso contrário.} \end{cases}$$

e seja $X = \sum_{v \in V(G) \setminus C} X_v$ a quantidade de vértices que violam a Equação 3.1.

Pela desigualdade de Markov (Apêndice A), temos que para toda constante $a > 0$,

$$\mathbb{P}\left(X < a\mathbb{E}\left[X\right]\right) > 1 - \frac{1}{a}.$$

A esperança de X é dada por:

$$\mathbb{E}\left[X\right] = \sum_{v \in V(G) \setminus C} \mathbb{E}\left[X_v\right] < N \cdot \mathbb{P}\left(X(v)\right) < N\frac{\epsilon_1 \delta}{4}.$$

Tomando $a = \frac{4}{\delta}$, temos

$$\mathbb{P}\left(X < \epsilon_1 N\right) > 1 - \frac{\delta}{4},$$

de onde concluímos que, com probabilidade pelo menos $1 - \frac{\delta}{4}$, há no máximo $\epsilon_1 N$ vértices que violam a Equação 3.1; e portanto, U' é ϵ_1-clique-representativo. □

Corolário 3.6. Seja $t \geq c\left(\frac{\log(1/(\epsilon_1 \delta))}{\epsilon_1 \rho}\right)$, para uma certa constante positiva c, com $\epsilon_1 \in [0, \rho]$. Seja U um conjunto de t vértices sorteados uniformemente num grafo G com clique C de tamanho $|C| = \rho N$. Então, com probabilidade pelo menos $1 - \frac{\delta}{2}$, o conjunto U contém um conjunto ϵ_1-clique-representativo em relação a C de tamanho $\frac{\rho}{2}t$.

Prova Pelo Teorema 3.4, a probabilidade de $|U \cap C| \geq \frac{\rho}{2}t$ é pelo menos $1 - \frac{\delta}{4}$. Dado um conjunto $U' \subseteq U \cap C$ de tamanho $t' = \frac{\rho}{2}t$, temos pelo Teorema 3.5 que, com probabilidade pelo menos $1 - \frac{\delta}{4}$, o conjunto U' é ϵ_1-clique-representativo. Assim, a probabilidade de U conter um conjunto ϵ_1-clique-representativo de tamanho $\frac{\rho}{2}t$ é de pelo menos

$$\left(1 - \frac{\delta}{4}\right)^2 = \left(1 - \frac{\delta}{2} + \frac{\delta^2}{16}\right) \geq 1 - \frac{\delta}{2}.$$

□

Um bom estimador

Agora, provaremos (Teorema 3.8) que se um conjunto de vértices de tamanho suficiente r for sorteado uniformemente, ele será um bom estimador de graus para um conjunto arbitrário T de pelo menos ρN vértices do grafo.

Para isso, é necessário formalizar o que caracteriza um conjunto que ofereça *boas estimativas de graus* para um conjunto de vértices T. Diremos que um conjunto desses é ϵ_2-representativo em relação a T.

Definição 3.7. (Conjunto ϵ_2-representativo) Dado um grafo G e um conjunto $T \subseteq V(G)$, um conjunto $W \subseteq V(G)$ é ϵ_2-representativo em relação a T se para ao menos $(1-\epsilon_2)N$ vértices de G vale que

$$\left| \frac{\deg_{W\cap T}(v)}{|W|} - \frac{\deg_T(v)}{N} \right| \leq \epsilon_2. \tag{3.2}$$

Lembramos que, dado um conjunto $Q \in V(G)$, $\deg_Q(v)$ é o grau do vértice v em $G[Q]$ e equivale a $|\Gamma_Q(v)| = |\Gamma_G(v) \cap Q|$.

Teorema 3.8. *Sejam $r \geq c\left(\frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho}\right)$, para uma certa constante positiva c, onde $\epsilon_2 \in [0,1]$; e seja $T \subseteq V(G)$. Suponha que $|T| \geq \rho N$ e que W é um conjunto de r vértices de $V(G)$ sorteados uniformemente. Então a probabilidade de W ser ϵ_2-representativo em relação a T é ao menos $1-\frac{\delta}{4}$.*

Prova Começamos mostrando que a probabilidade de um vértice ser tal que a Equação 3.2 não vale para ele é menor que $\frac{\epsilon_2\delta}{4}$. Isso será usado para mostrar pela desigualdade de Markov que a probabilidade de que menos que $\epsilon_2 N$ vértices não satisfaçam a Equação 3.2 é maior que $1-\frac{\delta}{4}$, caso em que W é ϵ_2-representativo em relação a T.

Consideremos sem perda de generalidade que $W = \{w_1, \ldots, w_r\}$. Sejam $W_i(v)$: $i \in [r], v \in V(G)$, as variáveis aleatórias definidas da seguinte maneira:

$$W_i(v) = \begin{cases} 1, \text{ se } w_i \text{ é vizinho de } v \text{ e } w_i \in T; \text{ ou} \\ 0, \text{ caso contrário.} \end{cases}$$

Observe que $W(v) = \sum_{i=1}^{r} W_i(v) = \deg_{W\cap T}(v)$.

Seja $p(v) = \frac{\deg_T(v)}{N}$ a probabilidade de um vértice sorteado uniformemente em $V(G)$ ser vizinho de v e estar em T. Então temos que $\mathbb{E}[W(v)] = \sum_{i=1}^{r} \mathbb{E}[W_i(v)] = rp(v)$.

Por cotas de Chernoff aditivas (Apêndice A), temos que para todo $\alpha \in [0,1]$,

$$\mathbb{P}\left(\frac{W(v)}{r} > \frac{\mathbb{E}[W(v)]}{r} + \alpha\right) < e^{-2\alpha^2 r};\text{ e}$$
$$\mathbb{P}\left(\frac{W(v)}{r} < \frac{\mathbb{E}[W(v)]}{r} - \alpha\right) < e^{-2\alpha^2 r}.$$

Disso, obtemos que

$$\mathbb{P}\left(\left|\frac{\deg_{W\cap T}(v)}{r} - \frac{\deg_T(v)}{N}\right| > \alpha\right) < e^{-2\alpha^2 r}.$$

Tomando $\alpha = \epsilon_2$,

$$\mathbb{P}\left(\left|\frac{\deg_{W\cap T}(v)}{r} - \frac{\deg_T(v)}{N}\right| > \epsilon_2\right) < e^{-2\epsilon_2^2 r}.$$

