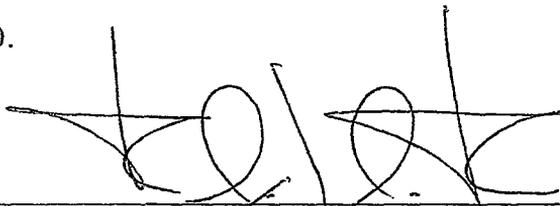


TRILHAS, OTIMIZAÇÃO DE CONCORRÊNCIA E INICIALIZAÇÃO
PROBABILÍSTICA EM SISTEMAS SOB REVERSÃO DE ARESTAS

Gladstone Moises Arantes Junior

TESE SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS
PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE
DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E
COMPUTAÇÃO.

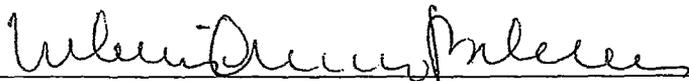
Aprovada por:



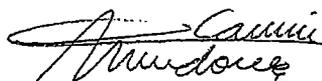
Prof. Felipe Maia Galvão França, Ph.D.



Prof. Luerbio Faria, D.Sc.



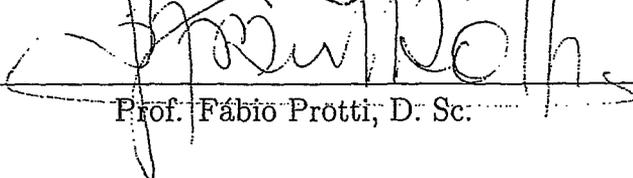
Prof. Valmir Carneiro Barbosa, Ph.D.



Prof. Candido Ferreira Xavier de Mendonça Neto, Ph.D.



Prof. Alberto Ferreira de Souza, Ph.D.



Prof. Fábio Protti, D. Sc.

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2006

ARANTES JUNIOR, GLADSTONE MOISES

Trilhas, Otimização de Concorrência e Inicialização Probabilística em Sistemas sob Reversão de Arestas [Rio de Janeiro] 2006

IX, 123 p. 29,7 cm (COPPE/UFRJ, D.Sc., Engenharia de Sistemas e Computação, 2006)

Tese – Universidade Federal do Rio de Janeiro, COPPE

1 - Algoritmos distribuídos

2 - Complexidade de algoritmos

3 - Algoritmos randômicos

I. COPPE/UFRJ II. Título (série)

*Para meu pai e meu filho
a essa linha contínua
estas que ficam*

Agradecimentos

Em primeiro lugar, quero dizer que escolheria mil vezes a família que nasci: agradeço ao meu pai por explicar sempre; à minha mãe pelo coração em que cabe o mundo; e à minha irmã por ser uma espécie de outro eu, mesmo tão diferente. E também quero dizer que escolheria mil vezes a família que ganhei. Agradeço à minha esposa por tudo que estamos construindo, pelo enorme apoio nesta reta final e pela plenitude do seu amor de mãe; e a você, meu filho, pelo mais belo de todos os espetáculos: ver você crescer.

Ao Prof. Felipe França, obrigado pela confiança, paciência e tudo mais. Acredite: este agradecimento é muito mais do que protocolar. E ao Prof. Luerbio Faria, pela energia e pelo exemplo de força e trabalho. Esta tese não teria sido possível sem seu esforço.

Agradeço ao Prof. Valmir Barbosa pelas excelentes aulas de Algoritmos Distribuídos e Redes de Autômatos (deu no que deu...). Ao Prof. Carlos Alberto Martinhon, pelo trabalho conjunto no estudo de algoritmos de inicialização probabilística.

Gostaria de fazer um agradecimento especial aos professores do Colégio Renovação, onde cursei o segundo grau e onde, pela primeira vez, convivi com um ambiente de estímulo ao questionamento e à busca de conhecimento. Agradeço, ainda mais especialmente, ao Prof. Calvo pela sua generosidade.

Posso dizer que devemos, eu e minha irmã, muito da nossa vida profissional e acadêmica a este professor.

Agradeço também à primeira pessoa que me ensinou matemática de verdade, meu colega Vinho, cuja boa vontade e dedicação nunca vou esquecer. Também agradeço a Márcio Guimarães, não apenas pela amizade mais sólida que construí em toda a vida, mas também por me acompanhar nos primeiros passos no interesse pela matemática ainda nas apostilas do Renovação.

Agradeço também a vários companheiros de BNDES. À Luciana Vasco, pela compreensão e pelo exemplo de trabalho e dedicação. Aos companheiros de “farda” acadêmica (ou fardo?), pelas conversas e filosofias sem chatices: José Ricardo Xavier; Vanessa Almeida; Nicolaas Ruberg; Leonardo Cardoso; e Flávio Mattos (entre outros). À Margarida Freire, pela força e amizade. Ao Herilmar Freire, pelo apoio e amizade. E à Lilian Mendes que, quis o destino, acabou sendo tão importante na retomada deste trabalho.

E, por fim, agradeço ao meu amigo Prof. Jorge de Abreu Soares, companheiro nesta caminhada longa, dura e traiçoeira, que é o curso de doutorado.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

TRILHAS, OTIMIZAÇÃO DE CONCORRÊNCIA E INICIALIZAÇÃO
PROBABILÍSTICA EM SISTEMAS SOB REVERSÃO DE ARESTAS

Gladstone Moises Arantes Junior

Setembro/2006

Orientadores: Felipe Maia Galvão França

Luerbio Faria

Programa: Engenharia de Sistemas e Computação

O Escalonamento por Reversão de Arestas (ERA) é um algoritmo de escalonamento para sistemas distribuídos com compartilhamento de recursos em alta carga. Este trabalho prova que os problemas de decisão relacionados à maximização e à minimização da concorrência do ERA são NP-completos para grafos com $\Delta \leq 4$ e para grafos cúbicos planares, respectivamente. É provada a não-aproximação com razão menor que $n^{\frac{1}{7}-\epsilon}$, se $P \neq NP$, para o problema de minimização do inverso da concorrência. Com o conceito de trilha, é provado que os ciclos principais determinam o máximo e o mínimo intervalo inoperante, além de outros atributos da execução do ERA. Sobre os algoritmos probabilísticos de inicialização do ERA, prova-se que Alg-Cor tem convergência sub-exponencial para grafos completos e linear se os dados têm $f = n$ faces. Para Alg-Arestas são analisadas expressões para $P[T(m) \geq \lfloor \log_f m \rfloor + 1 + w]$ e $P[T(m) = w]$, onde $T(m)$ é o tempo de convergência com m arestas e $w > 0$.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

TRACKS, CONCURRENCY OPTIMIZATION, AND PROBABILISTIC
INITIALIZATION ON SYSTEMS UNDER EDGES REVERSAL

Gladstone Moises Arnates Junior

September/2006

Advisors: Felipe Maia Galvão França

Luerbio Faria

Department: Systems Engineering and Computer Science

Scheduling by Edges Reversal (SER) is a scheduling algorithm for distributed systems with heavy loaded resource sharing. This work proves that the decision problems associated to the maximization and minimization of SER concurrency are NP-complete for graphs with $\Delta \leq 4$ and for cubic and planar graphs, respectively. It also proves the non-approximability in a factor less than $n^{\frac{1}{7}-\epsilon}$, if $P \neq NP$, of the inverse of concurrency minimization problem. Using track concept, it's also proven that critic cycles determine maximum and minimum non-operation interval, besides other properties of ERA execution. About the probabilistic ERA initialization algorithms, it's proven that Alg-Cor has sub-exponential convergence for complete graphs and linear convergence when $f = n$ faces dies are used. For Alg-Arestas, expressions for $P[T(m) \geq \lfloor \log_f m \rfloor + 1 + w]$ and $P[T(m) = w]$, where $T(m)$ is the convergence of the algorithm with m edges, are analysed and $w > 0$.

