

Algoritmos Randômicos para a Geração de Orientações Acíclicas em Sistemas Distribuídos

Gladstone M. Arantes Jr, Felipe M. G. França

Engenharia de Sistemas e Computação-COPPE, Universidade Federal do Rio de Janeiro
Caixa Postal 68511, 21945-970, Rio de Janeiro, RJ, Brasil
{glads, felipe}@cos.ufrj.br

Carlos A. Martinhon

Instituto de Computação, Universidade Federal Fluminense
R. Passo da Pátria 156, Bloco E, Sala 303, São Domingos, 24210-230, Niterói, RJ, Brasil
mart@dcc.ic.uff.br

Resumo

Este artigo discute dois novos algoritmos distribuídos randômicos (ou probabilísticos) para a geração de orientações acíclicas em um sistema distribuído anônimo de topologia arbitrária. Os algoritmos são analisados tanto em termos de convergência quanto velocidade de convergência. Em particular, é demonstrado que o novo algoritmo chamado *Alg-Arestas*, é capaz de produzir orientações acíclicas quase instantaneamente, isto é, em menos de dois (2) passos. Duas aplicações para essa forma de quebra de simetria serão discutidas: inicialização do *Escalonamento por Reversão de Arestas (ERA)*, um simples e poderoso algoritmo de escalonamento distribuído, e; uma estratégia de distribuição de uploads em redes de computadores

Palavras-chave: algoritmos distribuídos randômicos, exclusão mútua, quebra de simetria, sistemas anônimos.

Abstract

This paper discusses two new randomized (or probabilistic) distributed algorithms for the generation of acyclic orientations upon anonymous distributed systems of arbitrary topology. The set of algorithms is analysed both in terms of convergency as the rate of convergency. In particular, it is shown that the new algorithm called *Alg-Arestas* is able to produce acyclic orientations *quasi* instantaneously, i.e., in less than two (2) steps. Two applications of this form of symmetry breaking will be discussed: initialization of *Scheduling by Edge Reversal (SER)*, a simple and powerful distributed scheduling algorithm, and; a strategy for distributed uploading in computer networks.

Keywords: anonymous systems, mutual exclusion, randomized distributed algorithms, symmetry breaking.

I. INTRODUÇÃO

Neste trabalho apresentaremos dois novos algoritmos probabilísticos que geram orientações acíclicas em sistemas distribuídos anônimos, i.e., sistemas onde não existam identificações globais dos seus componentes. Sendo um dos

possíveis métodos de quebra de simetria (criação de conjuntos independentes maximais e coloração são outras formas, todas equivalentes entre si em algum nível), esta é uma aplicação muito útil, com muitos trabalhos correlatos publicados, incluindo algoritmos paralelos, distribuídos e seqüenciais, além de determinísticos e probabilísticos [GOL 87] [GOL 88] [ITA 90] [LUB 86] [PAN 92] [SZE 93]. As maiores motivações para a concepção de algoritmos probabilísticos, em oposição aos algoritmos puramente determinísticos, estão na simplicidade de suas implementações e na rápida geração de soluções com características distintas. Esta diversidade torna-se particularmente interessante em aplicações baseadas em orientações acíclicas, como discutiremos adiante.

Analisamos os dois novos algoritmos probabilísticos distribuídos, *Alg-Viz* e *Alg-Arestas*, quanto à corretude e velocidade de convergência. Lidamos com algoritmos randômicos onde, a cada nova execução do algoritmo uma saída distinta é obtida. Logo, o tempo deve ser expresso como uma *variável aleatória* com *valor esperado* indicando a *velocidade de convergência* do algoritmo. Ambos os algoritmos fazem o uso de *dados*, viciados ou não, com f faces. Foi determinado que o algoritmo *Alg-Viz* possui complexidade $O(D)$, onde D é grau máximo de G (grafo alvo), e assumindo $f = D + 1$ faces. A análise do novo algoritmo chamado *Alg-Arestas* foi baseada na noção de *recorrência probabilística* [KAR 94] e resultou em complexidade $O(\lceil \log m \rceil)$, onde m é o número de arestas de G .

Na Seção II, introduzimos conceitos básicos: sistemas distribuídos anônimos e sua representação em termos de grafos. Na Seção III, apresentamos o algoritmo básico de *Calabrese/França* [CAL 94] [CAL 97] e introduzimos os dois algoritmos de geração de orientações acíclicas (*Alg-Viz* e *Alg-Arestas*), discutindo corretude e velocidades de convergência. Algumas aplicações envolvendo a

determinação de orientações acíclicas no contexto de sistemas anônimos são analisadas na Seção IV. Finalmente, nossas conclusões, incluindo futuros desenvolvimentos, são expostas na Seção V.

II. CONCEITOS BÁSICOS

A. Sistemas Anônimos

Um sistema distribuído pode ser simplificado descrito como um conjunto de nós de processamento (podendo ser muito simples ou muito complexos) organizados de tal forma que cada nó está conectado a um subconjunto dos outros por um canal de comunicação qualquer. A comunicação entre cada par de nós se dá através de envio de mensagens. Comumente, assume-se que as conexões são bidirecionais, o que significa que um nó conectado a outro tanto pode enviar mensagens a ele quanto receber. Esta é uma definição bastante genérica, podendo ser aplicada para uma enorme gama de sistemas.

Um sistema distribuído é considerado anônimo quando não existe um identificador que diferencie um nó de outro. No nosso caso, estaremos assumindo ainda que não exista nenhuma informação global acerca do sistema, ou seja, não há disponível, *a priori*, informações como o número total de nós, o número de canais ou ainda qual a topologia do sistema. Esse tipo de sistema apresenta grandes dificuldades para ser manipulado [BAR 96].

B. Notação Adotada

Todo sistema distribuído pode ser modelado como um grafo $G(V, E)$. Nesse grafo, V , de cardinalidade n , é o conjunto dos nós do sistema e E , de cardinalidade m , é o conjunto de arestas do tipo (n_i, n_j) , onde n_i e n_j são nós de V . Se $(n_i, n_j) \in E$, então existe um canal de comunicação entre n_i e n_j .

