IME - Instituto de
Matemática e Estatística

Universidade De São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

MAC0499 Trabalho de formatura Supervisionado

Probabilidade e Grafos

Israel Danilo Lacerra
israeldl@gmail.com

São Paulo
1 de dezembro de 2008

Universidade De São Paulo
Instituto de Matemática e Estatística
Departamento de Ciência da Computação

Israel Danilo Lacerra
israeldl@gmail.com

Probabilidade e Grafos

MAC0499 Trabalho de formatura Supervisionado do Departamento de Ciência da Computação da Universidade De São Paulo para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: *José Coelho de Pina Júnior*

São Paulo
1 de dezembro de 2008

Resumo

Nesse trabalho abordaremos alguns aspectos relacionados com probabilidade e grafos. Introduziremos alguns conceitos básicos de probabilidade e mostraremos como ela tem papel importantíssimo em Ciência da Computação, em especial em teoria dos grafos. Mostraremos principalmente duas ferramentas: o **Método Probabilístico**, que é uma ferramenta que usa conceitos probabilísticos para provar existência de certos objetos e configurações, e o **Método de Monte Carlo**, que é uma ferramenta para estimar, através de simulações, a solução de problemas que são considerados difíceis.

Sumário

CAPÍTULO 1

Introdução

Probabilidade tem um papel importante em vários algoritmos[4]. Alguns algoritmos determinísticos têm sua eficiência comprovada probabilisticamente, como o *quick sort*. Há ainda algoritmos que fazem escolhas aleatórias durante sua execução, como o algoritmo usado para evitar colisão no protocolo *Ethernet*[1], são estes os chamados **algoritmos probabilísticos**. Embora possa parecer que usar um algoritmo que faz escolhas aleatórias não seja uma boa idéia, tais algoritmos são muito usados devido a sua eficiência e simplicidade.

Em teoria dos grafos não é diferente, várias ferramentas importantes se baseiam em probabilidade. O objetivo desse trabalho é justamente fazer uma pequena introdução a essas ferramentas.

Começaremos introduzindo alguns conceitos e definições de probabilidade e, depois, nos capítulos seguintes, mostraremos duas ferramentas muito usadas em teoria dos grafos.

1.1 Introdução à Probabilidade

Para entendermos com maior clareza os algoritmos e análises vistos adiante, veremos antes, com o auxílio de um exemplo de aplicação, uma introdução aos principais conceitos de probabilidade usados nesse trabalho. Alguns outros conceitos não serão vistos agora, mas serão introduzidos no decorrer do trabalho.

1.1.1 Exemplo: Corte mínimo

Considere o problema do corte mínimo em um grafo com n vértices. Um corte em um grafo é um conjunto de arestas que, se removidas, dividem o grafo em dois ou mais componentes. Um corte é **mínimo** se tem o menor número de arestas dentre todos os cortes do grafo. O **problema do corte mínimo** consiste em encontrar um corte mínimo de um dado grafo.

No grafo da figura 1.1, as arestas $(2,3)$, $(5,3)$ e $(5,4)$ formam um corte que divide o grafo em $\{1,5,2\}$ e $\{4,3\}$. Podemos ainda obter um corte de tamanho menor. As arestas $(2,3)$ e $(1,5)$ formam um corte mínimo, que divide o grafo em $\{1,2\}$ e $\{3,4,5\}$. Observe que o

[1] http://nostalgia.wikipedia.org/wiki/Ethernet

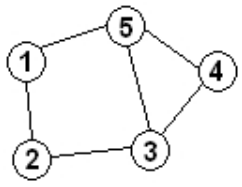

Figura 1.1 Cortes em um grafo

corte mínimo não é necessariamente único. Nesse grafo, por exemplo, temos mais de um corte mínimo. O corte formado pelas arestas $(5,4)$ e $(3,4)$ também é mínimo.

Podemos tentar encontrar o corte mínimo de um grafo usando um algoritmo probabilístico cujo consumo de tempo esperado é linear. A idéia básica é sortearmos a cada iteração uma aresta, que será contraída. Ao contrair uma aresta (u,v), removemos todas as arestas entre u e v, e transformamos u e v em um único vértice. Como a cada iteração diminuímos o tamanho do grafo em um vértice, teremos 2 vértice após $n-2$ iterações. As arestas entre esses vértices constituem um corte não necessariamente mínimo do grafo.

Vejamos dois exemplos de execução do algoritmo para o mesmo grafo. No primeiro exemplo (mostrado na figura 1.2) o algoritmo não conseguiu encontrar um corte mínimo. No primeiro passo ocorre a contração da aresta $(2,5)$ fazendo com que os vértices 2 e 5 se tranformem no vértice (ou conjunto de vértices) B. Em seguida ocorre a contração da aresta $(B,4)$ e o vértice 4 é incorporado ao vértice B. Finalmente o algoritmo contrai a aresta $(1,3)$ e obtem um corte de tamanho 3.