Sumário

1	Introdução	1
1.1	Escalonamento por Reversão de Arestas	2
1.2	Motivações e Principais Resultados	4
1.3	Estrutura da Tese	8
2	Conceitos Básicos	10
2.1	Escalonamento por Reversão de Arestas	10
2.1.1	Compartilhamento de Recursos	10
2.1.2	O Algoritmo ERA	11
2.1.3	Notações e Propriedades do ERA	12
3	Otimização da Concorrência do ERA	20
3.1	Introdução	20
3.2	Definições	21
3.3	NP-Compleitude da Concorrência Mínima	23
3.4	NP-Compleitude da Concorrência Máxima para Grafos com Grau no Máximo 4	24
3.4.1	NP-compleitude de NAEQ3SAT_3	25
3.4.2	Construção de uma Instância de CONMAX a partir de uma Instância de NAEQ3SAT_3	29

3.4.3	Prova da NP-Completeness de CONMAX	31
3.5	Não-aproximabilidade de MININCON para Grafos em Geral	38
3.6	Algoritmo Aproximativo para MAXCONMAX em Grafos com Grau Máximo Δ	40
3.7	Benchmark para Maximização de Concorrência	41
4	Trilhas e Propriedades do ERA	48
4.1	Introdução	48
4.2	Ciclo Principal (Senhor dos Anéis)	49
4.3	Trilhas na Execução do ERA	50
4.3.1	Propriedades das Trilhas	51
4.3.2	Trilha Estabelece Monotonicidade de Intervalo Inope- rante Anterior	52
4.3.3	Trilha Estabelece Monotonicidade de Intervalo Inope- rante Posterior	54
4.4	Execução Independente	55
4.4.1	Definição e Propriedades da Execução Independente	58
4.4.2	Execução Canônica	61
4.5	Propriedades das Trilhas e Ciclos Principais	62
4.5.1	Ciclo Percorrido por Trilha Infinita	62
4.5.2	Comportamento das Trilhas no Período do ERA	70
4.6	Novas Propriedades do ERA	74
4.6.1	Ciclos Principais Determinam Máximo Intervalo Ino- perante	74
4.6.2	Ciclos Principais Determinam Mínimo Intervalo Ino- perante	80
4.6.3	Maior Número de Operações Independentes	81
4.6.4	Ciclos Principais com Trilhas Consecutivas Equidistantes	82

5	Análise de Algoritmos Randômicos para Inicialização do ERA	85
5.1	Introdução	85
5.2	Algoritmo Calabrese/França	87
5.2.1	Notação	88
5.2.2	Algoritmo Não Polarizado	88
5.2.3	Algoritmo Polarizado	90
5.3	Alg-Viz	91
5.3.1	Alg-Cor	91
5.3.2	Análise de Complexidade	92
5.3.3	Alg-Viz Polarizado	95
5.4	Alg-Arestas	97
5.4.1	Corretude de Alg-Arestas	98
5.4.2	Análise de Tempo de Execução	99
5.4.3	Análise Através de Outra Técnica	108
5.4.4	Comparação dos Resultados	111
6	Conclusão	114
6.1	Trabalhos Futuros	116

Capítulo 1

Introdução

A área de algoritmos distribuídos apresenta hoje em dia interesse renovado da comunidade científica. Não só pela disseminação das redes sem fio (*wireless networks*), mas também pelo novo interesse despertado pelas redes de sensores [11][6][24], assim como para redes de alto desempenho. Conceitos de computação distribuída estão sendo utilizados até mesmo internamente pelas arquiteturas de processadores de última geração [15][14]. Nesse sentido, o desenvolvimento de novas aplicações não conhecidas e o surgimento de novos requisitos de funcionamento é bastante natural.

Desta forma, existe demanda pela descoberta de novos algoritmos, assim como pelo aprofundamento do conhecimento sobre técnicas já estabelecidas que possam ser aplicadas em novos contextos e de formas diferentes.

O presente trabalho revisita o *Escalonamento por Reversão de Arestas (ERA)*, algoritmo de escalonamento distribuído bastante conhecido e com grande espectro de possíveis aplicações, assim como outros algoritmos relacionados, na tentativa de aprofundar a compreensão sobre suas propriedades.

1.1 Escalonamento por Reversão de Arestas

Mesmo sendo possível descrever a execução do ERA em termos bastante simples e a partir apenas de diretivas locais, o ERA, como o chamamos na maior parte deste trabalho, ou o SER (*Scheduling by Edges Reversal*), como é conhecido na literatura internacional, apresenta um comportamento global com propriedades e complexidade surpreendentes.

Como descrito em BARBOSA[8], partindo-se de uma orientação acíclica em um grafo onde os nós representam as unidades computacionais que compartilham recursos e uma aresta representa o compartilhamento de pelo menos um recurso entre os nós adjacentes, cada sumidouro (nó com todas as arestas adjacentes orientadas em sua direção) utiliza todos os recursos e, em seguida, inverte todas as suas arestas. Com isso, novos sumidouros serão criados e o processo se repete. Considera-se que todo nó quer operar durante todo o tempo e que, para operar, precisa de todos os recursos que compartilha com outros nós.

Nenhum conhecimento global é necessário para que o algoritmo funcione corretamente e garanta propriedades como: a não existência de *deadlock* (bloqueio perpétuo), situação onde um subconjunto dos nós nunca opera; e a não existência de *starvation* (inanição), situação onde um subconjunto dos nós fica muito tempo sem operar. Portanto, a partir de um comportamento puramente local e simples, produz-se um comportamento global complexo e de grande interesse.

Uma das propriedades do ERA (a que garante a não existência de *starvation*) é sua convergência para uma fase periódica de tamanho p na qual todos os nós do sistema operam o mesmo número de vezes a cada período independentemente de arquitetura ou topologia da rede. Uma medida geral de *concorrência* do sistema, portanto, pode ser definida como o número de

vezes que cada nó opera no período dividido pelo tamanho do ciclo ($\gamma = \frac{m}{p}$). A concorrência é totalmente definida e pode ser calculada em tempo polinomial a partir de características estáticas da orientação acíclica inicial, o que significa que um atributo comportamental dinâmico e complexo como a concorrência do sistema pode ser previsto a partir de características estáticas deste. Sabe-se [8] que a concorrência é determinada por um subconjunto, possivelmente unitário, de ciclos do grafo, ao qual chamamos de *ciclos principais*.

Há resultados [19] que provam que o período pode ter tamanho exponencial em n (número de nós do grafo). Já a fase pré-periódica tem tamanho polinomial em n [20].

Em ARANTES[1] e, posteriormente, em ARANTES[2], foram introduzidos algoritmos randômicos e distribuídos que geram orientações acíclicas mesmo em grafos anônimos. Com isso, mesmo a orientação acíclica inicial não impõe nenhum conhecimento global e é conseguida a partir de um comportamento local também bastante simples, reproduzido no Capítulo 5 - Análise de Algoritmos Randômicos para Inicialização do ERA. Um dos algoritmos converge em até 1 passo, se o número de “faces do dado” utilizado no sorteio for suficientemente grande [1][2][3][4].

Além da velocidade de convergência, em ARANTES[1] foram avaliadas as qualidades da concorrência gerada. Neste mesmo trabalho foi verificado experimentalmente, através de simulações, que o algoritmo Alg-Cor (apresentado no Capítulo 5 - Análise de Algoritmos Randômicos para Inicialização do ERA) gera as mais eficientes orientações.

1.2 Motivações e Principais Resultados

Complexidade da Otimização de Concorrência

Para várias aplicações possíveis para o ERA, a questão da concorrência é crucial. Desta forma, estudando a complexidade do problema para classes de grafos simples, é apresentada uma demonstração de que a complexidade do problema de decisão relacionado à maximização da concorrência do ERA é NP-completo mesmo para grafos com grau máximo limitado a 4. Também é mostrado que, se $P \neq NP$, o problema de minimização do inverso da concorrência não pode ser aproximado por uma razão menor do que $n^{\frac{1}{7}-\varepsilon}$ para qualquer valor de $\varepsilon > 0$. Além disso, um algoritmo polinomial com razão de desempenho de, no máximo, $\frac{2}{\Delta}$, onde Δ é o grau máximo do grafo, baseado no algoritmo de coloração descrito em LOVÁSZ[18], é introduzido.

Existem aplicações em que a utilização do ERA é desejável em regimes de concorrência baixa. Entre estas aplicações está o uso do ERA para escalonamento em rede de sensores onde o consumo de energia deve ser minimizado [11][6]. Outra possibilidade de aplicação da concorrência mínima é no problema de descontaminação distribuída de grafos [9], onde o objetivo é espalhar o menor número possível de guardas nos nós contaminados de um grafo, de tal forma que os guardas descontaminem todos os nós, sem permitir recontaminação, que acontece quando um nó descontaminado torna-se vizinho de outro contaminado. Nestes casos, a localização dos guardas seria guiada pelo ERA. Como o número de guardas apresenta relação com a concorrência, esta deveria, portanto, ser minimizada.

Neste trabalho, é mostrado que o problema de minimizar a concorrência é também NP-completo para grafos em geral. Desta forma, ao contrário da coloração, que é um problema muito relacionado à otimização da concorrência

do ERA, este é um problema NP-difícil mesmo quando tentamos minimizar seu resultado.