III. OS ALGORITMOS DISTRIBUÍDOS

Descrevemos e analisaremos aqui os dois algoritmos distribuídos probabilísticos de nosso interesse. Estes dois algoritmos foram desenvolvidos a partir das idéias básicas contidas em [CAL 94] e [CAL 97], traduzidas pelo algoritmo aqui chamado *Calabrese/França*. Este é apresentado logo a seguir e precede a apresentação dos algoritmos *Alg-Viz* e *Alg-Arestas*.

A. Calabrese/França

Este algoritmo utiliza um gerador de números aleatórios simples que pode gerar 0 ou 1 randomicamente e com probabilidade 1/2. Por isso, dizemos tratar-se de uma *moeda equilibrada* ou *não-polarizada*.

O algoritmo executa sincronamente. Cada nó é dito probabilístico se ainda está participando do algoritmo (nesse caso, ele ainda tem arestas incidentes não orientadas

e continua tomando parte nos sorteios) ou determinístico, no caso de não participar mais por já ter tido todas as arestas incidentes orientadas. Em cada passo do algoritmo todos os nós probabilísticos lançam uma moeda, obtendo 0 ou 1. Um nó que obtiver 1 e cujos vizinhos probabilísticos restantes tiverem obtido 0 irá orientar todas as suas arestas ainda não orientadas na sua direção. O algoritmo executa até que todas as arestas sejam orientadas.

Tal algoritmo é bastante ineficiente se a probabilidade de obter-se 0 ou 1 no sorteio for 1/2. Por isso, comumente, a moeda será polarizada (ou viciada), da seguinte forma: $Pr\{moeda_i=1\} = 1/(|N(n_i)|+1)$ e $Pr\{moeda_i=0\} = 1 - Pr\{moeda_i=1\}$, onde $|N(n_i)|$ representa a cardinalidade do conjunto de vizinhos probabilísticos $N(n_i)$ do nó n_i . Desta forma, intuitivamente, em uma dada vizinhança, apenas um dos nós tende a obter 1 no sorteio.

A.1 Velocidade de convergência

O tempo ou velocidade de convergência é a medida do número de passos que o algoritmo precisa dar para que este convirja (termine), ou seja, até que todos os nós se tornem determinísticos. É fácil ver que o algoritmo de *Calabrese/França* [CAL 94] para moedas equilibradas é bastante ineficiente. Sucintamente, pode-se mostrar neste caso que, para grafos completos, aproximadamente $O(2^n)$ passos serão necessários para orientação completa de todas as arestas. A análise do tempo de processamento utilizando grafos completos é interessante já que, além de mais simples, um grafo completo representa, na verdade, a pior instância para o problema. Se S_i é um evento indicando que $n_i \in V$ obteve 1 e os demais nós probabilísticos obtiveram 0 então, no máximo, um evento S_i ocorrerá de cada vez. Para mostrar esse fato, seja S um evento indicando a união de n eventos disjuntos S_i ($p/ 1=1, \dots, n$).

Tem-se então que:

$$Pr(S) = Pr\left(\bigcup_{i=1}^n S_i\right) = \sum_{i=1}^n Pr(S_i) = \sum_{i=1}^n \left(\frac{1}{2}\right)^{n-1} = \frac{n}{2^n}$$

Assim, $O(2^n \cdot n^{-1})$ repetições serão necessárias (valor esperado) até que algum dos eventos S_i ocorra. Observe que o número de repetições pode ser modelado por uma variável aleatória com distribuição exponencial e valor esperado $2^n \cdot n^{-1}$. Logo, a orientação completa para cada um dos n vértices de V irá consumir aproximadamente $n \cdot 2^n \cdot n^{-1} = 2^n$ passos.

A complexidade do algoritmo de *Calabrese/França* cai drasticamente quando se troca as moedas equilibradas por moedas viciadas ou polarizadas. Novamente, para grafos completos, tem-se que $Pr\{moeda_i=1\}=1/n$ (considerando-se $n-1$ nós probabilísticos no passo). Assim

$$Pr(S) = Pr\left(\bigcup_{i=1}^n S_i\right) = \sum_{i=1}^n Pr(S_i) = \sum_{i=1}^n \left(\frac{1}{n}\right)^{n-1} \geq \frac{n}{n \cdot e} = \frac{1}{e}$$

Portanto, aproximadamente e repetições serão necessárias até que algum dos eventos S_i ($p/ \in \{1, \dots, n\}$) ocorra. Logo, para todos os nós tornarem-se determinísticos deverá ocorrer um número de passos aproximadamente igual a $n.e$, sendo portanto $O(n)$. Para maiores detalhes vide [CAL 94], [CAL 97].

B. Alg-Viz

Esse algoritmo é uma extensão muito simples do anterior e propõe que o sorteio seja realizado com números em uma faixa de valores inteiros. Isso equivale à utilização de um dado de f faces em vez de uma moeda. Assim, poderemos obter como resultado do sorteio números inteiros de 0 a $f-1$.

Assim, *Alg-Viz* pode ser descrito da mesma forma que o algoritmo *Calabrese/França*, sendo necessário chamar a atenção apenas para o fato de que, agora, um nó ganha no sorteio quando tira o maior valor entre todos seus vizinhos probabilísticos.

Da mesma forma que no algoritmo *Calabrese/França*, a utilização de um gerador de números aleatórios polarizado (ou viciado) torna o algoritmo mais eficiente. Entretanto, como será visto na seção seguinte, a redução não acontece de maneira drástica como no algoritmo de *Calabrese/França*. Veremos que, em ambas as versões (com dados polarizados e não-polarizados), o tempo esperado de processamento será igual a $O(n)$.

