Algoritmos Probabilísticos

Departamento de Ciência da Computação – IME/USP

Segundo Semestre de 2006

Algoritmo de Luby

Problema: Dado um grafo $G = (V, E)$, encontrar um conjunto independente maximal.

Para cada vértice v, denotamos por $N(v)$ o conjunto dos vizinhos de v em G e por $d(v)$ o grau de v, ou seja, o número de elementos em $N(v)$. Para um conjunto S de vértices de G, denotamos por $N(S)$ o conjunto de todos os vizinhos de vértices de S.

O seguinte algoritmo para o problema acima foi proposto por Luby.

```
AlgLuby (G)
 1  I ← ∅
 2  V_0 ← V(G)
 3  i ← 0
 4  enquanto V_i ≠ ∅ faça
 5        S_i ← ∅
 6        para cada v ∈ V_i (em paralelo) faça
 7              se d(v) = 0
 8                    então S_i ← S_i ∪ {v}
 9                    senão marque v com probabilidade 1/(2d(v))
10        para cada e = uv ∈ E(G[V_i]) em paralelo faça
11              se u e v estão marcados
12                    então desmarque aquele entre u e v de menor grau
13        para cada v ∈ V_i (em paralelo) faça
14              se v está marcado
15                    então S_i ← S_i ∪ {v}
16        I ← I ∪ S_i
17        V_{i+1} ← V_i \ (S_i ∪ N(S_i))
18        i ← i + 1
19  devolva I
```

Caso $d(u) = d(v)$ na linha 12 do algoritmo, escolha arbitrariamente um entre u e v para desmarcar. Observe que I é de fato um conjunto independente maximal.

Um pouquinho sobre computação paralela

Para os que já viram a descrição de uma RAM (*random access machine*), uma PRAM (*parallel RAM*) é uma versão paralela dessa, composta por um número de processadores (cada um equivalente a uma RAM) sincronizados e que compartilham uma memória global. No modelo CRCW (*concurrent read, concurrent write*), são permitidos acessos simultâneos (leituras) a posições da memória global, bem como tentativas de escrita simultâneas a posições da memória global (apenas um dos processadores que tenta escrever em um mesmo instante em uma mesma posição da memória global efetivamente escreve nessa posição). No modelo EREW (*exclusive read, exclusive write*), não são permitidos nem leituras nem tentativas de escrita simultâneas. Existe ainda um modelo intermediário, o CREW, que permite leituras simultâneas mas não

permite tentativas de escrita simultâneas. Se você ficou interessado nesse assunto, consulte na nossa biblioteca o livro do John Reif, *Synthesis of Parallel Algorithms* (QA754.C3 R361s).

Cada iteração do algoritmo pode ser implementada facilmente em uma CRCW PRAM em tempo constante e, mais interessante, pode ser implementada também em uma EREW PRAM em tempo $O(\lg(n+m))$.

Exercício: Descreva uma implementação do algoritmo acima para uma CREW PRAM, ou seja, explique uma implementação do algoritmo que não faça tentativas de escrita simultâneas. Você pode precisar gastar mais memória para evitar as tentativas de escritas simultâneas.

De volta à análise do algoritmo de Luby

Qual é o número esperado de iterações do algoritmo de Luby?

Para responder a essa pergunta, considere $V_i = S_i = \emptyset$ sempre que esses estiverem indefinidos. Também denote por E_i o conjunto de arestas do grafo $G[V_i]$ e por X_i o número de arestas em E_i, ou seja, $X_i = |E_i|$. Seja $m = |E(G)|$. Então $X_0 = m$. Vamos mostrar o seguinte:

Lema 1. *Existe uma constante $c < 1$ tal que $\mathrm{E}[X_i] \leq c\,\mathrm{E}[X_{i-1}]$, para $i = 1, 2, \ldots$.*

Exercício: Sejam variáveis aleatórias $X_0, X_1, \ldots$ onde $X_0 = m$ e, para $i = 1, 2, \ldots$, a variável $X_i \in \{0, \ldots, X_{i-1}\}$ e é tal que $\mathrm{E}[X_i] \leq c\,\mathrm{E}[X_{i-1}]$, onde c é uma constante positiva menor que 1. Seja t o menor i tal que $X_i = 0$. Mostre que $\mathrm{E}[t] = O(\lg m)$.

Conclua do exercício acima que o algoritmo de Luby faz $O(\lg m)$ iterações.

Para demonstrar esse lema, demonstramos primeiro alguns resultados intermediários. Esses resultados usam a seguinte terminologia. Um vértice v é *bom* se pelo menos $d(v)/3$ de seus vizinhos tem grau menor ou igual a $d(v)$, caso contrário v é *ruim*. Uma aresta $e = uv$ é *boa* se pelo menos um entre u e v é bom, do contrário e é *ruim*.

Antes de demonstrar os resultados intermediários, deixe-me dar uma idéia da prova do lema, para que você perceba que esses resultados poderão nos ajudar a provar o lema. O lema diz que o número esperado de arestas do grafo $G[V_i]$ diminui de uma fração constante. A idéia da prova é mostrar que o número esperado de arestas do grafo $G[V_i]$ que são removidas a cada iteração é uma fração do número de arestas do grafo $G[V_i]$. Para tanto, vamos mostrar que uma fração grande das arestas do grafo é boa, e que o número esperado de arestas boas removidas numa iteração é uma fração do total das arestas boas. Lembre-se que uma aresta é boa se tem pelo menos um extremo bom.

Fato 1. *Pelo menos metade das arestas de um grafo são boas.*

Agora considere o grafo $G[V_i]$ e a iteração i do algoritmo de Luby.

Fato 2. *Se v é bom e $d(v) > 0$, então* $\Pr[\text{pelo menos um vizinho de } v \text{ ser marcado}] \geq 1 - e^{-1/6}$.

Fato 3. *Se v foi marcado, então* $\Pr[v \in S_i] \geq 1/2$.

Fato 4. *Se v é bom, então* $\Pr[v \in S_i \cup N(S_i)] \geq \frac{1-e^{-1/6}}{2}$.

Dos quatro fatos acima, podemos derivar a demonstração do lema. Nesta aula, vamos ver a prova do fato 4 e a demonstração do lema.