Modulação do Sistema pelos Ciclos Principais

Um ponto de central interesse neste trabalho é o fenômeno da modulação do sistema pelos ciclos principais. Modulação, neste caso, refere-se ao fato de que várias propriedades do comportamento dinâmico do sistema podem ser previstas pela observação apenas dos ciclos principais do sistema. A mais conhecida destas propriedades dinâmicas é a própria concorrência. A concorrência apresenta ainda a característica interessante de ser totalmente determinada a partir da observação estática destes ciclos.

Neste trabalho, verificamos outras medidas de comportamento dinâmico que são determinadas pelos ciclos principais. Ao contrário da concorrência, tais medidas não são calculadas a partir de características estáticas do ciclo principal. Nos resultados apresentados são relacionadas características dinâmicas do sistema ao comportamento também dinâmico dos ciclos principais. Porém, este comportamento dinâmico dos ciclos principais apresenta características de certa forma estáticas, as quais são descritas nas definições de *execução independente* e de *execução canônica*, conceitos introduzidos no Capítulo 4 - Trilhas e Propriedades do ERA.

Acredita-se que este fenômeno da modulação do sistema pelos ciclos principais poderá ser explorado em algumas aplicações, através da configuração manual de um ciclo principal ou pela sua detecção distribuída, possivelmente através de heurísticas. Além disso, acredita-se que, futuramente, as já citadas características estáticas do comportamento dinâmico dos ciclos principais possam ser úteis como ferramenta para a descoberta de novas propriedades do ERA. Citamos como exemplo o momento em que o ERA entra no período

ou se uma dada concorrência ótima é equivalente a uma coloração ótima. Estas são algumas das perguntas que nos motivaram a tais estudos e ainda constituem-se como objetos para novas pesquisas.

Em relação ao estudo da modulação do sistema pelos ciclos principais, dois dos resultados apresentados versam sobre o máximo e o mínimo intervalo inoperante de um nó. O máximo intervalo de inoperância é uma medida dada pela coloração associada à orientação do sistema. Neste trabalho, é mostrado que esta grandeza é determinada pelo ciclo principal. Já o intervalo mínimo de inoperância pode ser utilizado para a determinação, por exemplo, do consumo mínimo de carga em uma rede de sensores e também é mostrado que este é determinado pelo ciclo principal.

Outro resultado relativo às medidas de justiça do ERA é referente ao número máximo de *operações independentes*. Esta é uma medida que aproxima o grau de alternância dos nós na operação do ERA. Dados dois nós v e u , esta medida determina o número máximo de vezes que v opera entre duas operações de u e vice-versa. Portanto, um resultado 1 indica que nenhum nó opera duas vezes sem que todos os nós do sistema tenham operado também 1 vez. Já um resultado 3 indica que, no máximo, é possível que um nó opere 3 vezes enquanto outro nó não operou nenhuma vez. Este resultado é obtido diretamente do máximo e mínimo intervalo inoperante e é relevante para aplicações em que se queira medir o grau máximo de desbalanceamento entre diferentes regiões do sistema, por exemplo. Ou seja, onde a coerência entre os momentos em que os nós do sistema operam é importante. Um fator relevante, neste caso, é quando um nó opera muitas vezes enquanto outro nó não opera nenhuma vez, ou quando uma região está ativa, com vários nós operando, enquanto outra região está pouco ativa.

Também é apresentada uma prova de que, quando os nós do ciclo principal operam em intervalos regulares de tamanho d , todo o sistema segue este comportamento. Tal resultado guarda forte relação com o resultado [8] que determina as características das orientações acíclicas que geram execuções do ERA nas quais este comportamento ocorre.

Convergência dos Algoritmos de Inicialização

Certamente, em muitas aplicações, o processo de inicialização do ERA é fator crucial para sua plena utilização. Dos 4 algoritmos desenvolvidos anteriormente para a geração distribuída da orientação acíclica inicial necessária para execução do ERA, Calabrese/França [13], Alg-Viz, Alg-Cor e Alg-Arestas [1][2], destaca-se Alg-Cor como o que gera orientações com maiores concorrências e Alg-Arestas como o de maior velocidade de convergência.

Anteriormente, resultados analíticos sobre o tempo de execução de Alg-Cor eram baseados em alguns resultados preliminares apresentados no trabalho de CALABRESE[13] e em resultados experimentais desenvolvidos em ARANTES[1] e ARANTES[2]. Nesta tese, é mostrado analiticamente que, para grafos completos, o algoritmo tem tempo de convergência sub-exponencial e que, também para grafos completos, quando o número de faces do dado utilizado para o sorteio é igual ao número de nós, o algoritmo é linear.

Foi estabelecida [1][2] uma expressão para a média do tempo de convergência do algoritmo, tendo sido sua acurácia confirmada com simulações. Porém, nos casos em que a convergência rápida do algoritmo, a princípio garantida por Alg-Arestas, é crucial para o sucesso da aplicação, a determinação da média pode não ser garantia suficiente. Por isso, foram aprofundadas as análises da convergência deste algoritmo, através da utilização de uma expressão [17] que permite determinar a probabilidade de um experi-

mento individual se afastar da média prevista. Estes resultados, publicados pelo autor e outros pesquisadores anteriormente [3][4], são parte integrante desta tese e são aqui reproduzidos. Novas análises ainda não publicadas sobre o tempo de convergência de Alg-Arestas foram desenvolvidas a partir de resultados anteriores [1] e são apresentadas no presente trabalho.

1.3 Estrutura da Tese

O restante da tese está dividida da seguinte forma. O Capítulo 2 - Conceitos Básicos resume alguns dos conceitos que são importantes para o entendimento do trabalho. Principalmente aqueles relacionados ao ERA.

O Capítulo 3 - Otimização da Concorrência do ERA apresenta algumas provas de NP-completude para problemas de decisão relacionados aos problemas de encontrar orientações acíclicas que maximizem e minimizem a concorrência na execução do ERA. No caso do problema de decisão relacionado à maximização da concorrência, é apresentada uma prova de NP-completude para grafos com grau máximo 4 e, no caso do problema de decisão relacionado à minimização da concorrência, é apresentada uma prova de NP-completude para grafos cúbicos e planares. O resultado de não-aproximabilidade e o algoritmo polinomial aproximativo também são estabelecidos nesse capítulo.

Para o problema de otimização associado à minimização do inverso da concorrência, damos uma má notícia. Dado um grafo $G = (N, E)$ em geral, é exibida uma prova para a não-aproximação, a menos que $P = NP$, com uma razão menor que $n^{\frac{1}{7}-\varepsilon}$, para todo $\varepsilon > 0$, onde $n = |N|$. A boa notícia que damos é que existe um algoritmo polinomial $\frac{2}{\Delta}$ -aproximativo para a maximização da concorrência se G tem seu grau limitado pelo inteiro positivo constante Δ .

O Capítulo 4 - Trilhas e Propriedades do ERA apresenta um conjunto de conceitos relativos ao ERA, como *trilhas*, *ciclo principal*, *orientação* e *execução canônica*, *execução independente*, além de outras. Neste capítulo são apresentados o máximo e o mínimo intervalos de inoperância e a medida de alternância na operação dos nós. Também é mostrado que, quando os nós dos ciclos principais operam 1 vez em cada d passos, tal comportamento é acompanhado por todo o sistema.

O Capítulo 5 - Análise de Algoritmos Randômicos para Inicialização do ERA aprofunda a análise do tempo de execução dos algoritmos que geram orientações acíclicas para o início da execução do ERA.

No Capítulo 6 - Conclusões e Trabalhos Futuros são apresentadas as conclusões e perspectivas para futuros trabalhos.

Capítulo 2

Conceitos Básicos

2.1 Escalonamento por Reversão de Arestas

2.1.1 Compartilhamento de Recursos

O problema a ser resolvido através do algoritmo *Escalonamento por Reversão de Arestas* (ERA), introduzido em BARBOSA[7] e BARBOSA[8], é o controle da concorrência no acesso a recursos. Genericamente, esse problema consiste em, dado um conjunto P de processadores, um conjunto R de recursos e um conjunto $D \subseteq (P \times R)$ dos pares $(p \in P, r \in R)$ em que o processador p tem acesso ao recurso r , determinar a ordem de execução dos processadores de tal forma que um recurso compartilhado não seja utilizado ao mesmo tempo por dois processadores que o compartilham. Os processadores aqui podem ser não apenas nós de processamento em um sistema distribuído, mas também processos, tarefas, *threads* ou quaisquer entidades computacionais que possam compartilhar recursos.

O ERA é uma solução para o problema de compartilhamento de recursos na situação em que o sistema encontra-se em regime de alta carga. Um

sistema *altamente carregado* é aquele no qual todos os nós requisitam todo o tempo todos os recursos para realizarem a computação a que se destinam, ou seja, para *operar*.