Por definição, $r \geq c\left(\frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho}\right)$, para uma certa constante c. Assim,

$$-2\epsilon_2^2 r \geq 2c\frac{\log(\epsilon_2\delta)}{\rho};$$

E então

$$e^{-2\epsilon_2^2 r} \leq e^{\frac{2c}{\rho}\log(\epsilon_2\delta)} = (\epsilon_2\delta)^{\frac{2}{\rho}c}.$$

Para que $e^{-2\epsilon_2^2 r} < \frac{\epsilon_2\delta}{4}$, basta encontrar c tal que

$$\epsilon_2\delta(\epsilon_2\delta)^{\frac{2}{\rho}c-1} < \epsilon_2\delta\frac{1}{4}.$$

Como $\epsilon_2 \in [0,1]$ e $\delta \in [0,\frac{1}{2})$, então

$$\epsilon_2\delta < \frac{1}{2} \implies (\epsilon_2\delta)^2 < \frac{1}{4}.$$

Assim, basta encontrar c tal que

$$\frac{2}{\rho}c - 1 \geq 2;\text{ e}$$
$$c \geq \frac{3}{2}\rho.$$

Logo, com $\alpha = \epsilon_2$ e $c \geq \frac{3}{2}\rho$ temos que

$$\mathbb{P}\left(\left|\frac{\deg_{W\cap T}(v)}{r} - \frac{\deg_T(v)}{N}\right| > \epsilon_2\right) < \frac{\epsilon_2\delta}{4}.$$

O que conclui a primeira parte da prova. Agora vamos mostrar que a probabilidade de que menos que $\epsilon_2 N$ vértices não satisfaçam a Equação 3.2 é maior que $1 - \frac{\delta}{4}$.

Seja $X(v)$ o evento "v não satisfaz a Equação 3.2" e sejam $X_v, v \in V(G)$, as variáveis aleatórias definidas da seguinte forma:

$$X_v = \begin{cases} 1, \text{ se } X(v); \text{ ou} \\ 0, \text{ caso contrário.} \end{cases}$$

e seja $X = \sum_{v \in V(G)} X_v$ a quantidade de vértices que não satisfazem a Equação 3.2.

Pela desigualdade de Markov (Apêndice A), temos que para toda constante $a > 0$,

$$\mathbb{P}\left(X < a\mathbb{E}\left[X\right]\right) > 1 - \frac{1}{a}.$$

A esperança de X é

$$\mathbb{E}\left[X\right] = \sum_{v \in V(G)} \mathbb{E}\left[X_v\right] = N\mathbb{P}\left(X(v)\right) < N\frac{\epsilon_2 \delta}{4}.$$

Assim, tomando $a = \frac{4}{\delta}$, temos

$$\mathbb{P}\left(X < \epsilon_2 N\right) > 1 - \frac{\delta}{4}.$$

Portanto, com probabilidade maior que $1 - \frac{\delta}{4}$, o conjunto W é ϵ_2-representativo em relação a T. $\square$

Obtenção da quase-clique

Mostraremos agora (Corolário 3.10) que dados um conjunto U' que seja ϵ_1-clique-representativo em relação à clique C (Definição 3.3) e um conjunto W que seja ϵ_2-representativo em relação a $T(U')$ (Definição 3.7), os $|C|$ vértices de $T(U')$ com maior $\deg_{W \cap T(U')}(v)$ formam uma quase-clique.

Para isso, mostraremos primeiro (Teorema 3.9) que se uma boa parte dos vértices de um conjunto $\widetilde{C} \subseteq T(U')$ tiver $\deg_{T(U')}(v)$ grande o suficiente, então $\widetilde{C}$ é uma quase-clique.

Teorema 3.9. *Seja $\epsilon_2 < \frac{\epsilon}{\rho}$ e $\epsilon_1 = \epsilon_2^2$. Seja U' ϵ_1-clique-representativo em relação a C e $T = T(U')$. Seja α tal que αN é o grau em T do $(\rho - \epsilon_2)N$-ésimo vértice de T com maior grau em T. Seja $\widetilde{C} \subseteq T$ um conjunto de tamanho ρN que contém pelo menos $(\rho - 3\epsilon_2)N$ vértices com grau em T pelo menos $(\alpha - 2\epsilon_2)N$. Então $\widetilde{C}$ forma uma $9\epsilon_2\rho$-quase-clique.*

Prova Para esta prova, é necessário retomar a noção de conjunto ϵ_1-clique-representativo (Definição 3.3). Lembramos que para $U' \subseteq V(G)$ ser ϵ_1-clique-representativo em relação à clique C, ao menos $(1-\epsilon_1)N$ vértices de G devem satisfazer a Equação 3.1, isto é,

$$|\Gamma(v) \cap C| < (\rho - \epsilon_1)N \implies \Gamma(v) \cap U' \neq U'.$$

Claramente, $T \supseteq C$. Seja H o conjunto dos vértices de T fora de C com muitos vizinhos em C, mais precisamente,

$$H = \{v \in T \setminus C : |\Gamma(v) \cap C| \geq (\rho - \epsilon_1)N\}.$$

Seja também $R = T \setminus (C \cup H)$ (i.e. o resto de T).

Note que R é composto por vértices de T que não estão em $C \cup H$. Um vértice que esteja nessa situação é ao mesmo tempo vizinho de todo U' e vizinho de menos que $(\rho - \epsilon_1)N$ vértices em C, o que significa que ele não satisfaz a Equação 3.1. Como U' é ϵ_1-clique-representativo, segue que $|R| < \epsilon_1 N$.

Temos que

$$\begin{aligned}
\frac{\sum_{v \in C} \deg_T(v)}{|C|} &\geq \frac{|\{(u,v) : u, v \in C, u \neq v\}| + |\{(u,v) : u \in C, v \in H\}|}{|C|} \\
&\geq \frac{|C|^2 - |C| + |H|(\rho - \epsilon_1)N}{|C|} \\
&> \rho N - 1 + |H| - \frac{\epsilon_1}{\rho} N.
\end{aligned}$$

Denotaremos esse limite inferior da média dos valores de $\deg_T(v)$ para $v \in C$ por μ_m, isto é,

$$\mu_m = \rho N - 1 + |H| - \frac{\epsilon_1}{\rho} N.$$

Por outro lado, o valor máximo de $\deg_T(v)$ para $v \in C$ é limitado superiormente por $|T| - 1 = (|C| + |H| + |R|) - 1 < \rho N - 1 + |H| + \epsilon_1 N$. Denotaremos esse limite superior por

$$\delta_M = \rho N - 1 + |H| + \epsilon_1 N.$$

Assim, a diferença entre o valor máximo e a média de $\deg_T(v)$, com $v \in C$, é menor que $\delta_M - \mu_m = (\epsilon_1 + \frac{\epsilon_1}{\rho})N$.