O método de polarização utilizado é diferente para cada nó do sistema e é mostrado abaixo:

$$f \leq |N(n_i)| + 1:$$

$$\Pr\{dado_i = f-1\} = \frac{1}{|N(n_i)| + 1}$$

$$\Pr\{dado_i = a\} = \frac{1}{|N(n_i)| + 1} \left(1 - \sum_{k=a+1}^{f-1} \Pr\{dado_i = k\} \right), 0 \leq a \leq f-2$$

$$f > |N(n_i)| + 1:$$

$$\Pr\{dado_i = a\} = \frac{1}{f}, \forall a \text{ tal que } 0 \leq a \leq f-1$$

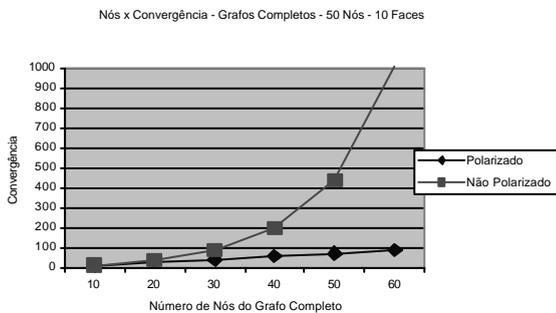


Figura 1: Alg-Viz. Dados polarizados e não-polarizados.

A Figura 1 dá uma idéia, através de resultados de simulações, da melhora ocorrida com a utilização de polarizações. As simulações apresentadas nesse trabalho são versões sequenciais dos algoritmos escritos em linguagem C. Cada ponto, de todos os gráficos produzidos, é a média aritmética obtida através da simulação de mil (1000) de grafos conexos gerados aleatoriamente.

Desta forma, mantém-se uma polarização similar àquela utilizada em *Calabrese/França* para os casos em que $f \leq |N(n_i)| + 1$ e, para os casos em que $f > |N(n_i)| + 1$, aproveita-se o fato de haver mais faces do que vizinhos para particionar melhor a probabilidade de cada face. Esta polarização foi escolhida por ter se mostrado a mais eficiente na comparação com outras alternativas.

O teorema seguinte garante convergência do *Alg-Viz*.

Teorema 1. O algoritmo *Alg-Viz* ($p/$ dados polarizados ou não) sempre gera orientações acíclicas em um grafo $G(V,E)$ qualquer.

Prova: Vamos assumir que um nó, ao ganhar a disputa de dados com os vizinhos, recebe como rótulo um número r inteiro positivo, que corresponde ao número de disputas de dados que ele participou até ser vitorioso. Sabemos que dois vizinhos não terão o mesmo rótulo, já que no passo que um ganha o outro obrigatoriamente não ganha. Além disso, dados dois nós $u, v \in V$ e o seus rótulos r_u e r_v , tais que, sem perda de generalidade, $r_u < r_v$, uma aresta $(u,v) \in E$ terá sempre a direção $v @ u$. Desta forma, o algoritmo corresponde a realizar uma coloração no grafo e orientar as arestas da maior para a menor cor o que certamente, não cria ciclos. •

Obviamente, as análises de convergência realizadas para o algoritmo *Alg-Viz* são igualmente válidas para *Calabrese/França*, já que, *Alg-Viz* apenas faz um particionamento maior na distribuição de probabilidades de *Calabrese/França*.

B.1 Velocidade de convergência

Como veremos, nesta seção, a utilização de dados em vez de moedas não modifica significativamente a complexidade da convergência a não ser que haja uma relação entre f e uma característica global do sistema. A grande vantagem de *Alg-Viz* em relação a *Calabrese/França* é diminuir (e, no caso, de dados com muitas faces, anular) o efeito dos empates entre vizinhos, incrementando, desta forma, a velocidade de convergência do algoritmo.

Mostraremos nesta seção que, para grafos completos, o tempo esperado de processamento do *Alg-Viz* para dados equilibrados com n faces será idêntico ao algoritmo *Calabrese/França* com moedas viciadas.

Seja $d_i = \infty$ p/ $\infty \in \{0,1,2,\dots,f-1\}$, o resultado associado a um nó $n_i \in V$, obtido após jogarmos um dado equilibrado com f faces. É fácil ver que neste caso que $Pr(d_i = \mathbf{a}) = 1/f$. Diremos que um evento S_i , associado ao nó $n_i \in V$ ocorre, se e somente se, $d_j < d_i, \forall n_j \in \mathbf{N}n_i$, onde $\mathbf{N}n_i$ representa o conjunto dos vizinhos probabilísticos de n_i . Temos então o seguinte resultado preliminar:

Teorema 2. Considere $G(V,E)$ um grafo conexo qualquer onde $|V| = n \geq 2$. Considere ainda um dado equilibrado com $f \geq 2$ faces. Se $\mathbf{D} = \max\{|N(n_i)|: n_i \in V\}$ então:

$$Pr(S_i) \geq h(\Delta, f), \text{ onde } h(\Delta, f) = \left(\frac{1}{\Delta+1}\right) \left(1 - \frac{1}{f}\right)^{\Delta+1} + \frac{1}{2f} \left(1 - \frac{1}{f}\right)^{\Delta}$$

e $n_i \in V$.

Prova: Da expressão de probabilidade absoluta (vide [JAM 81]) temos que:

$$Pr(S_i) = \sum_{\mathbf{a}=0}^{f-1} Pr(d_i = \mathbf{a}) \cdot Pr(S_i | d_i = \mathbf{a}), \quad \forall n_i \in V \quad (I)$$

Seja $B_j(n_i, \infty)$ p/ $j=1, \dots, |\mathbf{N}n_i|$, um conjunto de eventos onde $d_j < d_i$ para $n_j \in \mathbf{N}n_i$ e $d_i = \infty$. Note que estes eventos são independentes entre si. Além disso, temos $Pr(B_j(n_i, \infty)) = \infty/f$. Assim:

$$Pr(S_i | d_i = \mathbf{a}) = Pr\left(\bigcap_{n_j \in \mathbf{N}(n_i)} B_j(n_i, \mathbf{a})\right) = \prod_{n_j \in \mathbf{N}(n_i)} \left(\frac{\mathbf{a}}{f}\right) = \left(\frac{\mathbf{a}}{f}\right)^{|\mathbf{N}(n_i)|} \quad (II)$$