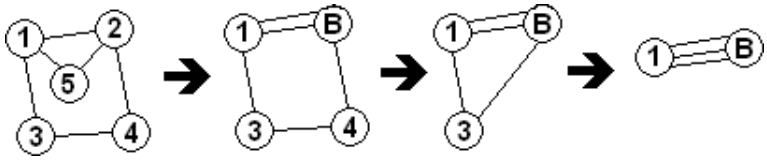

Figura 1.2 Algoritmo encontra um corte de tamanho 3

Já no segundo exemplo (figura 1.3), o algoritmo consegue encontrar um corte mínimo.

Analisaremos agora qual a probabilidade desse corte obtido com o algoritmo ser mínimo. Para analisarmos a probabilidade de determinada saída em um processo aleatório, devemos antes definir o seu **espaço amostral(chamaremos de Ω)** , que é o conjunto de todas as saídas possíveis para o processo. Além disso precisamos saber o que é uma função de probabilidade:

Definição 1.1 (Função de probabilidade)**.** Uma função de probabilidade é uma função $\Pr : F \rightarrow \mathbb{R}$ onde F é o conjunto dos eventos possíveis no espaço amostral, e:

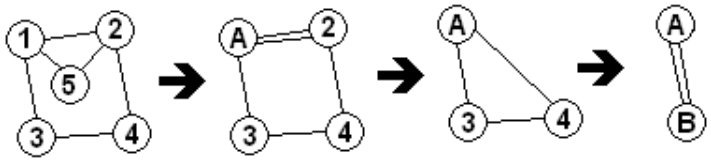

Figura 1.3 Algoritmo encontra um corte mínimo

- para qualquer evento E, $0 \leq \Pr(E) \leq 1$
- $\Pr(\Omega) = 1$
- para qualquer sequência finita ou infinita enumerável de eventos mutuamente disjuntos $E_1, E_2, E_3 \ldots : \Pr(\bigcup_i E_i) = \sum_i (\Pr(E_i))$

Seja C um corte mínimo do grafo, que divide o grafo em V e $V - S$ e k o tamanho desse corte. Se em todas as iterações forem sorteadas arestas em V ou em $V - S$, ou seja, arestas que não estejam em C, ao final da execução teremos um corte mínimo. Em nosso exemplo analisaremos uma iteração de nosso algoritmo e nosso espaço amostral será o conjunto de todas as arestas existentes no grafo no momento dessa iteração. Seja E_i o evento onde a aresta escolhida na i-ésima iteração não esteja em C. Queremos calcular $\Pr(\bigcap_i (E_i))$.

Começamos então calculando $\Pr(E_1)$. Como o tamanho do corte mínimo é k, então cada vértice possui no mínimo k arestas ligadas a ele e, portanto, o grafo possui no mínimo $nk/2$ arestas. Então a $\Pr(E_1)$, ou seja, a probabilidade de escolhermos uma aresta que não está em C na primeira iteração é:

$$\Pr(E_1) = (|G| - |C|)/|G| \geq ((nk/2) - k)/(nk/2) = 1 - 2/n$$

Aqui, podemos perceber intuitivamente que quanto maior a diferença $|G| - |C|$, a probabilidade do algoritmo retornar um corte mínimo aumenta. Ou seja, o algoritmo é melhor para casos onde o grafo possui muitas arestas e seu corte mínimo é relativamente pequeno.

Para continuarmos os nossos cálculos, devemos introduzir o conceito de probabilidade condicional.

Definição 1.2. A probabilidade de um evento A ocorrer dado que um evento B ocorreu, representada por $\Pr(A|B)$ é:

$$\Pr(A|B) = \Pr(A \bigcap B)/\Pr(B)$$

Seja F_i o evento de nenhuma aresta de C ser escolhida nas primeiras i iterações. Nesse caso $\Pr(E_1) = \Pr(F_1)$ e então:

$$\Pr(E_2|F_1) \geq \frac{(n-1)k/2 - k}{((n-1)k)/2} = 1 - 2/(n-1)$$

Para entendermos melhor essa igualdade é só pensarmos que dado que F_1 ocorreu, temos 1 vértice a menos no grafo e temos no mínimo k arestas por vértice.

Podemos facilmente generalizar para:

$$\Pr(E_i|F_{i-1}) = 1 - 2/(n-i+1)$$

O que queríamos inicialmente é calcular $\Pr(\bigcap_i^{n-2} E_i)$, ou seja, $\Pr(F_{n-2})$ que é igual a $\Pr(E_{n-2} \bigcap F_{n-3})$.

Então:

$$\begin{aligned}
\Pr(F_{n-2}) &= \Pr(E_{n-2} \bigcap F_{n-3}) \\
&= \Pr(E_{n-2}|F_{n-3})\Pr(F_{n-3}) \\
&= \Pr(E_{n-2}|F_{n-3})\Pr(E_{n-3}|F_{n-4})\Pr(F_{n-4}) \\
&= \Pr(E_{n-2}|F_{n-3})\Pr(E_{n-3}|F_{n-4})\Pr(E_{n-4}|F_{n-5})\ldots\Pr(E_2|F_1)\Pr(F_1) \\
&\geq (1-(2/n-(n-2)+1) \times (1-(2/n-(n-3)+1) \times \ldots \times (1-2/n) \\
&= 1/3 \times 2/4 \times 3/5 \times 6/4 \times 5/7 \times \ldots \times (n-3)/(n-1) \times (n-2)/n \\
&= 2/(n(n-1))
\end{aligned}$$

Portanto o algoritmo devolve um corte mínimo com probabilidade maior ou igual a $2/(n(n-1))$.