Pela desigualdade de Markov (Apêndice A), conseguimos que para toda constante positiva k,

$$|\{v \in C : \deg_T(v) < \mu_m - k(\delta_M - \mu_m)\}| < \frac{|C|}{k}; \text{ e}$$
$$\left|\left\{v \in C : \deg_T(v) < \left(\rho N + |H| - \frac{\epsilon_1}{\rho} N\right) - k\left(\epsilon_1 + \frac{\epsilon_1}{\rho}\right) N\right\}\right| < \frac{|C|}{k}.$$

Usando $k = \rho/\epsilon_2$, temos $|C|/k = \epsilon_2 N$ e $k(\epsilon_1 + \frac{\epsilon_1}{\rho}) \leq 2\epsilon_2$. Assim, pelo menos $(\rho - \epsilon_2)N$ vértices em C têm $\deg_{T(U')}(v)$ de pelo menos $\rho N + |H| - \frac{\epsilon_1}{\rho}N - 2\epsilon_2 N$. Como $\epsilon_1/\rho = \epsilon_2^2/\rho$, $\epsilon_2 < \epsilon/\rho$ e $\epsilon < \rho^2$, temos que α, como definido no enunciado, satisfaz

$$\alpha \geq \rho + \frac{|H|}{N} - 3\epsilon_2 = \rho + \frac{|T| - (|R| + |C|)}{N} - 3\epsilon_2 > \frac{|T|}{N} - 4\epsilon_2.$$

Pelo enunciado, temos que $|\widetilde{C}| = \rho N$. Além disso, denotando $Z \subseteq \widetilde{C}$ o conjunto dos vértices v em que $\deg_T(v) \geq (\alpha - 2\epsilon_2)N$, temos pelo enunciado que $|Z| \geq (\rho - 3\epsilon_2)N$. Pelo que acabamos de demonstrar acerca de α, temos para cada $v \in Z$,

$$\begin{aligned}
\deg_{\widetilde{C}}(v) &\geq \deg_T(v) - |T \setminus \widetilde{C}| \\
&> [(|T| - 4\epsilon_2 N) - 2\epsilon_2 N] - (|T| - \rho N) \\
&= (\rho - 6\epsilon_2)N.
\end{aligned}$$

Somando os graus (em $\widetilde{C}$) de todos os vértices em $\widetilde{C}$, obtemos

$$\begin{aligned}
\sum_{v \in \widetilde{C}} \deg_{\widetilde{C}}(v) &\geq \sum_{v \in Z} \deg_{\widetilde{C}}(v) \\
&> |Z|((\rho - 6\epsilon_2)N) \\
&\geq (\rho - 3\epsilon_2)(\rho - 6\epsilon_2)N^2 \\
&> (\rho^2 - 9\epsilon_2\rho)N^2 = |\widetilde{C}|^2 - 9\epsilon_2\rho N^2.
\end{aligned}$$

Segue que $\widetilde{C}$ está $9\epsilon_2\rho$-perto de ser uma clique e então o grafo induzido por $\widetilde{C}$ satisfaz a proposição. □

Corolário 3.10. Sejam C uma ρ-clique em G, $\epsilon_2 < \frac{\epsilon}{\rho}$ e $\epsilon_1 = \epsilon_2^2$. Suponha que $U' \subseteq C$ é ϵ_1-clique-representativo em relação a C e que W é ϵ_2-representativo em relação a $T = T(U')$. Então o conjunto dos ρN vértices de maior $\hat{d}(v) := \deg_{W \cap T}(v)$ está $9\epsilon_2\rho$-perto de ser uma ρ-clique.

Prova Para esta prova, retomamos a definição de conjunto ϵ_2-representativo (Definição 3.7). Dado um conjunto $T \subseteq V(G)$, dizemos que um conjunto de vértices $W \subseteq V(G)$ é ϵ_2-representativo em relação a T se para ao menos $(1 - \epsilon_2)N$ vértices de G vale a Equação 3.2, isto é,

$$\left| \frac{\deg_{W \cap T}(v)}{|W|} - \frac{\deg_T(v)}{N} \right| \leq \epsilon_2.$$

Seja $\widetilde{C}$ o conjunto de ρN vértices de T com maior $\hat{d}$.

Dado que W é ϵ_2-representativo em relação a T, segue-se que pelo menos $(1-\epsilon_2)N$ vértices de $V(G)$ satisfazem

$$\frac{\hat{d}(v)}{|W|} \geq \frac{\deg_T(v)}{N} - \epsilon_2.$$

Seja α como definido no Teorema 3.9. Então $\deg_T(v) \geq \alpha N$ para pelo menos $(\rho - \epsilon_2)N$ vértices em T. Assim, temos que pelo menos $(\rho - 2\epsilon_2)N$ vértices de T satisfazem

$$\frac{\hat{d}(v)}{|W|} \geq \alpha - \epsilon_2.$$

Como $\widetilde{C}$ contém vértices de T com os maiores $\hat{d}$, ele deve conter pelo menos $(\rho - 2\epsilon_2)N$ vértices em que $\hat{d}(v) \geq (\alpha - \epsilon_2)|W|$.

Por outro lado, sendo W ϵ_2-representativo, é verdade que $\deg_T(v) < \left(\frac{\hat{d}(v)}{|W|} - \epsilon_2\right)N$ para no máximo $\epsilon_2 N$ vértices de G. Disso, concluímos que $\widetilde{C}$ contém pelo menos $(\rho - 2\epsilon_2)N - \epsilon_2 N$ vértices de $\deg_T(v)$ pelo menos

$$\left(\frac{\hat{d}(v)}{|W|} - \epsilon_2\right)N \geq \left(\frac{(\alpha - \epsilon_2)|W|}{|W|} - \epsilon_2\right)N \geq (\alpha - 2\epsilon_2)N.$$

Enfim, como U' é ϵ_1-clique-representativo, pelo Teorema 3.9 temos que $\widetilde{C}$ está $9\epsilon_2\rho$-perto de ser uma ρ-clique. □

3.4.3 Valores das constantes

O sucesso de ENCONTRAρ depende de que o Corolário 3.6, o Teorema 3.8 e o Corolário 3.10 possam ser aplicados durante a execução do algoritmo, implicando a obtenção de uma ϵ-quase-clique. Assim, é necessário definir que valores de c_t e c_r são adequados para que o algoritmo funcione devidamente.