Substituindo (II) em (I) e lembrando que $Pr(d_i = \mathbf{a}) = 1/f$ obtemos:

$$Pr(S_i) = \sum_{\mathbf{a}=0}^{f-1} \left(\frac{1}{f}\right) \left(\frac{\mathbf{a}}{f}\right)^{|\mathbf{N}(n_i)|} \geq \sum_{\mathbf{a}=0}^{f-1} \left(\frac{1}{f}\right) \left(\frac{\mathbf{a}}{f}\right)^{\Delta}, \text{ p/ } \Delta = \max\{|N(n_i)|: n_i \in V\}$$

Logo:

$$Pr(S_i) \geq \left(\frac{1}{f^{\Delta+1}}\right) \sum_{\mathbf{a}=0}^{f-1} \mathbf{a}^{\Delta} \quad (III)$$

Como $n \geq 2$ e G é conexo então $\mathbf{D} \geq 1$. Temos então de [SCH 92] que:

$$\sum_{\mathbf{a}=0}^{f-1} \mathbf{a}^{\Delta} \geq \left(\frac{(f-1)^{\Delta+1}}{\Delta+1} + \frac{(f-1)^{\Delta}}{2}\right) \quad (IV)$$

Finalmente, substituindo (IV) em (III) chegamos à expressão desejada. •

Seja S um evento indicando a união de n eventos distintos S_i , ou seja, S ocorre se S_i ocorre para algum $i \in \{1, \dots, n\}$. O resultado seguinte (conseqüência do Teorema 2), mostra que, se G é completo e o número de faces tende a infinito a probabilidade de orientação das arestas incidentes a algum $n_i \in V$, tende a 1. Em outras palavras, isto significa que o empate entre dados adjacentes tende a 0.

Corolário 1: Se G é completo e $f \geq 2$ então $Pr(S) \geq 1/n$.

Prova: Como G é completo então $\Delta = n-1$. Fazendo $f \geq 2$ na expressão de $h(\mathbf{D}, f)$ (Teorema 1), temos $Pr(S_i) \geq 1/n$ e portanto $Pr(S) \geq 1/n$. •

Teorema 3. Considere $G(V,E)$ um grafo conexo qualquer, onde $|V| = n \geq 2$. Além disso, considere \mathbf{D} o grau máximo de G , e um dado equilibrado com $f = \mathbf{D}+1$ faces. O algoritmo *Alg-Viz* terá tempo esperado de processamento igual a $O((\mathbf{D}+1).e)$.

Prova: Temos que $f = \mathbf{D}+1$ (hipótese). Segue então do Teorema 2 que:

$$Pr(S_i) \geq \left(\frac{1}{\Delta+1}\right) \left(1 - \frac{1}{\Delta+1}\right)^{\Delta} \left(1 - \frac{1}{\Delta}\right) + \frac{1}{2(\Delta+1)} \left(1 - \frac{1}{\Delta+1}\right)^{\Delta}$$

onde o evento S_i (associado a $n_i \in V$) ocorre, se e somente se, $d_j < d_i, \forall n_j \in \mathbf{N}n_i$. Substituindo $(1 - 1/(\Delta+1))^{\Delta} \geq 1/e$ na desigualdade acima obtemos:

$$Pr(S_i) \geq \left(\frac{1}{\Delta+1}\right) \left(\frac{3}{2} - \frac{1}{\Delta+1}\right) \left(\frac{1}{e}\right)$$

Como G é um grafo conexo e $n \geq 2$ (hipótese), conclui-se diretamente que $\mathbf{D} \geq 1$. Logo, o evento S representando a união dos eventos disjuntos S_i (p/ $i=1, 2, \dots, n$) será obtido pela seguinte expressão:

$$Pr(S) = Pr\left(\bigcup_{i=1}^n S_i\right) = \sum_{i=1}^n Pr(S_i) \geq \frac{n}{(\Delta+1).e}$$

Note agora que $(\mathbf{D}+1).e$ n^{-1} repetições (valor esperado) serão necessárias até que pelo menos um evento S_i ocorra. Como temos $|V| = n$, segue que a complexidade esperada será igual a $O((\mathbf{D}+1).e)$. •

Este resultado é bastante interessante já que expressa a complexidade de tempo independentemente do número de vértices e arestas. É importante lembrar que, em um sistema distribuído anônimo, o número de informações que se tem do sistema é bastante reduzido. Além disso, como conseqüência imediata do Teorema 2 temos o seguinte resultado:

Corolário 2: Se G é completo então *Alg-Viz* terá complexidade esperada igual a $O(n.e)$. •

Apesar da melhoria introduzida pela utilização de dados polarizados (Figura 2), pode-se constatar que, para grafos completos, o tempo esperado de processamento nunca será inferior a $O(n)$. Isto se deve ao fato de que, em um grafo completo, apenas uma orientação (evento S_i) ocorre por vez (eventos disjuntos).

Antes de prosseguirmos na apresentação das propriedades de convergência de *Alg-Viz*, precisamos definir exatamente o que consideraremos conectividade de um grafo daqui para frente. Isso se deve ao fato de a conectividade de um grafo, no sentido de "quão conexo" é o grafo, exerce forte influência sobre o tempo de convergência, já que grafos mais conexos tenderão a convergir mais devagar para o fim.

Para nós, a conectividade q de um grafo será a razão entre o número de arestas que o grafo possui e o número de arestas que ele poderia possuir com o mesmo número de nós. Logo:

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

Pela Figura 2, pode-se observar uma característica interessante acerca da velocidade de convergência do *Alg-Viz*. Utilizaremos daqui para frente números grandes de f de tal forma que este não prejudique a velocidade de convergência do algoritmo. Como discutido anteriormente, isto praticamente anula a possibilidade de empates nos sorteios. Pelo gráfico, claramente, temos que, mantendo-se q constante, a velocidade de convergência é função linear de n . Além disso, o coeficiente da reta fica muito próximo de q .