Modelagem em Grafos

Vamos definir um grafo $G = (N, E)$ que modela o sistema, tal que o conjunto P de processadores é o conjunto N de nós do sistema. Além disso, se, para algum recurso r_k , tivermos que $(v_i, r_k) \in D$ e $(v_j, r_k) \in D$, definimos que $(v_i, v_j) \in E$.

Assim, podemos definir que dois nós v e u compartilham um determinado subconjunto de recursos se houver uma aresta (v, u) adjacente a estes. Uma orientação \vec{vu} significa que u tem prioridade de operação em relação a v e, portanto, irá operar antes deste.

Sabendo que todos os nós precisam de todos os recursos, concluímos que um nó, para operar, precisa tornar-se um *sumidouro* (todas as arestas estão orientadas para ele), já que deve ter precedência em relação a todos os vizinhos para tal.

2.1.2 O Algoritmo ERA

O *algoritmo ERA* admite a existência inicial de uma orientação acíclica no grafo. A necessidade de a orientação ser acíclica é facilmente percebida, já que, de acordo com o significado da orientação de uma aresta dada acima, um ciclo implicaria uma cadeia de precedências entre nós resultando em um bloqueio perpétuo ou *deadlock* (ver Figura 2.1). Um bloqueio perpétuo é uma situação na qual um ciclo do grafo tem todas as arestas orientadas no mesmo sentido, fazendo com que todos os nós fiquem sem operar por tempo infinito, uma vez que cada nó tem precedência em relação a um de seus

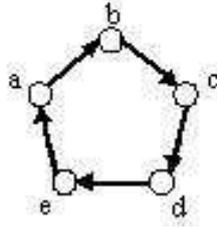


Figura 2.1: Como cada nó tem precedência sobre o vizinho, nenhum irá operar e todos ficarão bloqueados para sempre.

vizinhos, porém, não tem precedência em relação ao outro, formando uma cadeia cíclica e infinita de esperas.

Uma orientação acíclica em um grafo implica a existência de pelo menos um sumidouro. Os nós sumidouros, portanto, serão os primeiros a operar. O algoritmo prossegue da seguinte forma: após os sumidouros operarem, estes revertem todas as suas arestas, de tal forma a dar a precedência para os vizinhos na próxima operação. Esta nova orientação gerada também é acíclica, já que os sumidouros passam a ser fontes e esse tipo de operação não produz um ciclo [8]. Por isso, um outro conjunto de sumidouros será obtido. O processo se repete, definindo o escalonamento na operação dos nós (ver Figura 2.2).

2.1.3 Notações e Propriedades do ERA

A própria descrição da operação do algoritmo ERA deixa claro que este algoritmo não produz *deadlock*. Serão apresentadas de agora em diante diversas características e propriedades do algoritmo, além de um conjunto de notações utilizadas para descrever estas propriedades.

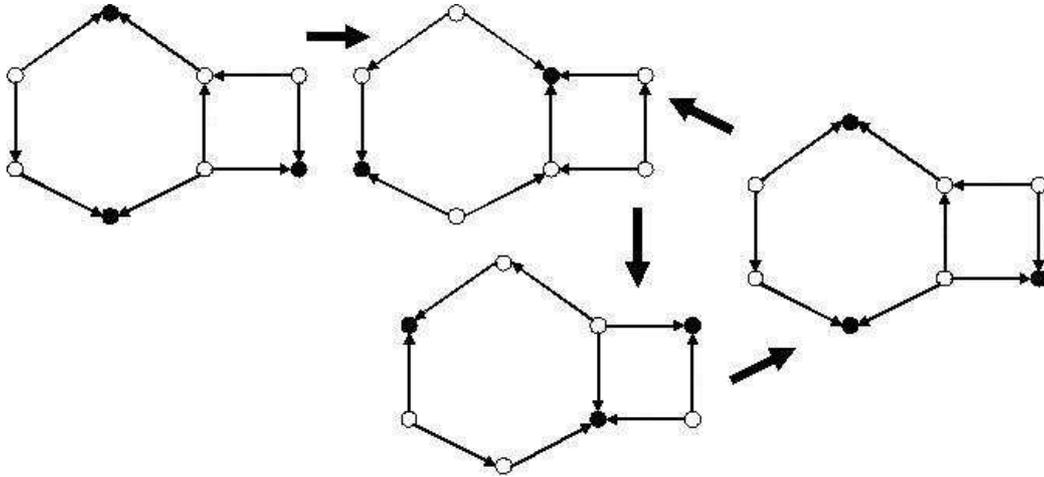


Figura 2.2: Exemplo de execução do ERA.

Primeiramente, $\omega_1, \omega_2, \dots$ denotam a *seqüência de orientações acíclicas* geradas pela operação do algoritmo e formam um *escalonamento* $\sigma(\omega_1)$, que é o escalonamento começando com a orientação acíclica ω_1 . Alternativamente, será utilizada apenas σ para indicar um escalonamento qualquer.

Dizemos, também, que $g(\omega)$ é a orientação acíclica gerada pela reversão das arestas dos sumidouros da orientação acíclica ω . Assim, na seqüência $\omega_1, \omega_2, \omega_3, \dots$, temos $\omega_2 = g(\omega_1)$.

O *conjunto de sumidouros* (*sinks*, em inglês) de uma dada orientação acíclica ω é representado por $sinks(\omega)$. O número de operações de um nó i nas q primeiras orientações de um escalonamento σ (lembre que um nó opera quando se torna um sumidouro) é dado por $m_i(\sigma, q)$.

Com essa notação, demonstramos uma propriedade simples [8], mas que tem importantes implicações:

Lema 2.1. *Considere ω e $\omega' = g(\omega)$. Um sumidouro em ω' tem pelo menos um vizinho que é sumidouro em ω .*

Demonstração. Pela operação do algoritmo, é fácil perceber que as únicas arestas que são modificadas de ω para ω' são as arestas ligadas aos sumidouros de ω . Como os sumidouros de ω' não eram sumidouros em ω , algumas das suas arestas foram modificadas e, portanto, estas arestas eram incidentes a sumidouros em ω . \square

Com isso, podemos mostrar uma propriedade introduzida em BARBOSA[8] que garante a inexistência de *starvation* na operação do ERA.

Teorema 2.2. *Considere dois nós $i, j \in N$ e admita que o menor caminho não orientado que os liga em G contenha r arestas. Temos que $|m_i(\sigma, q) - m_j(\sigma, q)| \leq r$ para todas os escalonamentos σ e todos os $q \geq 1$.*

Demonstração. A prova é por indução em r . Para $r = 1$, i e j são vizinhos e, portanto, pelo lema anterior, se alteram em suas operações, confirmando a propriedade. Para a hipótese da indução, vamos assumir que a propriedade vale para quaisquer par de nós cujo menor caminho não orientado tem, no máximo, $r - 1$ arestas. Assim, dados os nós i e j distantes r arestas entre si (segundo o critério assumido), tomemos um nó k pertencente a esse caminho e que dista d arestas do nó i . Como $d \leq r$, temos

$$|m_i(\sigma, q) - m_k(\sigma, q)| \leq d$$

e $|m_k(\sigma, q) - m_j(\sigma, q)| \leq r - d$ e, assim, como queríamos provar,

$$|m_i(\sigma, q) - m_j(\sigma, q)| \leq r.$$

\square

O número de orientações acíclicas de um grafo é obviamente finito, portanto um escalonamento $\sigma = (\omega_1, \omega_2, \dots)$ entra, em algum momento, em

período repetitivo de comprimento p . A isto, segue uma propriedade [8] que garante uma classe de justiça provida pelo algoritmo:

Corolário 2.3. *O número de vezes que um nó opera em um período é igual para todos os nós.*

Demonstração. Suponha que temos um período que é composto de um número p de orientações acíclicas, temos $m_i(\sigma, p) = k_1$ e $m_j(\sigma, p) = k_2$, $k_1 \neq k_2$ para dois nós i e j quaisquer. Conforme os períodos se repetem, a diferença entre o número de vezes que i e j operam aumentará indefinidamente, contradizendo o teorema anterior. Portanto, $k_1 = k_2$. \square

É fácil perceber que todo o escalonamento produzido pelo ERA e, conseqüentemente, todas as características deste são determinadas pela primeira orientação. Consideremos, portanto $m(\omega_1)$ o número de vezes que qualquer nó opera no período formado a partir de ω_1 (observe que ω_1 não obrigatoriamente participa do período) e $p(\omega_1)$ o número de orientações diferentes no mesmo período. Sabe-se que, segundo o corolário anterior, $m(\omega_1)$ é igual para todos os nós.