Como do Corolário 3.10 temos a obtenção de uma $9\epsilon_2\rho$-quase-clique e queremos uma ϵ-quase-clique, podemos tomar $\epsilon_2 = \frac{\epsilon}{9\rho} < \frac{\epsilon}{\rho}$ e $\epsilon_1 = \epsilon_2^2 = \frac{\epsilon^2}{81\rho^2}$.

Valor de c_t Da prova do Corolário 3.6, precisamos que $t > 24\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}$. Assim, temos que

$$t > 24\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho} = 24 \cdot 81\frac{\rho\log(81\rho^2/(\epsilon^2\delta))}{\epsilon^2}.$$

Do Algoritmo 1, temos $t = c_t \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2}$. Assim, precisamos encontrar c_t tal que

$$c_t \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2} > 24 \cdot 81 \frac{\rho \log(81\rho^2/(\epsilon^2\delta))}{\epsilon^2}; \text{ e}$$
$$c_t > 24 \cdot 81 \frac{\log(81\rho^2/(\epsilon^2\delta))}{\log(1/(\epsilon\delta))}.$$

Como $\frac{\log(81\rho^2/(\epsilon^2\delta))}{\log(1/(\epsilon\delta))}$ é crescente em função de $\rho \leq 1$, de $\delta < \frac{1}{2}$ e de $\epsilon \leq 1$, então fixando c_t tal que

$$c_t > 24 \cdot 81 \frac{\log(81 \cdot 1^2/(1^2 \cdot \frac{1}{2}))}{\log(1/(1 \cdot \frac{1}{2}))} = 1944 \frac{\log(162)}{\log(2)} \approx 14268.67$$

temos um valor adequado de c_t para quaisquer valores de ϵ, δ e ρ.

Valor de c_r Da prova do Teorema 3.8, precisamos que $r \geq \frac{3}{2}\rho \frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho}$. Assim, é necessário que

$$r \geq \frac{3}{2}\rho \frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho} = \frac{3}{2} 81\rho \frac{\rho \log(9\rho/(\epsilon\delta))}{\epsilon^2}.$$

Mas como do Algoritmo 1 temos $r = c_r \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2}$, então precisamos de um c_r tal que

$$c_r \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2} > \frac{3}{2} 81\rho \frac{\rho \log(9\rho/(\epsilon\delta))}{\epsilon^2}; \text{ e}$$
$$c_r > \frac{3}{2} 81 \frac{\rho \log(9\rho/(\epsilon\delta))}{\log(1/(\epsilon\delta))}.$$

E como $\frac{\rho \log(9\rho/(\epsilon\delta))}{\log(1/(\epsilon\delta))}$ é crescente em função de $\rho \leq 1$, $\epsilon \leq 1$ e $\delta < \frac{1}{2}$, então basta que

$$c_r > \frac{3}{2} 81 \frac{1 \cdot \log(9 \cdot 1/(1 \cdot \frac{1}{2}))}{\log(1/(1 \cdot \frac{1}{2}))} = \frac{243}{2} \frac{\log(18)}{\log(2)} \approx 506.65$$

o qual será um valor adequado para todo ϵ, δ e ρ dados.

3.5 Testador de Grau da Clique

Algoritmo 2: Testador de Grau da Clique (TESTAρ)

Instância : Um grafo G

Resposta : sim , se $G \in C_\rho$; ou
não , se G está ϵ-longe de C_ρ.

$t \leftarrow \left\lceil c_t \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2} \right\rceil$

$r \leftarrow \left\lceil c_r \frac{\rho \log(1/(\epsilon\delta))}{\epsilon^2} \right\rceil$

$m \leftarrow \left\lceil c_m \frac{t + \log(1/\delta)}{\epsilon^2} \right\rceil$

$U \leftarrow t$ vértices de $V(G)$ sorteados uniformemente

$W \leftarrow r$ vértices de $V(G)$ sorteados uniformemente

$S = \{s_1, \ldots, s_m\} \leftarrow m$ vértices de $V(G)$ sorteados uniformemente

para cada *$U' \subseteq U$ de tamanho $\lceil t\frac{\rho}{2} \rceil$* **faça**
  $S(U') \leftarrow \{s \in S : \Gamma_G(s) \cap U' = U'\}$
  $W(U') \leftarrow \{w \in W : \Gamma_G(w) \cap U' = U'\}$
  se $|S(U')| < \lfloor \rho m \rfloor$ **então**
    $\hat{C}(U') \leftarrow S(U')$
  senão
    $\hat{C}(U') \leftarrow \lfloor \rho m \rfloor$ vértices de $S(U')$ com maior $|\Gamma_G(v) \cap W(U')|$
  se
  (Condição 1) $|\hat{C}(U')| \geq (\rho - \epsilon/80)m$; e
  (Condição 2) $|\{(s_{2k-1}, s_{2k}) \in (\hat{C}(U') \times \hat{C}(U')) \setminus E(G)\}| \leq \frac{2\epsilon}{3}\frac{m}{2}$ **então**
    devolva *sim*
devolva *não*

Lembramos que C_ρ é definido como o conjunto de todos os grafos que possuem ρ-clique, isto é $C_\rho := \{G : \omega(G) \geq \rho|V(G)|\}$.

Observe que ENCONTRAρ, como descrito no Algoritmo 1, constrói para cada U' um candidato a ϵ-quase-clique $C(U')$ através da ordenação dos vértices de $T(U')$, os vizinhos de todo U'. Como $T(U')$ deve ter tamanho pelo menos $|C| = \rho N$ para conter uma quase-clique, a construção de $C(U')$ leva tempo pelo menos linear em função de ρN.

O testador TESTAρ descrito no Algoritmo 2, por sua vez, coleta uma amostra $\hat{C}(U')$ de $C(U')$ para cada U' evitando a construção de cada candidato e determina através disso se $C(U')$ é uma ϵ-quase-clique. Isso é feito através de um conjunto adicional de vértices S e considerando a ordenação dos vértices de S vizinhos de todo U', isto é, $S(U') = S \cap T(U')$.