Podemos executar o algoritmo um número x de vezes, o que certamente aumentaria nossa chance de sucesso. Nesse caso, após as x execuções, devolveríamos o menor corte encontrado nas execuções. Nesse caso, as execuções do algoritmo seriam eventos independentes.

Definição 1.3 (Eventos independentes)**.** Dois eventos A e B são independentes se e somente se $\Pr(A|B) = \Pr(A) \times \Pr(B)$.

Como a probabilidade de retornar um corte que não é mínimo é menor que $1-(\frac{2}{n(n-1)})$, se executassemos o algoritmo x vezes, nossa chance de erro diminuiria para menos de $\left(1-(\frac{2}{n(n-1)})\right)^x$.

CAPÍTULO 2

O método probabilístico

O **método probabilístico** [6, 3, 5] é usado essencialmente em teoria de grafos como uma ferramenta para provar a existência de determinada configuração em um grafo, ou para provar a existência de um grafo com certas especificidades. A idéia básica é definir um espaço amostral com todas as configurações possíveis e mostrar que a probabilidade de selecionar aleatoriamente a configuração desejada é maior do que zero e, portanto, tal grafo deve existir.

2.1 O teorema de Erdős

Paul Erdős foi um dos primeiros, senão o primeiro, a usar o método probabilístico em suas provas[2]. Em 1947 ele mostrou, usando o método probabilístico, que se $\binom{n}{k}^{-\binom{k}{2}+1} < 1$ então é possível bicolorir um grafo completo com n vértices de forma que nenhum clique de tamanho menor ou igual a k seja monocromático. Um **grafo completo** é um grafo onde todos os pares de vértices estão conectados por uma aresta, isto é, um grafo com n vértices e todas as $\binom{n}{2}$ arestas. Um **clique** é um subgrafo completo.

Teorema 2.1 (Erdős (1947) [2])**.** *Se num grafo completo com n vértices temos $\binom{n}{k}2^{-\binom{k}{2}+1} < 1$, então podemos colorir usando apenas duas cores as arestas do grafo de modo que nenhum clique de tamanho k (com k vértices) seja monocromático.*

Demonstração. Queremos provar que existe uma coloração que faça com que o teorema seja válido. Para começarmos a analisar o problema precisamos definir, como no problema anterior, o espaço amostral Ω. Nesse caso, o espaço amostral será o conjunto de todas as bicolorações possíveis do grafo. Temos $\binom{n}{2}$ arestas e cada aresta pode ser colorida com uma das duas cores com probabilidade $1/2$, então temos $2^{\binom{n}{2}}$ colorações possíveis.

Em uma coloração aleatória do grafo, as escolhas das cores de cada aresta são independentes, isto é, a escolha da cor de uma aresta não exerce qualquer influência na escolha da cor das próximas arestas. Ou seja, esses eventos são independentes.

Temos $\binom{n}{k}$ cliques de tamanho k. Seja K um desses cliques e E_K o evento onde esse clique é monocromático. Se a primeira aresta desse clique foi colorida com uma das duas cores, então, para que o clique seja monocromático, todas as $\binom{k}{2} - 1$ arestas restantes devem ter essa mesma cor. Como as escolhas das cores são eventos independentes e temos que colorir $\binom{k}{2} - 1$ com

uma mesma cor:

$$\Pr(E_K) = (1/2)^{\binom{k}{2}-1} = 2^{-(\binom{k}{2}-1)} = 2^{-\binom{k}{2}+1}$$

Seja $K_1, K_2, K_3 \ldots K_{\binom{n}{k}}$ uma ordenação qualquer dos cliques de tamanho k do grafo, E_{K_i} o evento onde o i-ésimo clique é monocromático e A o evento onde o grafo possui ao menos um clique de tamanho k monocromático, temos então:

$$\Pr(A) = \Pr(\bigcup_{i=1}^{\binom{n}{k}} E_{K_i})$$

Lema 2.1. *Para quaisquer dois eventos C e D temos:*

$$\Pr(C \bigcup D) = \Pr(C) + \Pr(D) - \Pr(C \bigcap D)$$

Não provaremos esse lema, mas o mesmo é bem intuitivo se pensarmos em conjuntos. Calcularmos efetivamente $\Pr(A)$ não seria tão fácil, pois envolve uma união entre diversos eventos. Mas podemos facilmente conseguir um limitante superior, se observarmos na igualdade do lema, que $\Pr(C \bigcup D) \leq \Pr(C) + \Pr(D)$ já que $\Pr(C \bigcap D) \geq 0$. Portanto:

$$\Pr(A) = \Pr(\bigcup_{i=1}^{\binom{n}{k}} E_{K_i}) \leq \sum_{i=1}^{\binom{n}{k}} \Pr(E_{K_i}) = \binom{n}{k} 2^{-\binom{k}{2}+1} < 1$$

Mas se $\Pr(A) < 1$, então $\Pr(\overline{A}) > 0$ e, portanto, temos uma bicoloração que faz com que nenhum clique de tamanho k seja monocromático.