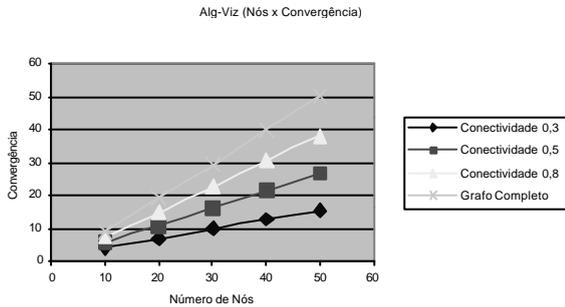


Figura 2: *Alg-Viz* (c/ dados polarizados)- com q constante, a convergência é função linear de n .

C. Alg-Arestas

Este algoritmo apresenta uma filosofia um pouco diferente dos anteriores (*Alg-Viz* e *Calabrese/França*). Em vez de tentar gerar sumidouros entre os nós probabilísticos, *Alg-Arestas* muda o foco de atenção para as arestas.

O algoritmo também é simples e funciona assim: considere um nó n_i e todos os seus vizinhos v_{ij} ligados a n_i pelas arestas m_{ij} . Todos os nós do sistema sorteiam um número com um dado de f faces, obtendo $dado(n_i)$. n_i irá comparar $dado(n_i)$ com $dado(v_{ij})$ para todo v_{ij} tal que m_{ij} ainda não foi orientada. Para cada m_{ij} em que

$dado(n_i) > dado(v_{ij})$, orienta m_{ij} na direção de n_i . O processo é repetido até não sobrar nenhuma aresta sem orientação.

Uma prova formal de que *Alg-Arestas* funciona é dada abaixo:

Teorema 4. *Alg-Arestas* sempre gera orientações acíclicas em qualquer grafo $G(V,E)$.

Prova. Definimos K como sendo o conjunto de todos os ciclos simples (sem nós repetidos) em $G(V,E)$. A idéia é provar que o algoritmo não gera ciclos para nenhum K . K é composto dos nós $u_0, \dots, u_{|K|-1}$, considerados em ordem de vizinhança no ciclo, de tal forma que u_i é conectado por uma aresta de K a $u_{(i+1) \bmod |K|}$, $0 \leq i < |K|$. Os números sorteados para os nós u_i são dados por u_i^s . Temos duas possibilidades:

(a) $u_0^s = u_1^s = \dots = u_{|K|-1}^s$ - Nesse caso, nenhuma aresta do ciclo é orientada e o sorteio é executado novamente;

(b) $\exists a, 0 \leq a < |K|-1$, tal que $u_a^s < u_{(a+1) \bmod |K|}^s$ - Nesse caso, tomemos uma nova nomeação dos nós como $v_0, v_1, \dots, v_{|K|-1}$ tal que $v_i = u_{(a+i) \bmod |K|}$. Como $v_0 = u_a$ e $v_1 = u_{(a+1) \bmod |K|}$, já sabemos que $v_0^s < v_1^s$, portanto a aresta (v_0, v_1) terá a direção $v_0 \rightarrow v_1$. Desta forma, se " $j, 1 \leq j < |K|-2$, tivermos que $v_j^s < v_{j+1}^s$, então $v_{|K|-1}^s > v_0^s$ e, portanto, a aresta $(v_{|K|-1}, v_0)$ será orientada na direção $v_0 \rightarrow v_{|K|-1}$, tornando v_0 um sumidouro nas arestas de K impossibilitando a formação de ciclo em K nos passos seguintes do algoritmo. Por outro lado, se $\exists j, 1 \leq j < |K|-2$ tal que $v_j^s > v_{j+1}^s$, então a aresta (v_j, v_{j+1}) será orientada na direção $v_j \rightarrow v_{j+1}$. Assim, teremos a seguinte situação (o símbolo \leftrightarrow significa orientação indefinida) que define a impossibilidade de formação de um ciclo em K nos passos seguintes do algoritmo.

$$v_{|K|-1} \leftrightarrow v_0 \rightarrow v_1 \leftrightarrow \dots \leftrightarrow v_j \leftarrow v_{j+1} \leftrightarrow \dots \leftrightarrow v_{|K|-1} \bullet$$

C.1 Velocidade de convergência

É bastante fácil perceber que *Alg-Arestas* tem convergência muito mais rápida que *Alg-Viz*. Nesta seção provaremos que o tempo esperado de convergência será inferior a $O(n)$ mesmo considerando-se grafos completos.

A idéia da prova de complexidade consiste no seguinte: em um sorteio, a probabilidade de empate entre dois vizinhos é de $1/f$. Assim, enquanto o número de arestas é grande, esta é a proporção de empates ocorridos a cada passo do algoritmo. Assim, em cada passo, uma proporção de $1/f$ das arestas não orientadas até aquele passo permanecem sem serem orientadas. Portanto, a convergência é dada por $O(\lceil \log_f m \rceil)$, onde m é o número de arestas do grafo original $G(V,E)$. Apesar de intuitivamente razoável e de apresentar resultados próximos da realidade, esta análise é dificultada já que a proporção $1/f$ só é válida quando o número de arestas sendo consideradas é suficientemente grande.

Mais formalmente, este processo pode ser descrito por uma relação de recorrência do tipo $T(x)=a(x)+T(h(x))$, onde x é variável real não-negativa, $a(x)$ é função real não-decrescente e não-negativa de x , e $h(x)$ é variável aleatória assumindo valores no intervalo $[0,x]$. Além disso, considere uma função real $g(x)$ onde $E(h(x)) \leq g(x)$, sendo $g(x)$ e $g(x)/x$ funções não-decrescentes e não-negativas de x [KAR 94]. A abordagem de [KAR 94] é bastante genérica já que, não se exige, a priori, nenhuma informação sobre a distribuição de $h(x)$. Note que $a(x)$ representa o esforço necessário para se quebrar o problema original em um problema de tamanho $h(x)$.