3.5.1 Prova de corretude

A corretude do testador provém de duas observações:

1. Com alta probabilidade, existe uma iteração em que os conjuntos U' e W possuem as características necessárias para que o Corolário 3.10 possa ser aplicado; e

2. O conjunto S é uma *boa amostra* de $T(U')$.

Nesta seção, provaremos esta segunda observação (Teorema 3.12) e demonstraremos a corretude de ENCONTRAρ (Corolário 3.13). No entanto, antes disso provaremos a Proposição 3.11, que será usada no Teorema 3.12.

Proposição 3.11. *Seja $S = \{s_1, \ldots, s_m\}$ um conjunto de m vértices sorteados uniformemente, com $m \geq \frac{t+\log(32/\delta)}{2\epsilon_3^2}$, sendo $t > 0$ e $\epsilon_3 \in [0, 1]$. Então, para qualquer conjunto de vértices X,*

$$\mathbb{P}\left(\left|\frac{|S \cap X|}{m} - \frac{|X|}{N}\right| > \epsilon_3\right) < \frac{\delta}{16}2^{-t}.$$

Prova Sejam $Z_i : i \in [m]$, as variáveis aleatórias assim definidas:

$$Z_i = \begin{cases} 1, \text{ se } s_i \in X; \text{ ou} \\ 0, \text{ caso contrário.} \end{cases}$$

e seja $Z = \frac{\sum_{i=1}^m Z_i}{m}$ a média de todos os Z_i. Note que $\sum_{i=1}^m Z_i = |S \cap X|$.

Seja também $p = \frac{|X|}{N}$ a probabilidade de um vértice sorteado uniformemente em G estar em X. Então

$$\mathbb{E}[Z] = \frac{\sum_{i=1}^m \mathbb{E}[Z_i]}{m} = p = \frac{|X|}{N}$$

Pelas cotas de Chernoff aditivas (Apêndice A), temos que para todo $\alpha \in [0, 1]$,

$$\mathbb{P}(|Z - \mathbb{E}[Z]| > \alpha) < e^{-2\alpha^2 m}; \text{ e}$$
$$\mathbb{P}\left(\left|\frac{|S \cap X|}{m} - \frac{|X|}{N}\right| > \alpha\right) < e^{-2\alpha^2 m}.$$

Tomando $\alpha = \epsilon_3$, temos

$$\mathbb{P}\left(\left|\frac{|S \cap X|}{m} - \frac{|X|}{N}\right| > \epsilon_3\right) < e^{-2\epsilon_3^2 m}.$$

Por definição, $m \geq \frac{t+\log(32/\delta)}{2\epsilon_3^2}$. Assim,

$$-2\epsilon_3^2 m \leq -2\epsilon_3^2 \frac{t + \log(32/\delta)}{2\epsilon_3^2} = -t - \log(32/\delta).$$

Logo,

$$e^{-2\epsilon_3^2 m} \leq \frac{\delta}{32}e^{-t} = \frac{\delta}{16}\frac{e^{-t}}{2}.$$

E como $\frac{e^{-t}}{2} < 2^{-t}$ para $t > 0$, obtemos

$$\mathbb{P}\left(\left|\frac{|S \cap X|}{m} - \frac{|X|}{N}\right| > \epsilon_3\right) < \frac{\delta}{16}2^{-t}.$$

$\square$

Teorema 3.12. *Seja X qualquer subconjunto de $V(G)$ e assuma que $|X| \geq (\rho - \epsilon/40)N$. Considere uma ordem fixa $x_1, \ldots, x_{|X|}$ dos vértices de X e seja X' o conjunto dos primeiros min$\{\rho N, |X|\}$ vértices de X de acordo com essa ordem. Seja $S = \{s_i, \ldots, s_m\}$ um conjunto de vértices de $V(G)$ sorteados uniformemente de tamanho $m \geq c\frac{t+\log(1/\delta)}{\epsilon^2}$, para uma certa constante c; e seja $S' \subseteq S$ o conjunto dos primeiros min$\{\lfloor \rho m \rfloor, |S \cap X|\}$ vértices em $S \cap X$ segundo a ordem definida em X. Então*

$$\mathbb{P}\left(\left|\frac{|\{(s_{2k-1}, s_{2k}) \in E(S', S')\}|}{m/2} - \frac{|E(X', X')|}{N^2}\right| > \frac{\epsilon}{3}\right) < \frac{\delta}{8}2^{-t}.$$

(Dados dois conjuntos de vértices A e B, denotaremos por $E(A, B)$ o conjunto das arestas com um vértice em A e outro em B).

Prova Seja $\epsilon_3 = \epsilon/40$ e X'' os primeiros $\lfloor(\rho - \epsilon_3)N\rfloor$ vértices em X. Pelo Corolário 3.13, temos

$$\mathbb{P}\left(\left|\frac{|S \cap X''|}{m} - (\rho - \epsilon_3)\right| > \epsilon_3\right) < \frac{\delta}{16}2^{-t}.$$

Assim, vamos assumir que S é tal que

Propriedade 1 $|S \cap X''| \leq \lfloor \rho m \rfloor$, do que temos que $S' \cap X'' = S \cap X''$;

Propriedade 2 $|S \cap X''| \geq \lfloor(\rho - 2\epsilon_3)m\rfloor$, do que temos que $|S' \cap X''| \geq (\rho - 2\epsilon_3)m$.

Observe que

$$\begin{aligned}\{(s_{2k-1}, s_{2k}) \in E(S', S')\} = &\{(s_{2k-1}, s_{2k} \in E(S' \cap X'', S' \cap X'')\} \\ &\bigcup \{(s_{2k-1}, s_{2k}) \in E(S' \setminus X'', S')\}.\end{aligned} \tag{3.3}$$

Seja $Z_i : i \in [m/2]$, uma variável aleatória assim definida:

$$Z_i := \begin{cases} 1, \text{ se } (s_{2i-1}, s_{2i}) \in E(X'', X''); \text{ ou} \\ 0, \text{ caso contrário.} \end{cases}$$

Naturalmente,

$$\begin{aligned} Z := \frac{\sum_{i \in [m/2]} Z(i)}{m/2} &= \frac{|\{(s_{2k-1}, s_{2k}) \in E(S \cap X'', S \cap X'')\}|}{m/2}; \text{ e} \\ &= \frac{|\{(s_{2k-1}, s_{2k}) \in E(S' \cap X'', S' \cap X'')\}|}{m/2}, \end{aligned}$$

onde essa última igualdade é proveniente da Propriedade 1 mencionada acima.