Passamos a estudar uma medida da "quantidade de concorrência" propiciada pelo ERA. Essa medida é importante já que, quanto maior a concorrência do sistema, mais eficientemente os recursos compartilhados serão utilizados. Uma maneira razoável de mensurar essa grandeza nos k primeiros passos do escalonamento e que representamos por $\gamma(\sigma, k)$ é a fração média do total de nós que opera em cada orientação acíclica gerada, ou seja:

$$\gamma(\sigma, k) = \frac{1}{nk} \sum_{v_i \in N} m_i(\sigma, k)$$

No limite, temos a medida de concorrência de todo o escalonamento. Esta pode ser escrita como:

$$\gamma(\sigma) = \lim_{k \rightarrow +\infty} \frac{1}{nk} \sum_{v_i \in N} m_i(\sigma, k)$$

Poderíamos, também, dado que $\sigma = (\omega_1, \omega_2, \dots)$, escrever $\gamma(\omega_1)$ em vez de $\gamma(\sigma)$, já que a primeira orientação determina todo o escalonamento posterior. Da mesma forma, podemos escrever outras grandezas em função de ω_1 , como $m_i(\omega_1, k)$, por exemplo. É isso que faremos na maior parte das vezes daqui para frente.

A seguir, apresentamos uma expressão para o limite da concorrência [8].

Teorema 2.4. $\gamma(\omega_1) = \frac{m(w_1)}{p(w_1)}$.

Demonstração. Considere que, para algum $l \geq 1$, ω_l é a primeira orientação acíclica que participa do período produzido a partir de ω_1 . Podemos reescrever a expressão como a média ponderada das concorrências em diferentes seqüências do escalonamento (até ω_l e a partir de ω_{l+1}):

$$\gamma(\omega_1) = \lim_{k \rightarrow \infty} \frac{l \cdot \gamma(\sigma, l) + (k - l) \cdot \gamma(\sigma)}{k}$$

Essa expressão é obviamente dominada pela segunda parte e é igual a:

$$\gamma(\omega_1) = \lim_{k \rightarrow \infty} \frac{1}{n(k-l)} \sum_{v_i \in N} m_i(\omega_l, k-l)$$

Como ω_l já é parte do período, temos

$$\gamma(\omega_1) = \frac{m(w_l)}{p(w_l)} = \frac{m(w_1)}{p(w_1)}.$$

□

A seguinte relação é facilmente obtida a partir da observação de que, no melhor caso, um nó só pode operar uma vez a cada 2 passos e de que, no pior caso, teríamos apenas um nó operando por vez no período:

$$\frac{1}{n} \leq \gamma(\omega_1) \leq \frac{1}{2}$$

Em BARBOSA[8], é mostrado também o seguinte teorema que, aqui, deixamos sem prova:

Teorema 2.5. *Se G é uma árvore então $\gamma(\omega_1) = \frac{1}{2}$.*

Surpreendentemente, quando G não é uma árvore, existe uma expressão fechada para a concorrência na operação do ERA em função de uma propriedade da orientação acíclica inicial. A surpresa advém do fato de o funcionamento de um sistema altamente dinâmico como o ERA poder ser expresso a partir de uma característica estática de um grafo. A prova para essa expressão está longe de ser simples e pode igualmente ser obtida em BARBOSA[8].

Se G não é uma árvore, tomemos um ciclo simples não orientado κ (ciclos simples são, nesse caso, aqueles que passam, no máximo, uma vez por cada nó). Escrevemos $|\kappa|$, o número de nós do ciclo. Após aplicada a orientação ω_1 no grafo, denotamos por $n^+(\kappa, \omega_1)$ e $n^-(\kappa, \omega_1)$ o número de arestas, em κ , orientadas na direção *horária* e *anti-horária*, respectivamente. Definimos

$$\zeta(\kappa, \omega_1) = \min \left\{ \frac{n^+(\kappa, \omega_1)}{|\kappa|}, \frac{n^-(\kappa, \omega_1)}{|\kappa|} \right\}$$

e denotamos K como sendo o conjunto de todos os ciclos simples não direcionados do grafo. Assim, segue-se o seguinte teorema [8]:

Teorema 2.6. *Dado um grafo G orientado inicialmente por ω_1 , a concorrência do escalonamento gerado pela operação do ERA é dada pela expressão:*

$$\gamma(\omega_1) = \min_{\kappa \in K} \zeta(\kappa, \omega_1)$$

Ciclo Principal

No trabalho que se segue, muitas vezes será utilizado o conceito de ciclo principal, principalmente no Capítulo 4 - Trilhas e Propriedades do ERA. Aqui, citamos rapidamente este conceito, já conhecido anteriormente na literatura [19]. Basicamente, os ciclos principais são aqueles que satisfazem o mínimo da fórmula $\gamma(\omega_1) = \min_{\kappa \in K} \zeta(\kappa, \omega_1)$, ou seja, os ciclos $\kappa^* \in K$ tais que $\zeta(\kappa^*, \omega_1) = \min_{\kappa \in K} \zeta(\kappa, \omega_1)$. Será mostrado no Capítulo 4 - Trilhas e Propriedades do ERA que, além de definir a concorrência do sistema, os ciclos principais também determinam outras grandezas na execução do ERA, modulando o sistema como um todo.

Decomposição por Sumidouros

Dado um grafo $G = (N, E)$ orientado aciclicamente, a *decomposição por sumidouros* (ou *sink decomposition*) é uma partição de N em conjuntos disjuntos S_0, \dots, S_λ de tal forma que $v_i \in S_k$, $0 \leq k \leq \lambda$, se e somente se o maior caminho direcionado de v_i até um sumidouro tem k arestas. Estes conjuntos podem ser facilmente construídos da seguinte forma: em S_0 , insira todos os sumidouros da orientação acíclica inicial e, depois, retire-os do grafo, juntamente com todas as arestas incidentes a eles. Do grafo que resta, insira todos os sumidouros agora existentes em S_1 , retirando-os do grafo junto com suas arestas. O processo termina quando S_λ é preenchido e o grafo corrente não tenha vértices.

Definimos também a função $\lambda(G)$ como sendo o maior nível da decomposição por sumidouros existente no grafo. Como, para haver decomposição por sumidouros, G precisa estar orientado aciclicamente, também podemos nos referir, quando se fizer necessário, a $\lambda(\alpha)$ para uma orientação acíclica α qualquer. Neste caso, o grafo G fica subtendido.

Três propriedades importantes de uma decomposição por sumidouros:

- (1) não há vizinhos dentro de nenhum $S_k, 0 \leq k \leq \lambda$;
- (2) para $1 \leq k \leq \lambda$, qualquer nó em S_k tem, pelo menos, um vizinho em S_{k-1} .
- (3) considerando uma execução do ERA com escalonamento dado pelas orientações acíclicas $\alpha_0, \alpha_1, \dots$ temos que $\lambda(\alpha_{i+1}) \leq \lambda(\alpha_i)$, para todo $i \geq 0$.

Capítulo 3

Otimização da Concorrência do ERA

3.1 Introdução

Neste capítulo são apresentadas algumas provas e questões sobre o problema de encontrar concorrências máximas e mínimas para o ERA.

Em BARBOSA[8] foi provado que, dado um grafo $G = (N, E)$ e um racional r , decidir se G admite uma orientação acíclica ω para a qual a concorrência $\gamma(G, \omega) \geq r$ é NP-completo. Para tanto, foi utilizada uma redução a partir do problema de coloração. Outras questões, porém, podem ser levantadas na tentativa de caracterizar melhor a complexidade do problema. Por exemplo: qual é o mínimo grau máximo de $G = (N, E)$ que garante a NP-completude?

Neste capítulo é apresentada uma demonstração de que o problema de decisão relacionado ao problema de encontrar a concorrência mínima é NP-completa mesmo para grafos cúbicos e planares e, em geral, para todas as restrições da prova de NP-completude do problema de decisão do Ciclo Hamil-

toniano [16]. Posteriormente, são apresentadas possíveis aplicações para o problema de encontrar a concorrência mínima do ERA. Nas seções seguintes é apresentada uma prova de que a NP-completude para o problema de encontrar a concorrência máxima do ERA se mantém mesmo para grafos com grau máximo 4. É também apresentada uma prova de que o problema de minimização do inverso da concorrência não pode ser aproximado por uma razão menor do que $n^{\frac{1}{7}-\varepsilon}$, para qualquer $\varepsilon > 0$. Finalmente, é apresentado um algoritmo polinomial aproximativo com desempenho $\frac{2}{\Delta}$ para o problema da maximização da concorrência.