A equação $\mathbf{t}(x)=a(x)+\mathbf{t}(g(x))$, pode ser vista como uma versão determinística da relação de recorrência probabilística $T(x)$. Neste caso, para todo x , a variável aleatória $h(x)$ será exatamente igual ao limite superior de seu valor esperado, ou seja, faremos $E(h(x))=g(x)$. Como discutido em [KAR 94], o Teorema do Ponto-Fixo de Tarski garante que, sempre que esta equação tem solução não-negativa, existe uma única solução (não-negativa) $u(x)$ onde $u(x) \leq v(x)$, $\forall x$, e qualquer que seja $v(x)$ não-negativa. A função $u(x)$ é dada explicitamente pela fórmula:

$$u(x) = \sum_{i=0}^{\infty} a(g^{(i)}(x))$$

onde $g^{(i)}(x)$ é definido indutivamente por $g^{(i)}(x) = g(g^{(i-1)}(x))$ para $i=1,2,\dots$ e $g^{(0)}(x)=x$.

Temos então o seguinte resultado:

Teorema 5. Suponha que exista uma constante d tal que $a(x)=0$, $x < d$ e $a(x)=1$, $x \geq d$. Seja $c_t = \min\{x/u(x)\}$. Então, para todo real positivo x e todo inteiro positivo w ,

$$\Pr[T(x) \geq u(x) + w] \leq \left(\frac{g(x)}{x}\right)^{w-1} \frac{g(x)}{c_{u(x)}}$$

Prova: [KAR 94]. •

Note, da proposição acima que, se $u(x)$ representa o tempo esperado de processamento então $\Pr[T(x) \geq u(x) + w]$ indica a probabilidade de nos afastarmos de $u(x) + w$ (onde w é inteiro positivo). Este resultado é interessante pois nos dá uma medida do grau de variação de $T(x)$ em torno de seu valor esperado. O Teorema 6, como veremos a seguir, pode ser utilizado na prova da velocidade de convergência do procedimento Alg-Arestas.

Teorema 6. Dado um grafo $G(V,E)$ com m arestas e um dado equilibrado com f faces, o algoritmo Alg-Arestas tem tempo esperado de processamento igual a $O(\lfloor \log_f m \rfloor + 1)$. Além disso:

$$\Pr[T(m) \geq \lfloor \log_f m \rfloor + 1 + w] \leq \left(\frac{1}{f}\right)^{w-1} \frac{m}{f^{\lfloor \log_f m \rfloor + 1}}$$

Prova. Observe em nosso caso que, fazendo-se $x=m$ e $d=1$, temos $a(m)=1$ para $m \geq 1$ e $a(m)=0$ para $m < 1$. Além disso, $h(m)$ é variável aleatória indicando quantidade de arestas resultantes não orientadas após a primeira iteração.

Cada aresta é orientada independentemente das outras. Portanto, podemos definir uma variável aleatória de Bernoulli que modela cada tentativa de orientação de uma aresta, ou seja, a disputa de dados entre os dois nós que são conectados por esta. A probabilidade de fracasso, ou seja, de a aresta não se orientar, é igual à de dar empate na disputa entre dois dados equilibrados de número de faces iguais a f , ou seja, $1/f$. Assim, $1/f$ representa a probabilidade de fracasso e $1-1/f$ a probabilidade de sucesso.

Note que número total de arestas orientadas a cada iteração (somatório dos ensaios de Bernoulli) pode ser representado por uma variável aleatória com distribuição binomial e valor esperado $m(1-1/f)$. Logo, $g(m) = m/f$, irá representar o número de arestas não orientadas após a primeira iteração. Além disso, temos que $E[h(m)]=g(m)$, $g(m)$ e $g(m)/m$ são funções não-decrescentes. A solução $u(m)$ de $\mathbf{t}(m)=a(m)+\mathbf{t}(g(m))$ pode então ser obtida da seguinte forma:

$$\begin{aligned} u(m) &= a(m) + a(m/f) + u(m/f^2) \\ &= 2 + u(m/f^2) \\ &= 3 + u(m/f^3), \text{ etc.} \end{aligned}$$

Após k passos obtemos $u(m)=k + u(m/f^k)$. Repetimos o processo até que $m/f^k = 1 \Rightarrow k = \log_f m$. Como k é inteiro fazemos $k = \lfloor \log_f m \rfloor$. É fácil ver agora que $u(1)=1$, assim $u(m) = \lfloor \log_f m \rfloor + 1$.

Para provar a segunda parte, mostraremos que $c_t = f^{t-1}$. De fato, $c_t = \min\{m/u(m)\}$ (Teorema 5). Assim:

$$u(m) \geq t \Rightarrow \Pr[u(m) \geq t] \leq \Pr[m/u(m) \geq t] \leq \Pr[\log_f m \geq t-1] \leq f^{t-1}$$

Portanto, temos que: $c_t = \min\{m/m^{t-1}\} = f^{t-1}$. Finalmente, substituindo $u(m)$, $c_{u(m)}$ e $g(m)=m/f$ em:

$$\Pr[T(m) \geq u(m) + w] \leq \left(\frac{g(m)}{m}\right)^{w-1} \frac{g(m)}{c_{u(m)}}$$

obtemos:

$$\Pr[T(m) \geq \lfloor \log_f m \rfloor + 1 + w] \leq \left(\frac{1}{f}\right)^{w-1} \frac{m}{f^{\lfloor \log_f m \rfloor + 1}}$$

como queríamos demonstrar. •

Para ilustrar esse resultado, note que, se $m = f^k$ para algum $k \in \mathbb{Z}^+$ e $w=1$ então: $Pr[T(m) \leq u(m)+1] \approx 1/f$. Este resultado é interessante pois mostra que a probabilidade de nos afastarmos do valor esperado independe do número de arestas em G e é inversamente proporcional ao número de faces do dado.