□

2.2 Exemplo: Conjunto de vértices independentes

Um **conjunto de vértices independentes** é um conjunto onde, para quaisquer vértices u e v pertencentes ao conjunto, não existe nenhuma aresta (u, v). Provaremos a seguir, usando uma abordagem que tem como base o método probabilístico, um teorema que limita o tamanho do maior conjunto de vértices independentes de um grafo.

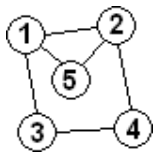

Figura 2.1 Os conjuntos {1, 4}, {4, 5}, {3, 2} e {3, 5} são independentes.

Teorema 2.2. *Todo grafo G com n vértices e m arestas possui um conjunto de vértices independentes de tamanho maior ou igual a* $n^2/4m$.

Demonstração. Considere o seguinte algoritmo probabilístico que encontra um conjunto de vértices independentes:
Passo 1. Cada vértice é removido, junto com todas as suas arestas, com probabilidade $1-\frac{1}{2m/n}$.
Passo 2. Para cada aresta restante, remove-se um dos vértices da aresta e a própria aresta.
Passo 3. Retorna os vértices restantes.

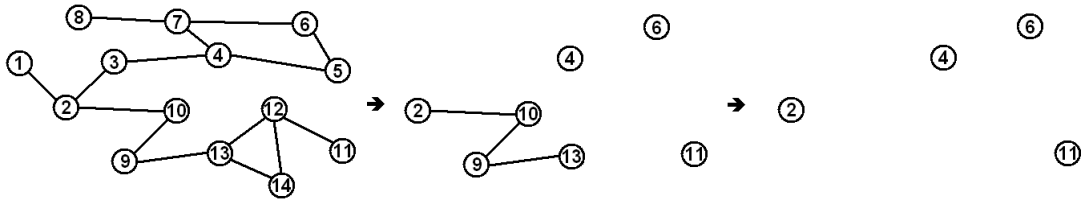

Figura 2.2 Exemplo de execução do algoritmo

No exemplo da figura 2.2, os vértices 1, 3, 5, 7, 8, 12 e 14 foram removidos no primeiro passo do algoritmo. Repare que, como o segundo passo também não é determinístico, diferentes saídas poderiam ocorrer também nesse passo. Abaixo analisaremos probabilisticamente o algoritmo.

Seja V o número de vértices e A o número de arestas que sobreviveram ao primeiro passo. Repare que a probabilidade de um dado vértice sobreviver é $\frac{1}{2m/n}$ e uma aresta só permanece no grafo se nenhum de seus vértices foi removido.

$$E[V] = n \times \frac{1}{2m/n} = n^2/2m$$

$$E[A] = m \times \frac{1}{(2m/n)^2} = n^2/4m$$

O segundo passo irá remover todas as A arestas e no máximo A vértices. Portanto, ao final do segundo passo, teremos pelo menos $V-A$ arestas. E então:

$$E[V-A] = E[V] - E[A] = n^2/2m - n^2/4m = n^2/4m$$

Mas para qualquer variável aleatória X em um espaço de probabilidades, temos que se $E[X] = x$ então então $\Pr(X \geq x) > 0$. Imagine um espaço amostral de idades de pessoas. Se a esperança, ou seja, a média das idades é 15, com certeza alguém tem 15 anos ou mais. Usando esse fato e a esperança obtida com relação ao algoritmo acima, podemos concluir então que qualquer grafo com n vértices e m arestas possui um conjunto de vértices independentes de tamanho maior ou igual a $n^2/4m$.

□

2.3 O lema local de Lovász

Mostraremos agora uma das mais importantes abordagens do método probabilístico. Muitas vezes queremos provar a existência de um objeto que tenha uma série de características. Isso pode ser fácil quando essas características podem ser traduzidas em eventos mutualmente independentes. Mas nem sempre isso acontece. O lema local de Lovász oferece uma maneira de abordarmos esses problemas mesmo quando os eventos têm um certo **grau de dependência**.

Definição 2.1. Seja G um grafo onde cada vértice é um evento E_i do espaço de probabilidades Ω, e dois eventos tem vértices adjacentes se possuem dependência entre si. O grau de dependência de E_i é o grau de seu vértice, isto é, o número de arestas que saem de E_i.