Antes, porém, são introduzidas algumas definições que serão importantes para as provas das seções seguintes.

3.2 Definições

Definição 3.1. *Ciclo Hamiltoniano* - É um ciclo que percorre todos os nós de um grafo, passando por cada nó uma única vez.

Definição 3.2. *HAM - Ciclo Hamiltoniano - Problema de decisão*

Instância: Grafo $G = (N, E)$ cúbico planar.

Questão: Existe um ciclo hamiltoniano para o grafo G ?

Definição 3.3. *CONMIN - Concorrência Mínima- Problema de decisão*

Instância: Grafo $G = (N, E)$ cúbico planar e número racional α .

Questão: Existe uma orientação acíclica ω em G tal que $\gamma(G, \omega) \leq \alpha$?

Definição 3.4. *MAXCONMAX - Maximização da Concorrência - Problema de otimização*

Instância: $G = (N, E)$.

Objetivo: Encontrar uma orientação acíclica ω em G tal que esta maximize o valor de $\gamma(\omega) = \frac{m}{p}$.

Definição 3.5. *CONMAX - Concorrência Máxima - Problema de decisão*

Instância: Grafo $G = (N, E)$ e número racional α .

Questão: Existe uma orientação acíclica ω em G tal que $\gamma(G, \omega) \geq \alpha$?

Definição 3.6. *MININCON - Minimização do Inverso da Concorrência - Problema de otimização*

Instância: $G = (N, E)$.

Objetivo: Encontrar uma orientação acíclica ω em G tal que esta minimize o valor de $\gamma(\omega)^{-1} = \frac{p}{m}$.

Definição 3.7. *NAEQ3SAT - NOT ALL EQUAL 3 SAT - Problema de decisão*

Instância: Conjunto U de variáveis booleanas, coleção C de cláusulas concatenadas no formato $c_1 \wedge c_2 \dots$, cada cláusula utilizando literais sobre as variáveis de U no formato $(l_1 \vee l_2 \vee l_3)$ (onde um literal é a ocorrência de uma variável negada ou não), sendo que $\forall c \in C$, temos que $|c| = 3$ literais.

Questão: Existe uma atribuição de verdade para as variáveis de U tal que, para cada cláusula de C , exista pelo menos 1 literal verdadeiro e um falso?

Definição 3.8. *NAEQ3SAT₃ - NOT ALL EQUAL 3 SAT COM EXATAMENTE 3 OCORRÊNCIAS POR VARIÁVEL - Problema de decisão*

Instância: Conjunto U de variáveis booleanas, coleção C de cláusulas concatenadas no formato $c_1 \wedge c_2 \dots$, cada cláusula utilizando literais sobre as variáveis de U no formato $(l_1 \vee l_2 \dots)$ (onde um literal é a ocorrência de uma variável negada ou não), sendo que $\forall c \in C$, temos que $|c| = 2$ ou 3 literais, cada variável $u \in U$ ocorre exatamente 3 vezes em C , cada literal ocorre pelo menos 1 vez em C e cada cláusula de tamanho 2 contém pelo menos um literal que não ocorre nenhuma outra vez em C (ou seja, cada cláusula de tamanho 2 contém um literal tal que as outras duas ocorrências da variável deste literal em C aparecem como uma negação deste).

Questão: Existe uma atribuição de verdade para as variáveis de U tal que o resultado total da expressão seja verdadeiro e tal que, para cada cláusula de C , exista pelo menos 1 literal verdadeiro e um falso?

Definição 3.9. *Dado um problema de decisão Π e uma instância I de Π , denota-se $Opt_{\Pi}(I)$ como sendo o problema de otimização correspondente. Além disso, dada uma instância I do problema, denota-se $Opt_{\Pi}(I)$ como sendo o custo da solução ótima do problema para a particular instância I .*

3.3 NP-Completeness da Concorrência Mínima

Nesta seção será apresentada uma prova de que CONMIN é um problema NP-completo para grafos cúbicos planares. A prova se dará através da redução a partir de HAM.

Teorema 3.1. *CONMIN é um problema NP-completo para grafos cúbicos planares.*

Demonstração. A prova se dará através de uma redução de HAM, que é um problema NP-completo para grafos cúbicos planares [16]. É possível verificar que CONMIN está em NP, já que o certificado de solução SIM para o problema é um grafo orientado para o qual é possível verificar a concorrência em $O(n^6)$ com o algoritmo apresentado em BARBOSA[8], tornando possível a validação do certificado em tempo polinomial no tamanho da instância.

Dado $G = (N, E)$ uma instância de HAM, é possível formar uma instância de CONMIN considerando o mesmo grafo G e tomando-se $\alpha = \frac{1}{n}$, onde $n = |N|$. Agora, passamos a provar que G possui um ciclo Hamiltoniano se, e somente se, G admite uma orientação acíclica com concorrência menor ou igual a α . Primeiramente, considerando-se uma solução para CONMIN com

as entradas G e $\frac{1}{n}$, não podemos ter $\gamma(G, \omega) < 1/n$, já que, no mínimo, um dos nós opera por cada passo. Além disso, $\gamma(G, \omega) = 1/n$, se, e somente se, existem κ e ω em G (um ciclo simples e uma orientação acíclica, respectivamente) de tal forma que $|\kappa| = n$ e das duas uma: ou $n_{cw}(\kappa, \omega) = 1$ ou $n_{ccw}(\kappa, \omega) = 1$. Como, para que isso aconteça, κ deverá ser um ciclo simples que passa por todos os nós do grafo, então temos que existe um ciclo hamiltoniano em G . Por outro lado, considerando uma solução de HAM para G , então existe um ciclo simples que passa por todos os nós do grafo. Neste caso, é possível construir uma orientação ω da seguinte forma: assumamos κ como o ciclo hamiltoniano do grafo G composto, em ordem de vizinhança, pelos nós v_0, v_1, \dots, v_n e oriente cada aresta na direção do nó de menor índice para o nó de maior índice. Teremos uma orientação acíclica ω e, além disso, o ciclo κ formado pelas arestas $(v_0, v_1), (v_1, v_2), \dots, (v_{n-1}, v_n), (v_n, v_0)$ é tal que, sem perda de generalidade, $n_{cw}(\kappa, \omega) = 1$ e $|\kappa| = n$ e, portanto, $\gamma(G, \omega) = 1/n$. \square

3.4 NP-Completude da Concorrência Máxima para Grafos com Grau no Máximo 4

A NP-completude do problema de encontrar a concorrência máxima para grafos foi mostrada em BARBOSA[8]. Nesta seção, é apresentada uma prova de que a NP-completude para este problema se mantém mesmo no caso mais restrito dos grafos em geral com grau máximo 4. Deve-se lembrar, porém, a existência de algoritmos polinomiais para encontrar a concorrência máxima em algumas famílias de grafos: árvores e grafos bipartites ($Opt_{CONMAX}(G) = \frac{1}{2}$); grafos completos ($Opt_{CONMAX}(G) = \frac{1}{n}$); ciclos ($Opt_{CONMAX}(G) = \frac{\lfloor \frac{n}{2} \rfloor}{n}$). Também é conhecido que, se G apresenta um

clique K_s de tamanho s , qualquer orientação ω resultará em uma concorrência $\frac{m}{p} \leq \frac{1}{s}$. A prova da NP-completude para a classe de grafos com grau máximo 4 se dá através de uma redução a partir do problema NAEQ3SAT_3 para o problema CONMAX . Esta redução é tal que, em tempo polinomial no tamanho de G , obtém-se, a partir de uma instância $I = (U, C)$ de NAEQ3SAT_3 , uma instância de CONMAX $G = (N, E)$ com grau no máximo 4 e constantes $m = 1$ e $p = 3$ tal que I é satisfatível se, e somente se, o grafo G admite uma orientação acíclica ω para a qual $\gamma(G, \omega) = \frac{m}{p}$. Sabe-se por SCHAEFER[23] que NAEQ3SAT é NP-completo. Provamos primeiramente que NAEQ3SAT_3 é NP-completo e, depois, fazemos a redução de CONMAX para provar sua NP-completude.