Note ainda que, mesmo para grafos completos (onde $m=O(n^2)$), o desempenho do *Alg-Arestas* é significativamente superior ao *Alg-Viz*. O tempo de convergência de *Alg-Arestas* é bastante pequeno quando comparado com *Alg-Viz*. Na verdade, na situação em que $f \gg m$, podemos considerar que o algoritmo converge em apenas um (1) passo, o que é um resultado muito bom, já que podemos sem muito custo utilizar valores bastante grandes de f .

TABELA I

COMPARAÇÃO ENTRE A VELOCIDADE ESPERADA DE CONVERGÊNCIA E AS OBTIDAS NAS SIMULAÇÕES

Arestas	Faces	Simulação	$\lceil \log_2 m \rceil$
100	10	2.70	2
200	110	1.82	2
300	210	1.79	2
400	310	1.72	2
500	410	1.70	2

A Tabela I exhibe, para alguns valores de m e f , os resultados obtidos por simulação (média aritmética obtida através da simulação de mil (1000) de grafos conexos gerados aleatoriamente), pela expressão aproximada $\lceil \log_2 m \rceil$.

Na Figura 3 podemos observar o tempo de convergência do algoritmo para diferentes valores de f e m . Note que os valores são realmente pequenos, mesmo para um número reduzido de faces do dado.

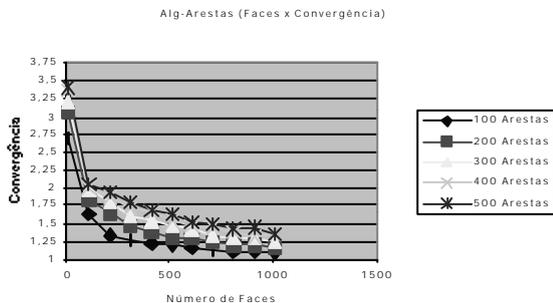


Figura 3: Velocidade de convergência de *Alg-Arestas*.

IV. APLICAÇÕES ALVO

Nesta seção estaremos analisando duas aplicações onde a definição de uma orientação acíclica pode ser um passo frequentemente requerido durante o processamento da aplicação ou sua correta inicialização. A primeira aplicação é sobre o problema do escalonamento distribuído de recursos entre processos em sistemas distribuídos. Uma segunda aplicação versa sobre uma estratégia de distribuição de *uploads* em redes de computadores.

A. Escalonamento por Reversão de Arestas

O *Escalonamento por Reversão de Arestas* [BAR 89] [BAR 96] visa resolver o problema de controlar o acesso a recursos compartilhados entre processadores em um ambiente distribuído. Assim, dado um conjunto V de nós de processamento, um conjunto R de recursos e um conjunto $D \subseteq (V \times R)$ dos pares $(n \in V, r \in R)$ em que o nó n utiliza o recurso r , determinar a ordem de execução (ou operação) dos nós de tal forma que dois nós que compartilham o mesmo recurso não executem ao mesmo tempo e de tal forma que não ocorra *deadlock* nem *starvation*. O ERA foi criado para condições de alta carga no sistema, ou seja, cada nó está sempre precisando de todos os recursos compartilhados ao seu alcance para que possa operar.

A partir dessa definição podemos construir um *Grafo de Dependências* $G^D(V, E^D)$ onde $E^D \subseteq V \times V$, que irá representar os padrões de compartilhamento do sistema. O conjunto E^D é composto das arestas que ligam dois nós que compartilham pelo menos um recurso, ou seja, $m = (n_i, n_j) \in E^D$ se, e somente se, $(n_i, n_j) \in E$ e $\exists r \in R$, tal que, $(n_i, r) \in D$ e $(n_j, r) \in D$. Observe que, nessa definição, assumimos que um recurso só poderá ser compartilhado entre nós que tenham um canal de comunicação entre si, o que é razoável. Também vale observar que, nessa notação, cada recurso compartilhado corresponde a um subgrafo completamente conectado, já que todos os nós que o compartilham terão arestas entre eles.

O algoritmo ERA admite a existência inicial de uma orientação acíclica no grafo G^D . Basicamente uma aresta (n_i, n_j) orientada no sentido $n_i \rightarrow n_j$ indica que o nó n_j tem precedência em relação ao nó n_i na próxima operação. A necessidade de a orientação ser acíclica é facilmente percebida, já que, de acordo com a definição acima, um ciclo implicaria uma cadeia de precedências entre nós organizada de forma a gerar um bloqueio perpétuo ou *deadlock*.

Assim, um nó para operar precisa ser um *sumidouro*, ou seja, deverá ter todas as arestas orientadas em sua direção, já que isso equivale a ter precedência sobre todos os vizinhos (como já foi dito, vizinhos em G^D não podem operar ao mesmo tempo). Uma orientação acíclica em G^D implica a existência de pelo menos um sumidouro em V , assim é garantido que o algoritmo pode sempre iniciar com

os sumidouros operando. Após sua operação, cada sumidouro reverte as suas arestas na direção oposta. A nova orientação gerada é também acíclica, pois todos os sumidouros foram transformados em fontes e esse tipo de operação claramente não cria ciclos. Por isso, um novo conjunto de sumidouros será obtido. O processo, então se repete, com os sumidouros operando e revertendo suas arestas, definindo assim o escalonamento pretendido.

B. Upload Distribuído

Nesse problema, visamos contornar o congestionamento natural no tráfego de informações em uma determinada rede de comunicação, na situação particular onde um único ponto de recepção é responsável pela captação de todas as informações provenientes de todos os pontos dessa rede. Esta é uma situação bastante comum, onde uma data limite para entrega de dados por toda uma população específica é estabelecida e o fluxo de dados se intensifica nas proximidades dessa data provocando o congestionamento dos canais de comunicação, e.g., entrega da declaração do imposto de renda.