Além disso, a probabilidade de dois vértices sorteados uniformemente em $V(G)$ estarem ambos em X'' e serem vizinhos é $p = \frac{|E(X'',X'')|}{N^2}$. Portanto, a esperança $\mathbb{E}[Z]$ é dada por

$$\mathbb{E}[Z] = \frac{\sum_{i\in[m/2]} \mathbb{E}[Z_i]}{m/2} = \frac{(m/2)p}{m/2} = \frac{|E(X'',X'')|}{N^2}.$$

Por uma cota de Chernoff aditiva (Apêndice A), temos que para todo $\alpha \in [0,1]$,

$$\mathbb{P}(|Z - \mathbb{E}[Z]| < \alpha) > 1 - e^{-\alpha^2 m}.$$

Tomando $\alpha = \frac{\epsilon}{9}$, temos que

$$\begin{aligned}
&\mathbb{P}\left(|Z - \mathbb{E}[Z]| < \frac{\epsilon}{9}\right) \\
&= \mathbb{P}\left(\left|\frac{|\{(s_{2k-1}, s_{2k}) \in E(S' \cap X'', S' \cap X'')\}|}{m/2} - \frac{|E(X'',X'')|}{N^2}\right| < \frac{\epsilon}{9}\right) > 1 - e^{-\epsilon^2 m/81}
\end{aligned}$$

Para que $e^{-\epsilon^2 m/81} < \frac{\delta}{16}2^{-t}$, é necessário que

$$\begin{aligned}
-\frac{\epsilon^2 m}{81} &< \log(2^{-t}\delta/16) \\
m &> -81\frac{\log(\delta/2^{t+4})}{\epsilon^2}
\end{aligned}$$

Como por definição $m \geq c\frac{t+\log(1/\delta)}{\epsilon^2}$ para uma certa constante c, então basta que

$$\begin{aligned}
&c\frac{t+\log(1/\delta)}{\epsilon^2} > 81\frac{\log(2^{t+4}/\delta)}{\epsilon^2}; \text{ e} \\
&c(t+\log(1/\delta)) > 81\log(2^{t+4}/\delta); \text{ e} \\
&c > 81\frac{\log(2^{t+4}/\delta)}{t+\log(1/\delta)}
\end{aligned}$$

E como $\frac{\log(2^{t+4}/\delta)}{t+\log(1/\delta)}$ é crescente em função de δ, que está restrito ao intervalo $[0, \frac{1}{2})$,

$$81\frac{\log(2^{t+4}/\delta)}{t+\log(1/\delta)} < 81\lim_{\delta\to\frac{1}{2}} \frac{t+\log(1/\delta)}{\log(2^{t+4}/\delta)} = 81\frac{\log(2^{5+t})}{t+\log(2)}.$$

A função $\frac{\log(2^{5+t})}{t+\log(2)}$, por sua vez, é descrescente em $t \geq 0$. Dessa forma,

$$81\frac{\log(2^{5+t})}{t+\log(2)} < 81\lim_{t\to 0}\frac{\log(2^{5+t})}{t+\log(2)} = 81\cdot 5.$$

Portanto, para $e^{-\epsilon^2 m/81} < \frac{\delta}{16}2^{-t}$, basta tomarmos $c > 81 \cdot 5 = 405$.

Assim, temos que

$$\mathbb{P}\left(\left|\frac{|\{(s_{2k-1}, s_{2k}) \in E(S' \cap X'', S' \cap X'')\}|}{m/2} - \frac{|E(X'', X'')|}{N^2}\right| < \frac{\epsilon}{9}\right) > 1 - \frac{\delta}{16}2^{-t} \qquad (3.4)$$

Pela definição de X' e X'', temos que

$$-\frac{\epsilon}{9} < \frac{|E(X', X')|}{N^2} - \frac{|E(X'', X'')|}{N^2} \leq 2\epsilon_3\rho < \frac{\epsilon}{9} \qquad (3.5)$$

Pelo item 2 acima, sabemos que $|S' \setminus X''| \leq 2\epsilon_3 m$, e por isso

$$-\frac{\epsilon}{9} < \frac{|\{(s_{2k-1}, s_{2k}) \in E(S' \setminus X'', S')\}|}{m/2} \leq \frac{|\{s_k \in S' \setminus X''\}|}{m/2} \leq 4\epsilon_3 < \frac{\epsilon}{9}. \qquad (3.6)$$

Usando a Equação 3.5 e a Equação 3.6 na Equação 3.4, temos

$$\mathbb{P}\left(\left|\frac{|\{(s_{2k-1}, s_{2k}) \in E(S', S')\}|}{m/2} - \frac{|E(X', X')|}{N^2}\right| < \frac{\epsilon}{3}\right) > 1 - \frac{\delta}{16}2^{-t}$$

Assim, a probabilidade que queremos pode ser calculada em termos das probabilidades de que aconteça

$$\left|\frac{|S \cap X''|}{m} - (\rho - \epsilon_3)\right| > \epsilon_3$$

ou que aconteça

$$\left|\frac{|\{(s_{2k-1}, s_{2k}) \in E(S', S')\}|}{m/2} - \frac{|E(X', X')|}{N^2}\right| > \frac{\epsilon}{3},$$

isto é,

$$\mathbb{P}\left(\left|\frac{|\{(s_{2k-1}, s_{2k}) \in E(S', S')\}|}{m/2} - \frac{|E(X', X')|}{N^2}\right| > \frac{\epsilon}{3}\right) < \frac{\delta}{16}2^{-t} + \frac{\delta}{16}2^{-t} = \frac{\delta}{8}2^{-t}.$$

□

Corolário 3.13. Seja A o algoritmo TESTAρ e G um grafo.

$$\begin{aligned} G \in C_\rho \quad &\Longrightarrow \quad \mathbb{P}\,(A(G) = \text{ sim }) > 1 - \delta;\text{ e} \\ G \text{ é } \epsilon\text{-longe de } C_\rho \quad &\Longrightarrow \quad \mathbb{P}\,(A(G) = \text{ não }) > 1 - \delta. \end{aligned}$$

Prova Seja $\epsilon_2 := \frac{\epsilon}{27\rho}$ e seja $\epsilon_1 := \epsilon_2^2$. Vamos começar provando que se $G \in C_\rho$, então A aceita a G com probabilidade maior que $1 - \delta$.