3.4.1 NP-completude de NAEQ3SAT_3

Lema 3.2. *NAEQ3SAT_3 é NP-completo.*

Demonstração. Primeiramente, temos que o problema está em NP porque NAEQ3SAT está em NP . Tome $I = (U, C)$ uma instância do problema NP-completo NAEQ3SAT . É possível gerar, em tempo polinomial no tamanho de I , uma instância $I' = (U', C')$ de NAEQ3SAT_3 tal que I é satisfatível se, e somente se, I' é satisfatível e vice-versa. O processo inicializa $U' := U$ e $C' := C$ e prossegue alterando U' e C' de acordo com o número de ocorrências de cada variável u em C . Assuma u como sendo uma variável de U que ocorre $k = k_1 + k_2$ vezes em C , onde k_1 e k_2 são o número de vezes em que o literal correspondente de u é positivo e negativo, respectivamente. Serão considerados três casos. Os casos (1) e (2) ocorrem quando $k \leq 2$ e, nestes, I' é ajustado de tal forma que u apareça três vezes em C' . O caso (3) ocorre quando $k \geq 3$ e, neste caso, a excessiva ocorrência de uma variável é ajustada através da transformação de cada literal desta em literais de outras variáveis

diferentes, de tal forma que os novos literais cumprirão não apenas a exigência do número de ocorrências, mas também assumirão obrigatoriamente, para satisfazer ao problema, os mesmos valores que possuíam com a variável que lhes originou. As variáveis destes literais terão também o mesmo valor da variável geradora inicial na hipótese de satisfabilidade. Seguem as regras de transformação.

- (i) ($k = 1$) - Adicionar a cláusula $(u \vee \bar{u})$ a C' .
- (ii) ($k = 2$) - Ocorrem 3 casos:
 - (a) ($k_1 = 2$) - Adicionar u_1 a U' e $(u_1 \vee \bar{u})$, $(u_1 \vee \bar{u}_1)$ a C' .
 - (b) ($k_2 = 2$) - Adicionar u_1 a U' e $(\bar{u}_1 \vee u)$, $(u_1 \vee \bar{u}_1)$ a C' .
 - (c) ($k_1 = k_2 = 1$) - Adicionar u_1 e u_2 a U' e $(u_1 \vee \bar{u} \vee u_2)$, $(u_1 \vee \bar{u}_2)$ e $(\bar{u}_1 \vee u_2)$ a C' .
- (iii) ($k_1 \geq 3$ ou $k_2 \geq 3$) - Substituir as k_1 ocorrências de u por novos literais u_1, u_2, \dots, u_{k_1} e as k_2 ocorrências de \bar{u} por novos literais v_1, v_2, \dots, v_{k_2} . Também adicionar 2 novos conjuntos de variáveis $T = \{t_1, \dots, t_{k_1}\}$ e $W = \{w_1, \dots, w_{k_2}\}$ a U' formando cláusulas de acordo com as instruções abaixo:

- (a) Se u ocorre positivamente, adicionar a C' a coleção de cláusulas a seguir:

ordem de ocorrência de u	cláusulas a serem adicionadas
1	$(u_1 \vee \bar{t}_1 \vee t_2), (\bar{u}_1 \vee t_2)$
2	$(u_2 \vee \bar{t}_2 \vee t_3), (\bar{u}_2 \vee t_3)$
3	$(u_3 \vee \bar{t}_3 \vee t_4), (\bar{u}_3 \vee t_4)$
\vdots	\vdots
$k_1 - 1$	$(u_{k_1-1} \vee \bar{t}_{k_1-1} \vee t_{k_1}), (\bar{u}_{k_1-1} \vee t_{k_1})$

- (b) Se u ocorre negativamente, adicionar a C' a coleção de cláusulas a seguir:

ordem de ocorrência de u	cláusulas a serem adicionadas
1	$(\overline{v_1} \vee w_1 \vee \overline{w_2}), (v_1 \vee \overline{w_2})$
2	$(\overline{v_2} \vee w_2 \vee \overline{w_3}), (v_2 \vee \overline{w_3})$
3	$(\overline{v_3} \vee w_3 \vee \overline{w_4}), (v_3 \vee \overline{w_4})$
\vdots	\vdots
$k_2 - 1$	$(\overline{v_{k_2-1}} \vee w_{k_2-1} \vee \overline{w_{k_2}}), (v_{k_2-1} \vee \overline{w_{k_2}})$

- (c) As cláusulas seguintes devem ser adicionadas de acordo com o caso:

- i. ($k_2 = 0$) - Adicionar $(u_{k_1} \vee \overline{t_{k_1}} \vee t_1), (\overline{u_{k_1}} \vee t_1)$ a C' .
- ii. ($k_1 = 0$) - Adicionar $(\overline{v_{k_2}} \vee w_{k_2} \vee \overline{w_1}), (v_{k_2} \vee \overline{w_1})$ a C' .
- iii. ($k_1 \neq 0$ e $k_2 \neq 0$) - Adicionar $(u_{k_1} \vee \overline{t_{k_1}} \vee t_1), (\overline{u_{k_1}} \vee w_1), (\overline{v_{k_2}} \vee w_{k_2} \vee \overline{w_1}), (v_{k_2} \vee \overline{t_1})$ a C' .

Está concluída a construção da instância $I' = (U', C')$ de NAEQ3SAT_3 . Suponha que $I = (U, C)$ é uma instância satisfatível de NAEQ3SAT . Vamos estender a atribuição de verdades de U na solução para as variáveis correspondentes de U' . Antes, definimos o conjunto $C' \pitchfork C$ como sendo o conjunto das cláusulas de C' iguais a alguma cláusula de C ou geradas pela troca direta de variáveis de alguma cláusulas de C . Definimos também o conjunto $C' \ominus C = C' - (C' \pitchfork C)$ das cláusulas adicionadas a parte em C' em função das variáveis de U . Desta forma, dada η uma atribuição de verdades a U que satisfaz I , é óbvio notar que as cláusulas do conjunto $C' \pitchfork C$ serão igualmente satisfatíveis, bastando apenas manter o valor das variáveis preservadas e atribuir a cada variável u' em $c' \in C' \pitchfork C$, substituta de u em $c \in C$, o mesmo valor de u em η . Para o conjunto de cláusulas $C' \ominus C$, pode-se notar

facilmente que a atribuição do mesmo valor a todas as variáveis geradas, segundo as regras definidas, a partir de u satisfaz todas as cláusulas de $C' \ominus C$. Assim, verificou-se que a uma solução viável de I corresponde sempre uma solução viável de I' . Fazendo o caminho inverso, considere $I' = (U', C')$ uma instância satisfatível de NAEQ3SAT₃ segundo uma atribuição de verdade η para U' . Pode-se provar que, para ser satisfatível, dada uma variável u de U , η deverá atribuir o mesmo valor a todas as variáveis geradas a partir desta na construção da instância I' a partir de I . Tal propriedade é facilmente verificável para as variáveis geradas nos casos (1) e (2) descritos acima. No caso (3), pode-se perceber que, nas cláusulas dos tipos $(\overline{u}_i \vee t_{i+1})$ (incluídas em (a)) e $(v_i \vee \overline{w_{i+1}})$ (incluídas em (b)), ambas as variáveis deverão possuir o mesmo valor verdade, já que a solução só é admitida se cada cláusula possuir, pelo menos 1 literal com valor verdadeiro e outro com valor falso. Por isso, nas cláusulas dos tipos $(u_i \vee \overline{t}_i \vee t_{i+1})$ (incluídas também em (a)) e $(\overline{v}_i \vee w_i \vee \overline{w_{i+1}})$ (incluídas também em (b)), as variáveis também terão os mesmos valores já assumidos por conta das cláusulas duplas anteriormente citadas, já que, nestas cláusulas, dois literais já terão o mesmo valor. Como as variáveis t_i e w_i são utilizadas em uma linha e na linha seguinte das tabelas acima, ocorre que todas as linhas conterão variáveis com o mesmo valor. Para entender que as variáveis adicionadas em (a) e em (b) terão o mesmo valor, basta verificar as cláusulas incluídas em (c), que fazem a ligação entre os conjuntos de variáveis incluídas. Mais ainda, as outras 2 ocorrências das variáveis u_i ou v_i em cláusulas de tamanho 2 são de negações destas. Por conta desta propriedade, portanto, é possível atribuir o valor das variáveis substituídas da variável original, desta forma, construindo a solução de NAEQ3SAT a partir de NAEQ3SAT₃ e completando a prova. \square

3.4.2 Construção de uma Instância de CONMAX a partir de uma Instância de NAEQ3SAT₃

Nesta seção pretende-se provar que, para uma instância $I = (U, C)$ qualquer de NAEQ3SAT₃, é possível construir, em tempo polinomial no tamanho de I , uma instância $G = (N, E)$ correspondente de CONMAX. Assuma $U = \{u_1, u_2, \dots, u_n\}$ e $C = \{c_1, c_2, \dots, c_m\}$, respectivamente, como o conjunto de variáveis e a coleção de cláusulas de I . O grafo G a ser construído contém 2 tipos de subgrafos que se relacionam com U e C : os subgrafos *Truth Setting* T_i , que formam o conjunto $T_* = \bigcup_{i=1}^n T_i$ e correspondem às variáveis definidas em U ; e os subgrafos *Satisfaction Testing* S_j , que formam o conjunto $S_* = \bigcup_{j=1}^m S_j$ e correspondem às cláusulas definidas em C . Estes estão destacados na Figura 3.1(a) e (b).