Teorema 2.3 (Lema local de Lovász)**.** *Sejam $E_1 \dots E_n$ eventos do espaço de probabilidades Ω, e:*

1. $\Pr(E_i) \leq p, \forall i$
2. *Nenhum evento possui grau de dependência maior que d, tal que $d > 0$*
3. $4dp \leq 1$

Então:

$$\Pr(\bigcap_{i=0}^{n} \bar{E_i}) > 0$$

Demonstração. Seja $S \subset \{0,1,2 \dots n\}$, começaremos provando por indução em $s = 0,1 \dots n$ que se $|S| < s$, então para quaisquer $k \notin S$ temos:

$$\Pr(E_k | \bigcap_{i \in S} \bar{E_i}) \leq 2p$$

Para essa fórmula ser consistente, devemos ter $\Pr(\bigcap_{i \in S} \bar{E_i}) > 0$, pois não faria sentido uma probabilidade condicional, onde o evento que já ocorreu tem probabilidade zero de ocorrer. Para $s = 1$ temos:

$$\Pr(\bigcap_{i \in S} \bar{E_i}) = \Pr(\bar{E_i}) = 1 - \Pr(E_i) \geq 1 - p > 0$$

Para $s > 1$ suponha, sem perda de generalidade, $S = \{1,2,3 \dots s\}$. Então:

$$\Pr(\bigcap_{i=1}^{s} \bar{E_i}) = \prod_{i=1}^{s} \Pr(\bar{E_i} | \bigcap_{j=1}^{i-1} \bar{E_j}) = \prod_{i=1}^{s} (1 - \Pr(E_i | \bigcap_{j=1}^{i-1} \bar{E_j}) \geq \prod_{i=1}^{s} (1-2p) > 0$$

Voltemos então à indução. O passo base, quando $s = 0$, segue direto da definição do teorema. Pois, nesse caso:

$$\Pr(E_k|\bigcap_{i\in S}\bar{E_i}) = \Pr(E_k) < p < 2p$$

Seja S_1 o conjunto dos eventos em S que possuem relação de dependência com E_k, e seja S_2 o conjunto dos eventos em S que não têm relação de dependência com E_k, ou seja, $S_2 = S - S_1$. Se $S = S_2$ então E_k é independente de todos os eventos em S e, portanto, temos:

$$\Pr(E_k|\bigcap_{i\in S}\bar{E_i}) = \Pr(E_k) < p < 2p$$

Vamos trabalhar com a outra hipótese, isto é, com a hipótese $|S_1| > 0$. Nesse caso temos:

$$\Pr(E_k|\bigcap_{i\in S}\bar{E_i}) = \frac{\Pr(E_k \bigcap(\bigcap_{i\in S}\bar{E_i}))}{\Pr(\bigcap_{i\in S}E_i)}$$

Seja $F_A = \bigcap_{i\in A}\bar{E_i}$. Observe que como F_S é a intersecção de todos os eventos em S e $S = S_1 + S_2$, então $F_S = F_{S_1} \bigcap F_{S_2}$. Portanto temos:

$$\begin{aligned}
\Pr(E_k|F_S) &= \frac{\Pr(E_k \bigcap F_S)}{F_S} \\
&= \frac{\Pr(E_k \bigcap F_{S_1} \bigcap F_{S_2})}{\Pr(F_{S_1} \bigcap F_{S_2})} \\
&= \frac{\Pr(E_k \bigcap F_{S_1}|F_{S_2})\Pr(F_{S_2})}{\Pr(F_{S_1}|F_{S_2})\Pr(F_{S_2})} \\
&= \frac{\Pr(E_k \bigcap F_{S_1}|F_{S_2})}{\Pr(F_{S_1}|F_{S_2})}
\end{aligned}$$

Agora usaremos um lema bem intuitivo:

Lema 2.2. *Para quaisquer eventos A e B, temos:*

$$\Pr(A \bigcap B) \leq \Pr(B)$$

Usando o lema, e lembrando que S_2 possui os eventos que são independentes de E_k, podemos limitar o numerador da fração:

$$\Pr(E_k \bigcap F_{S_1}|F_{S_2}) \leq \Pr(E_k|F_{S_2}) = \Pr(E_k) \leq p$$

Agora, para prosseguirmos, precisaremos entender e usar mais um lema.

Lema 2.3. *Para quaisquer eventos $E_1, E_2 \dots E_n$, temos:*

$$\Pr(\bigcap_{i=1}^{n} \bar{E_i}) \geq 1 - \sum_{i=1}^{n} \Pr(E_i)$$

Para entendermos o lema, vamos pensar em um exemplo simples, com apenas dois eventos A e B. Observe o espaço de probabilidades no diagrama da figura 2.3. Repare que $\Pr(\bar{A} \bigcap \bar{B})$ corresponde a toda área cinza do diagrama, ou seja:

$$\Pr(\bar{A} \bigcap \bar{B}) = 1 - (a + b + c)$$

Já para $1 - (\Pr(A) + \Pr(B))$, teríamos uma área menor:

$$1 - (\Pr(A) + \Pr(B)) = 1 - ((a + b) + (b + c)) = 1 - (a + 2b + c)$$

Portanto, temos:

$$\Pr(\bar{A} \bigcap \bar{B}) = 1 - (a + b + c) \geq 1 - (a + 2b + c) = 1 - (\Pr(A) + \Pr(B))$$