Seja G um grafo com uma clique C de tamanho ρN. Pelo Corolário 3.6, temos que se um conjunto de vértices U tiver tamanho pelo menos $t \geq c_t \frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}$, para uma certa constante c_t, então com probabilidade pelo menos $1 - \frac{\delta}{2}$ ele contém um subconjunto U' que é ϵ_1-clique-representativo em relação a C de tamanho $\frac{\rho}{2}t$.

Assim, para que este teorema possa ser aplicado ao algoritmo, é necessário que para uma certa constante c_t,

$$c_t\frac{\rho\log(1/(\epsilon\delta))}{\epsilon^2} > c'_t\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho}.$$

Como tomamos $\epsilon_1 = \epsilon_2^2 = \frac{\epsilon^2}{27^2\rho^2}$, então

$$c'_t\frac{\log(1/(\epsilon_1\delta))}{\epsilon_1\rho} = 27^2c'_t\frac{\rho\log(27^2\rho^2/(\epsilon^2\delta))}{\epsilon^2}.$$

Logo, basta escolher c_t tal que

$$c_t\frac{\rho\log(1/(\epsilon\delta))}{\epsilon^2} > 27^2c'_t\frac{\rho\log(27^2\rho^2/(\epsilon^2\delta))}{\epsilon^2};\text{ e}$$
$$c_t > 27^2c'_t\frac{\log(27^2\rho^2/(\epsilon^2\delta))}{\log(1/(\epsilon\delta))}.$$

Note que $\frac{\log(27^2\rho^2/(\epsilon^2\delta))}{\log(1/(\epsilon\delta))}$ é crescente em função de $\rho \le 1$, $\epsilon \le 1$ e $\delta < \frac{1}{2}$. Por isso,

$$\frac{\log(27^2\rho^2/(\epsilon^2\delta))}{\log(1/(\epsilon\delta))} < \frac{\log(27^2\cdot 1^2/(1^2\cdot\frac{1}{2}))}{\log(1/(1\cdot\frac{1}{2}))} = \frac{\log(2\cdot 27^2)}{\log(2)}.$$

Portanto, basta escolher c_t tal que

$$c_t > 27^2c'_t\frac{\log(2\cdot 27^2)}{\log(2)} \approx 7661.63c'_t.$$

Consideraremos os passos do laço do algoritmo com esse subconjunto ϵ_1-clique-representativo U' de tamanho $t' := \frac{\rho}{2}t$. Lembramos que como $U' \subseteq C$, então $|T(U')| \ge \rho N$.

Pelo Teorema 3.8, se W tem tamanho pelo menos $r \ge c_r\frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho}$ e $|T(U')|$ é pelo menos ρN, então com probabilidade pelo menos $1-\frac{\delta}{4}$ o conjunto W é ϵ_2-representativo em relação a $T(U')$. Para isso, precisamos que para uma certa constante c_r

$$c_r\frac{\rho\log(1/(\epsilon\delta))}{\epsilon^2} > c'_r\frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho}$$

Como tomamos $\epsilon_2 = \frac{\epsilon}{27\rho}$, então

$$c'_r\frac{\log(1/(\epsilon_2\delta))}{\epsilon_2^2\rho} = c'_r27^2\frac{\rho\log(27\rho/(\epsilon\delta))}{\epsilon^2}$$

Logo, basta encontrar c_r tal que

$$c_r\frac{\rho\log(1/(\epsilon\delta))}{\epsilon^2} > c'_r27^2\frac{\rho\log(27\rho/(\epsilon\delta))}{\epsilon^2};\text{ e}$$
$$c_r > c'_r27^2\frac{\log(27\rho/(\epsilon\delta))}{\log(1/(\epsilon\delta))}.$$

Como $\frac{\log(27\rho/(\epsilon\delta))}{\log(1/(\epsilon\delta))}$ é crescente em função de $\rho \leq 1$, $\epsilon \leq 1$ e $\delta < \frac{1}{2}$, então

$$\frac{\log(27\rho/(\epsilon\delta))}{\log(1/(\epsilon\delta))} < \frac{\log(27 \cdot 1/(1 \cdot \frac{1}{2}))}{\log(1/(1 \cdot \frac{1}{2}))} = \frac{\log(2 \cdot 27)}{\log(2)}.$$

Assim, basta que

$$c_r > c_r' \frac{\log(2 \cdot 27)}{\log(2)} \approx 5,75 c_r'.$$

Seja $\widetilde{C}$ o conjunto dos ρN vértices de $T(U')$ com maior grau em $W \cap T(U')$. Pelo Corolário 3.10 e tomando $9\epsilon_2\rho = \frac{\epsilon}{3}$, temos que o conjunto $\widetilde{C}$ está $\frac{\epsilon}{3}$-perto de ser uma ρ-clique.

Sempre que $|S(U')| \geq \lfloor \rho m \rfloor$, temos que $|\hat{C}(U')| = \lfloor \rho m \rfloor$ e a Condição 1 ao fim do laço do algoritmo TESTAρ é satisfeita. Por outro lado, quando $|S(U')| < \lfloor \rho m \rfloor$, temos $|\hat{C}(U')| = |S(U')|$. Nesse caso, aplicando a Proposição 3.11 em $T(U')$ com $\epsilon_3 = \frac{\epsilon}{80}$, temos que a Condição 1 do algoritmo é satisfeita com probabilidade maior que $1 - \frac{\delta}{8}$, pois

$$\begin{gathered} -\frac{\epsilon}{80} \leq \frac{|S(U')|}{m} - \frac{|T(U')|}{N} \leq \frac{|S(U')|}{m} - \rho; \text{ e} \\ \left(\rho - \frac{\epsilon}{80}\right) m \leq |S(U')| = |C(U')|. \end{gathered}$$

Além disso, se $\widetilde{C}(U')$ é uma $\frac{\epsilon}{3}$-quase-clique, então pelo Teorema 3.12, com $X = T(U')$ e a ordem de X determinada por $\deg_{W \cap T(U')}(v)$, a Condição 2 é satisfeita com probabilidade maior que $1 - \frac{\delta}{8}$.

Dessa forma, $\mathbb{P}\left(A(G) = \text{ sim }\right) > 1 - \left(\frac{\delta}{2} + \frac{\delta}{4} + \frac{\delta}{8} + \frac{\delta}{8}\right) = 1 - \delta$.