Uma alternativa viável para atenuar esse quadro estaria na multiplicação de nós receptores na rede. O problema do *upload distribuído* consiste na determinação do escoamento apropriado dos dados em direção aos nós receptores. Uma abordagem possível estaria na geração de uma orientação acíclica sobre o grafo G representando a rede de comunicações alvo, onde os nós de G seriam os roteadores e as arestas de G os canais de comunicação existentes. Nessa orientação acíclica, feita sobre as arestas de G , os roteadores associados aos pontos de recepção corresponderiam aos únicos nós sumidouros. Para se obter tal configuração orientar-se-ia inicialmente as arestas adjacentes aos nós receptores de forma a definir o conjunto de sumidouros desejado. A seguir, o uso de qualquer dos algoritmos para geração de orientações acíclicas poderia ser utilizado sobre o restante de G . A eventual surgimento de novos sumidouros, indesejados, seria facilmente eliminado via a aplicação do algoritmo ERA apresentado anteriormente.

V. CONCLUSÕES

Foram apresentados e analisados dois algoritmos distribuídos probabilísticos para geração de orientações acíclicas em sistemas distribuídos anônimos: *Alg-Viz* (uma extensão direta do algoritmo *Calabrese/França*) e *Alg-Arestas* (que, em certas circunstâncias, gera orientações acíclicas em menos de 2 passos).

Provamos a convergência teórica de *Alg-Viz* e *Alg-Arestas*. Além disso, a análise de seus tempos esperados de processamento foi realizada, tanto através de simulações quanto teoricamente.

Apresentamos ainda duas aplicações onde a definição de uma orientação acíclica era passo essencial no

processamento da aplicação. Na primeira aplicação discutiu-se o problema do escalonamento distribuído de recursos entre processos em sistemas distribuídos. Na segunda aplicação esboçamos uma estratégia de distribuição de *uploads* em redes de computadores.

Um possível trabalho futuro é a utilização dos algoritmos apresentados para gerar orientações em multigrafos como passo inicial para *SMER* (*Scheduling by Multiple Edges Reversal*) [BAR 01], uma generalização de ERA para sistemas heterogêneos no tocante ao regime de carga dos processos.

REFERÊNCIAS

- [ARA 99] ARANTES Jr, G.M., “Orientações Acíclicas em Sistemas Distribuídos Anônimos e suas Aplicações no Compartilhamento de Recursos”, Tese de M.Sc., COPPE/UFRJ, Setembro de 1999.
- [ARA 01] ARANTES Jr, G.M. e FRANÇA, F.M.G., “Geração Quase-Instantânea de Orientações Acíclicas em Sistemas Distribuídos Anônimos”, *Anais do II Workshop em Sistemas Computacionais de Alto Desempenho*, Pirenópolis, GO, pp. 55-62, Setembro de 2001.
- [BAR 01] BARBOSA, V. C., BENEVIDES, M.R.F. and FRANÇA, F.M.G. “Sharing resources at nonuniform access rates”, *Theory of Computing Systems*, vol. 34, Springer, pp. 13-26, 2001.
- [BAR 89] BARBOSA, V. C. "Concurrency in Heavily Loaded Neighborhood-Constrained Systems", *ACM Transactions on Programming Languages and Systems*, Vol. 11, n. 4, pp. 562-584, Outubro 1989.
- [BAR 96] BARVOSA, V. C. *An Introduction to Distributed Algorithms*, MIT Press 1996.
- [CAL 94] CALABRESE, A. and FRANÇA, F.M.G. "Randomized Distributed Primer for the Updating Control of Anonymous ANNs", *Proceedings of ICANNR94*, Sorrento, Italy, 1994.
- [CAL 97] CALABRESE, A. "Distributed Acyclic Orientation of Asynchronous Networks", *11th International Symposium of Fundamentals of Computation Theory*, pp.129-137, Kraków, Poland, Setembro 1997.
- [FRA 91] FRANÇA, F.M.G. "A Self-Organizing Updating Network", *Artificial Neural Network*, eds. T. Kohonen, K. Mckisara, O. Simula, and J. Kangas, Elsevier Science Publisher B. V. (North-Holland), pp. 1349-1352, 1991.
- [FAR 99] FARIA, L. e FRANÇA, F. M. G. Comunicação Pessoal.
- [GOL 87] GOLDBERG, A. V. and PLOTKIN, S. A. "Parallel $(D+1)$ -Coloring of Constant-Degree Graphs", *Information Processing Letters*, n. 25, pp. 241-245, 1987.
- [GOL 88] GOLDBERG, A. V. and PLOTKIN, S. A. "Parallel Symmetry-Breaking in Sparse Graphs", *SIAM Journal of Discrete Mathematics*, Vol. 1, n .4, pp. 434-445, Novembro 1988.

- [ITA 90] ITAI, A. and RODEH, M. "Symmetry-Breaking in Distributed Networks", *Information and Control*, n. 88, pp. 60-87, 1990.
- [JAM 81] JAMES, B.R., "Probabilidade: um curso em nível intermediário", Projeto Euclides 1981.
- [KAR 94] KARP R., "Probabilistic Recurrence Relations", *Journal of Association for Computing Machinery*. Vol 41. No 6. November 1994, pp 1100-1136
- [LUB 86] LUBY, M., "A Simple Parallel Algorithm for the Maximal Independent Set Problem", *SIAM Journal of Computing*, Vol. 15, n. 4, pp. 1036-1053, Novembro 1986.
- [PAN 92] PANCONESI, A. and SRINIVASAN, A., "Improved Distributed Algorithms for Coloring and Network Decomposition Problems", *24th ACM Symposium on Theory of Computation*, pp. 581-591, Victoria B. C., Canada, Maio 1992.
- [SCH 92] SPIEGEL, M. R., "Manual de Fórmulas, Métodos e Tabelas de Matemática", Makron, McGraw-Hill 1992 (Coleção Schaum).
- [SZE 93] SZEGEDY, M. and VISHWANATHAN, S. "Locality Based Graph Coloring", *25th ACM Symposium on Theory of Computation*, pp. 201-207, California, USA, Maio 1993.