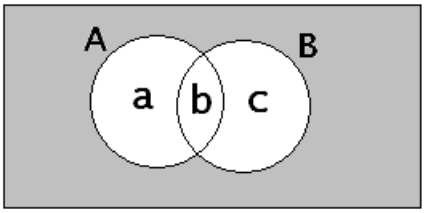

Figura 2.3

Voltemos agora à nossa indução. Podemos agora limitar o denominador da fração:

$$\Pr(F_{S_1} | F_{S_2}) = \Pr(\bigcap_{i \in S_1} \bar{E_i} | \bigcap_{j \in S_2} \bar{E_j}) \geq 1 - \sum_{i \in S_1} \Pr(E_i | \bigcap_{j \in S_2} \bar{E_j})$$

Agora, ainda no denominador, observando que $|S_2| < |S|$, podemos aplicar a hipótese de indução e obter:

$$\Pr(F_{S_1} | F_{S_2}) \geq 1 - \sum_{i \in S_1} \Pr(E_i | \bigcap_{j \in S_2} \bar{E_j}) \geq 1 - \sum_{i \in S_1} 2p$$

Mas sabemos que S_1 é o conjunto de eventos que têm relação de dependência com E_k e, portanto, $|S_1| \leq d$ já que d é o grau de dependência máximo que E_k pode ter. Portanto:

$$\Pr(F_{S_1}|F_{S_2}) \geq 1 - \sum_{i \in S_1} 2p \geq 1 - 2pd \geq 1/2$$

Substituindo agora o numerador e o denominador, conseguimos provar a indução:

$$\Pr(E_k|\bigcap_{i \in S} \bar{E_i}) = \Pr(E_k|F_S) = \frac{\Pr(E_k \bigcap F_{S_1}|F_{S_2})}{\Pr(F_{S_1}|F_{S_2})} \leq \frac{p}{1/2} = 2p$$

Agora, com esse resultado em mãos, podemos finalmente provar o lema local de Lovász:

$$\Pr(\bigcap_{i=1}^{n} \bar{E_i}) = \prod_{i=1}^{n} \Pr(\bar{E_i}|\bigcap_{j=1}^{i-1} \bar{E_j}) = \prod_{i=1}^{n} (1 - \Pr(E_i|\bigcap_{j=1}^{i} -1\bar{E_j}))$$

$$\begin{aligned}
\Pr(\bigcap_{i=1}^{n} \bar{E_i}) &= \prod_{i=1}^{n} \Pr(\bar{E_i}|\bigcap_{j=1}^{i-1} \bar{E_j}) \\
&= \prod_{i=1}^{n} (1 - \Pr(E_i|\bigcap_{j=1}^{i} -1\bar{E_j})) \\
&\geq \prod_{i=1}^{n} (1-2p) > 0
\end{aligned}$$

A última linha segue dos fatos do enunciado do teorema, pois como $d > 0$ e $4dp \leq 1$ temos $2p < 1$.

□

O lema local de Lovász é usado como ferramenta para solucionar uma série de problemas em ciência da computação. Mostraremos agora um exemplo de sua aplicação.

2.3.1 Aplicação: Caminhos disjuntos

Caminho disjuntos em um grafo, são caminhos que não compartilham nenhuma aresta. Suponha que n pares de nós em uma rede representada por um grafo precisam se comunicar por caminhos disjuntos. Podemos provar, usando o lema local de Lovász, que se os caminhos possíveis não compartilham muitas arestas, então podemos encontrar n caminhos disjuntos para os n pares.

Teorema 2.4. *Sejam* $1,2,3 \ldots n$ *uma ordenação qualquer dos n pares da rede que precisam se comunicar e seja* F_i *o conjunto dos caminhos possíveis entre os dois nós do par i, tal que* $|F_i| = m$*. Se qualquer caminho em* F_i *não compartilha aresta com mais que k caminhos em* F_j*, tal que* $i \neq j$*, e* $8nk/m \leq 1$*, então podemos encontrar n caminhos disjuntos para os n pares.*

Demonstração. Nosso espaço de probabilidades Ω, será o conjunto de todas as combinações possíveis de caminhos, sejam eles disjuntos ou não, entre todos os pares. Como temos n pares e para cada par temos m caminhos possíveis, $|\Omega| = nm$. Seja $E_{i,j}$ o evento onde os caminhos entre os pares i e j não são disjuntos. Suponha sem perda de generalidade que entre os pares i e j, escolhemos primeiro o caminho de i. Como um caminho em F_i não compartilha arestas com mais que k arestas em F_j, teremos em F_j no máximo k caminhos que se sorteados causariam o evento $E_{i,j}$. Então:

$$\Pr(E_{i,j}) \leq k/m$$

Já temos então um limitante que cumpre o primeiro requisito do lema local de Lovász. Precisamos agora limitar o grau de dependência d entre os eventos. O evento $E_{i,j}$ depende das escolhas dos caminhos entre os pares i e j, e então $E_{i,j}$ tem relação de dependência com qualquer outro evento $E_{u,v}$ tal que u e/ou $v \in \{i, j\}$, ou seja, qualquer outro evento que tenha relação com a escolha dos caminhos entre os pares i e/ou j. Então:

$$d < 2n$$

Agora só precisamos verificar se esses dois limitantes tornam válido também o terceiro requisito do lema local de Lovász:

$$4d(k/m) < 8nk/m \leq 1$$

Portanto, o lema local de Lovász vale para esse caso, e então:

$$\Pr(\bigcap_{i \neq j} \overline{E_{i,j}}) > 0$$

Ou seja, existe uma combinação de n caminhos em Ω tal que todos os caminhos são disjuntos.

□

CAPÍTULO 3

O método de Monte Carlo

O método de **Monte Carlo**[3] consiste em estimar algum valor usando um espaço amostral e uma simulação nesse espaço. Se quisermos, por exemplo, estimar a quantidade de vértices de cor azul em um grafo bicolorido, podemos sortear um número suficientemente grande de vértices e usar a proporção obtida de vértices de cor azul. A primeira vista isso pode parecer um modo inocente de estimar valores, mas podemos obter algoritmos muito robustos usando o método de Monte Carlo.

Nesse capítulo apresentaremos um exemplo de aplicação do método de Monte Carlo e, durante a apresentação desse exemplo, introduziremos os principais conceitos do método de Monte Carlo.

3.1 Exemplo: Conjuntos de vértices independentes

Um dos problemas em teoria de grafos, consiste em encontrar o número de conjuntos de vértices independentes que existem em um grafo. Encontrar esse número é um problema do tipo #*P*-completo (pronunciado *sharp-p-completo*)[1]. Estamos acostumados a ouvir sobre os problemas do tipo *NP-completo*, que são problemas de decisão para os quais não conhecemos nenhum algoritmo com consumo de tempo polinomial que os resolva. A classe de complexidade #*P*-completo é similar à classe *NP*-completo, mas se refere a problemas de contagem e não a problemas de decisão. Portanto, um problema do tipo #*P*-completo, é um problema de contagem para o qual não se conhece nenhum algoritmo de consumo de tempo polinomial.

Encontrar o número de conjuntos de vértices independentes é, portanto, um problema difícil. Mas podemos solucionar, pelo menos de maneira aproximada, esse problema usando o Método de Monte Carlo.

Seja $e_1, e_2, \ldots e_m$ uma ordenação das arestas do grafo G e seja G_i o grafo formado só com as primeiras i arestas. Seja $\Omega(G)$ o espaço formado por todos os conjuntos independentes de G. Queremos saber $|\Omega(G)|$. Podemos expressar da seguinte forma:

$$|\Omega(G)| = |\Omega(G_m)| = \frac{|\Omega(G_m)|}{|\Omega(G_{m-1})|} \times \frac{|\Omega(G_m - 1)|}{|\Omega(G_{m-2})|} \times \ldots \times \frac{|\Omega(G_1)|}{|\Omega(G_0)|} \times |\Omega(G_0)|$$

Notem que G_0 é um grafo sem arestas e então, qualquer subconjunto de vértices é inde-

pendente. Portanto $|\Omega(G_0)| = 2^n$. Então temos:

$$|\Omega(G)| = 2^n \prod_{i=1}^{m} \frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}$$

Usaremos o algoritmo (que é um método de Monte Carlo) abaixo para estimar $\frac{|\Omega(G_i)|}{|\Omega(G_{i-1})|}$ e então teremos uma estimativa de $|\Omega(G)|$.

```
ESTIME (G_i, G_{i-1})
1   x ← 0
2   repita M vezes
3       C ← sorteie um conjunto de vértices independentes de G_{i-1}
4       se C é um conjunto independente de G_i
5           faça x++
6   devolva x/M
```

Para se ter sucesso em nossa estimativa, o valor de M deve ser suficientemente grande. Além disso, o sorteio do conjunto de vértices independentes de G_{i-1} merece atenção pois não pode afetar muito a eficiência do algoritmo e deve ser um sorteio de qualidade. Veremos agora uma maneira simples de fazer esse ”sorteio”.

3.2 Obtendo uma amostra (Cadeias de Markov)

Cadeia de Markov é um processo estocático discreto $X_0, X_1 \ldots X_n$ onde qualquer estado X_i depende apenas do estado X_{i-1}. Dizemos que a cadeia de Markov não tem memória, pois depende apenas do estado imediatamente anterior, e não depende de todos os estados que se passaram para se chegar ao ponto atual.