Vamos agora provar que se G está ϵ-longe de C_ρ, então $\mathbb{P}\left(A(G) = \text{ não }\right) > 1 - \delta$.

Seja então G um grafo ϵ-longe de ter uma ρ-clique. Para qualquer dada escolha de U' e W, consideramos o conjunto $\widetilde{C}(U')$ dos $\min\{\rho N, |T(U')|\}$ vértices de $T(U')$ com maior grau $\deg_{W \cap T(U')}(v)$.

Observe que quando $|\widetilde{C}(U')| \leq (\rho - \frac{\epsilon}{40})N$, então necessariamente $\widetilde{C}(U') = T(U')$. Nesse caso, podemos verificar que com probabilidade ao menos $1 - 2^{-t}\delta$ a Condição 1 deixa de ser satisfeita aplicando a Proposição 3.11 sobre $T(U')$, pois tomando $\epsilon_3 = \frac{\epsilon}{80}$, temos

$$\begin{gathered} \frac{|\hat{C}(U')|}{m} - \frac{|T(U')|}{N} < \frac{\epsilon}{80}; \text{ e} \\ \frac{|\hat{C}(U')|}{m} - \left(\rho - \frac{\epsilon}{40}\right) < \frac{\epsilon}{80}; \text{ e} \\ |\hat{C}(U')| < \left(\rho - \frac{\epsilon}{80}\right) m. \end{gathered}$$

Por outro lado, quando $|\widetilde{C}(U')| > (\rho - \frac{\epsilon}{40})N$, lembramos que $d(\widetilde{C}(U'), C_N^\rho) \geq \epsilon$. Assim, aplicando o Teorema 3.12, com $X = T(U')$ e a ordem de X determinada por $\deg_{W \cap T(U')}(v)$, temos que a Condição 2 é violada com probabilidade maior que $1 - 2^{-t}\delta$.

Assim, para qualquer escolha possível de U' e W, a probabilidade do algoritmo aceitar a entrada é limitada por $2^{-t}\delta$. Como U e W são sorteados e menos que 2^t subconjuntos $U' \subset U$ são considerados, temos que $\mathbb{P}\left(A(G) = \textsf{não}\right) > 1 - \delta$. $\square$

Capítulo 4

Considerações Finais

No presente trabalho, apresentamos o teste de propriedades, suas principais características, a terminologia pertinente e um breve histórico dos estudos acerca do assunto. Em seguida, apresentamos o modelo denso para grafos proposto em Goldreich (2011) usado em testadores de grafos e passamos a estudar o Testador de Grau da Clique proposto em Goldreich et al. (1998), explicitando detalhes da prova de sua corretude que estão ausentes no trabalho original.

Além disso, foram explicitados valores apropriados no contexto da prova de corretude para as constantes de importância do Testador de Grau da Clique. Entretanto, tais valores foram superestimados e considerando-os na análise de complexidade do testador conclui-se que sua implementação é impraticável. A investigação de valores que amenizem o impacto das constantes na complexidade do algoritmo pode ser considerada em trabalhos futuros.

Enfim, o testador estudado no presente trabalho é um caso especial do *Problema Geral de Teste de Partição de Grafos* apresentado em Goldreich et al. (1998), o qual não foi estudado no presente trabalho e também pode ser objeto de estudos futuros.

Referências Bibliográficas

Noga Alon e Asaf Shapira. A characterization of the (natural) graph properties testable with one-sided error. *SIAM Journal on Computing*, 37(6):1703–1727, 2008.

Noga Alon, Eldar Fischer, Ilan Newman e Asaf Shapira. A combinatorial characterization of the testable graph properties: it's all about regularity. *SIAM Journal on Computing*, 39(1): 143–167, 2009.

Manuel Blum, Michael Luby e Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 73–83. ACM, 1990.

Eldar Fischer. The art of uninformed decisions: A primer to property testing. *Current Trends in Theoretical Computer Science: The Challenge of the New Century*, 1:229–264, 2004.

Michael R. Garey e David S. Johnson. *Computers and intractability*. Freeman San Francisco, 1979.

Oded Goldreich. Introduction to testing graph properties. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness e Computation*, pages 470–506. Springer, 2011.

Oded Goldreich, Shari Goldwasser e Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

Richard M Karp. Reducibility among combinatorial problems. In *Complexity of computer computations*, pages 85–103. Springer, 1972.

Christos H Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994. ISBN 0201530821,9780201530827.

Ronitt Rubinfeld e Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

Qinghua Wu e Jin-Kao Hao. A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3):693–709, 2015.

Apêndice A

Desigualdades úteis

Teorema A.1 (Desigualdade de Markov)**.** *Seja X uma variável aleatória que assuma apenas valores não-negativos. Então, para toda constante $a \in \mathbb{R}$, $a > 0$,*

$$\mathbb{P}\left(X \geq a\mathbb{E}\left[X\right]\right) \leq \frac{1}{a}$$

Teorema A.2 (Desigualdade de Chebyshev)**.** *Para toda constante $a \in \mathbb{R}$, $a > 0$ e qualquer variável aleatória S,*

$$\mathbb{P}\left(\left|S - \mathbb{E}\left[S\right]\right| \geq a\right) \leq \frac{\mathit{Var}\left(S\right)}{a^2}$$

Teorema A.3 (Cotas de Chernoff)**.** *Sejam $X_1, X_2, \ldots, X_m$ variáveis aleatórias independentes, onde $\forall i \in [m], X_i \in [0,1]$. Seja $p = m^{-1} \sum_{i=1}^{m} \mathbb{E}\left[X_i\right]$. Então para todo $\alpha \in [0,1]$,*

- *(Forma Aditiva)*

$$\mathbb{P}\left(m^{-1} \sum_{i=1}^{m} X_i > p + \alpha\right) < e^{-2\alpha^2 m}\text{; e}$$
$$\mathbb{P}\left(m^{-1} \sum_{i=1}^{m} X_i < p - \alpha\right) < e^{-2\alpha^2 m}$$

- *(Forma Multiplicativa)*

$$\mathbb{P}\left(m^{-1} \sum_{i=1}^{m} X_i > (1+\alpha)p\right) < e^{-\alpha^2 pm/3}\text{; e}$$
$$\mathbb{P}\left(m^{-1} \sum_{i=1}^{m} X_i < (1-\alpha)p\right) < e^{-\alpha^2 pm/2}$$