Podemos definir uma Cadeia de Markov para fazer o sorteio do algoritmo anterior, da seguinte forma:

1. X_0 é um conjunto de vértices independentes qualquer.
2. Para calcular X_{i+1}:
   - Sorteamos um vértice v do grafo.
   - Se $v \in X_i$, então $X_{i+1} = X_i \setminus v$
   - Se $v \notin X_i$ e $Y = X_i \bigcup \{v\}$ é um conjunto de vértices independentes, então $X_{i+1} = Y$
   - Senão $X_{i+1} = X_i$

CAPÍTULO 4

Parte Subjetiva

4.1 Dificuldades encontradas

A decisão sobre meu TCC ocorreu em meados do segundo semestre do ano passado. Nessa época decidi falar com o professor Coelho sobre a possibilidade dele orientar meu trabalho em algum tema relacionado com grafos, que é uma das áreas que mais gostei de ter contato em minha graduação. O Coelho me falou sobre alguns temas que ele tinha em mente, e um desses temas era o que mais o interessava para esse trabalho: algoritmos probabilísticos. Nesse ponto já estava com o tema praticamente decidido, e podia fazer o que eu tinha em mente: começar o tcc alguns meses antes. É ai que ficou aparente minha principal dificuldade: a disciplina. Com o semestre bem puxado e minha falta de organização, acabei não começando o TCC com a antecedência que pretendia. No primeiro semestre desse ano a situação continuou parecida, e minhas atividades se resumiram a leitura de algumas partes do livro indicado pelo Coelho[3], e com uma certa falta de base teórica que eu tinha em probabilidade e estatística, a leitura se tornava algo muito cansativo. No segundo semestre as coisas engrenaram. Começamos a fazer reuniões semanais de 2 a 3 horas, onde discutimos, eu, o professor Coelho, e dois colegas do BCC, o Gilson Dias e o Andre Jucovsky Bianchi. Nas discussões fomos nos aprofundando no livro que seguíamos[3] e consegui uma boa base teórica para começar a escrever minha monografia.

Acredito que foram essas minhas principais dificuldades: a organização do tempo, e a falta de base em estatística e probabilidade (essa que também é culpa minha).

4.2 Disciplinas do curso

A principal disciplina que utilizei em meu TCC foi MAC0328 - Algoritmos em Grafos, que foi inclusive a disciplina que me fez escolher o tema do TCC. Essa disciplina, que foi muito bem ministrada pelo professor Yoshi, me deu uma excelente base em teoria de grafos. Mas é claro que disciplinas como MAC122, MAC323, MAC338 também ajudaram e muito. Mas sobre as disciplinas, um ponto não posso deixar de comentar. As matérias MAEXXXX não contribuiram como eu acho que deviam contribuir tanto no meu TCC quanto na minha graduação. Acredito a principal culpa seja minha, mas acho que talvez falte uma matéria de MAE mais voltada ao BCC. Não fiz as matérias de MAE com o BCC, pois na época ainda cursava o BMAC mas, mesmo assim, percebo esse problema na maioria dos meus colegas de curso.

Acho que uma matéria que usasse os conceitos de estatística e probabilidade mais intensamente em algoritmos e em teoria da computação nos ajudaria a aprender e a ter mais interesse nessa área. Acho que uma disciplina que, por exemplo, ensinasse gradualmente o conteúdo de MAE0212 já aplicando os conceitos em teoria da computação, gerariam alunos com mais domínio na área de probabilidade. Acho que poderíamos ter uma disciplina obrigatória que poderia ser nos moldes de *CS 223 – Random Processes and Algorithms*[1], que é ministrada pelo próprio Mitzenmacher, autor do livro que estudei no TCC.

4.3 Conclusões e atividades futuras

Aprendi muita coisa interessante no meu TCC. Como comentei na seção anterior, minha base de probabilidade e estatística não era muito boa, e com o trabalho, acabei adquirindo uma base razoável na área. Acabei também me surpreendendo com o quanto gostei de estudar sobre *Métodos Probabilísticos* e acredito que se for continuar estudando sobre probabilidade e grafos, estudaria mais sobre esse assunto. Achei diversos papers que mostram soluções interessantes, onde podemos transformar uma prova feita com o Método Probabilístico, mais precisamente usando o lema local de Lovász, em um algoritmo eficiente. Acho que seria muito interessante se aprofundar mais nesse tópico, talvez em um mestrado. Achei também alguns artigos que mostram soluções do problema da árvore geradora mínima usando o Método de Monte Carlo, que também pode ser alvo de estudo.

[1] http://www.eecs.harvard.edu/ michaelm/CS223/syllabus.html

Referências Bibliográficas

[1] Martin Dyer and Catherine GreenhillMotwani. On markov chains for independent sets. *Journal of Algorithms*, 2000.

[2] P. Erdős. Some remarks on the theory of graphs. *Bulletin of the Amer. Math. Soc. 53:292-294*, 1947.

[3] Michael Mitzenmacher and Eli Upfal. *Probability and Computing - Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, 2005.

[4] Rejeev Motwani and Prabhakar Raghavan. Randomized algorithms. *ACM Compunting Surveys, Vol 28, No 1*, 1996.

[5] Joel Spencer. *Ten Lectures on the Probabilistic Method*. SIAM, second edition, 1994.

[6] Joel Spencer. Modern probabilistic methods in combinatorics. *British Combinatorial Conference, Stirling*, 1995.