Cada variável u_i , $i \in \{1, 2, \dots, n\}$, de U apresenta um subgrafo correspondente T_i de G formado pelos nós u_i , \bar{u}_i , a_i , b_i e d_i , como indicado na Figura 3.1(a). Observe que os dois primeiros nós desta lista correspondem aos literais da variável correspondente. Assim, tem-se que T_* possui um nó correspondente a cada literal das variáveis de U . Além disso, cada cláusula c_j , $j \in \{1, 2, \dots, m\}$, de C apresenta um subgrafo correspondente S_j de G , sendo este um subgrafo completo de 3 nós, indicado na Figura 3.1(b). Há ainda um conjunto adicional de nós $W = \{w_1, w_2, \dots, w_{n-1}\}$ e um conjunto de arestas $E_W = \{w_i a_i, w_i b_i, w_i a_{i+1}, w_i b_{i+1} : i \in 1, 2, \dots, n-1\}$ que partem dos nós de W .

As arestas adicionadas agora são a única parte da construção que depende de quais cláusulas contêm quais literais. Para cada literal l de cada cláusula $c_j \in C$, seleciona-se um nó x do subgrafo correspondente S_j que ainda não apresente conexões com nós de T_* e adiciona-se uma aresta xy , onde y é o nó correspondente ao literal l em T_* . Se c_j possui apenas dois literais e

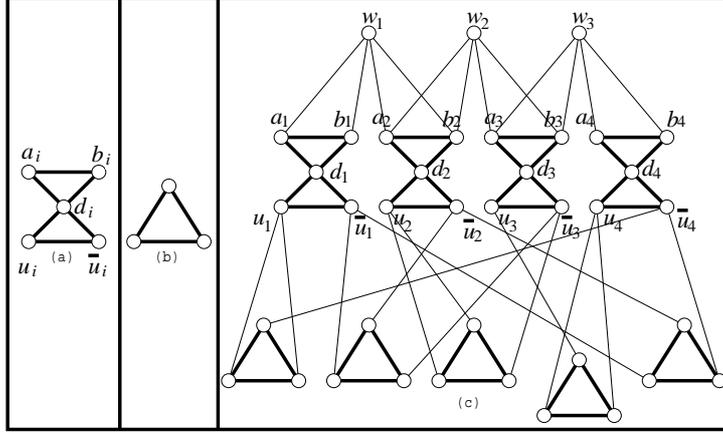


Figura 3.1: Subgrafos Truth Setting (a) e Satisfaction Testing (b). (c) apresenta instância $G = (N, E)$ de $CONMAX$ obtido da instância de $NAEQ3SAT_3$ $I = (U, C) = (\{u_1, u_2, u_3, u_4\}, \{(u_1 \vee \bar{u}_4), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_3), (u_2 \vee \bar{u}_3), (u_3 \vee u_4), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_4)\})$.

x é o nó de S_j ainda não conectado a um nó em T_* , então a aresta xy é adicionada a E , de tal forma que y é o nó de T_* correspondente a um literal de c_j que ocorre apenas uma vez, havendo a chance de serem válidas ligações a qualquer dos dois literais em c_j , dado que pode ocorrer de ambos serem literais únicos. Isto conclui a construção da instância especial de $CONMAX$. A Figura 3.1(c) mostra um exemplo de uma instância especial de G obtida a partir da seguinte instância satisfatível de $NAEQ3SAT_3$: $I = (U, C) = (\{u_1, u_2, u_3, u_4\}, \{(u_1 \vee \bar{u}_4), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_3), (u_2 \vee \bar{u}_3), (u_3 \vee u_4), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_4)\})$.

Devemos primeiramente garantir que o processo descrito produz um grafo cujo grau máximo é 4. O lema abaixo garante essa propriedade.

Lema 3.3. *Considere o grafo $G = (N, E)$. Se G pode ser obtido a partir de uma instância $I = (U, C)$ de $NAEQ3SAT_3$ segundo a construção descrita acima, então o grau máximo de G é 4.*

Demonstração. Para os nós contidos nos subgrafos S_j , $1 \leq j \leq m$, temos que a verificação é óbvia a partir da constatação de que cada nó de S_j é adjacente a 2 arestas que formam o triângulo S_j e a uma aresta que o liga a um nó de T_* . Para os nós do conjunto W , a verificação também é imediata, pela própria construção das arestas adjacentes a estes, descritas em E_W , formando 4 arestas para cada nó w_i , $1 \leq i < n$. Os nós d_i são adjacentes às arestas que os ligam aos nós a_i, b_i, u_i, \bar{u}_i , formando também 4 arestas. Os nós a_i e b_i são adjacentes às arestas que os ligam aos nós w_i, a_i ou b_i , conforme o caso, e d_i , formando 3 arestas por nó. Já os nós que representam os literais têm 2 arestas fixas, uma adjacente ao literal de mesma variável barrado (de u_i a \bar{u}_i e vice-versa) e uma aresta a d_i . Portanto, restam as arestas que conectam os literais aos nós de S_* e estas devem ser, no máximo 2. Esta condição é garantida pelo fato de que cada literal é conectado a um nó de um subgrafo S_j qualquer quando está contido na cláusula C_j correspondente e, por premissa, sabemos que cada literal aparece no máximo 2 vezes em C . E, no caso de cláusulas de tamanho 2, o literal conectado 2 vezes é aquele que aparece apenas mais uma vez em C , fazendo com que nunca um nó u_i ou \bar{u}_i apresente mais que 4 arestas. \square

3.4.3 Prova da NP-Compleitude de CONMAX

Para realizarmos a prova final, precisamos de alguns resultados intermediários que seguem. O Lema 3.4 é utilizado adiante e afirma que, em um grafo com concorrência $\frac{1}{3}$, se todos os nós pertencem a triângulos, então o período é igual a 3 e cada triângulo tem sempre um nó sumidouro em cada orientação.

Lema 3.4. *Assuma $G = (N, E)$ um grafo, tal que existe uma orientação acíclica ω com concorrência $\gamma(\omega) = \frac{m}{p} = \frac{1}{3}$. Se, para todo $x \in N$, existe um*

triângulo T , subgrafo de G contendo x , então, em todo passo na operação do ERA, existe exatamente um nó do triângulo T operando. Além disso, pode-se afirmar que $m = 1$ e $p = 3$, ou seja, o período é igual a 3 e cada nó opera exatamente 1 vez.

Demonstração. Assuma $\omega_1, \omega_2, \dots, \omega_p$ como as orientações acíclicas consecutivas do ERA onde ω_j é obtido pela reversão das arestas do conjunto de sumidouros de ω_{j-1} , $j \in \{2, 3, \dots, p\}$ e ω_1 é obtido pela reversão das arestas de ω_p . Dado um número inteiro positivo i e $v \in N$, chamamos de t_v^i o índice da orientação $\omega_{(t_v^i)}$ onde v opera pela i -ésima vez. Dados os nós x, y e z pertencentes a um triângulo T , temos claramente que $t_x^{i+1} \geq t_x^i + 3$, o que equivale, para um inteiro $\delta_x^i > 0$, a $t_x^{i+1} = t_x^i + 3 + \delta_x^i$. Como, pelo enunciado, $m = \alpha$ e $p = 3\alpha$, α inteiro positivo, temos que um período começado em $\omega_{t_x^i}$ é finalizado em $\omega_{t_x^{i+\alpha}-1}$. Logo é claro que, pelo enunciado, $\frac{\alpha}{t_x^{i+\alpha}-t_x^i} = \frac{1}{3}$ e, portanto, $t_x^{i+\alpha}-t_x^i = 3\alpha$. Observe que $t_x^{i+\alpha}-t_x^i = \sum_{l=i}^{i+\alpha-1} t_x^{l+1}-t_x^l = \sum_{l=1}^{\alpha} t_x^{l+1}-t_x^l$ (a segunda igualdade é verdadeira por conta do comportamento periódico). Porém, já que $t_x^{i+1} = t_x^i + 3 + \delta_x^{i+1}$, fazendo a substituição, temos que $t_x^{i+\alpha} - t_x^i = \sum_{l=1}^{\alpha} t_x^{l+1} - t_x^l = \sum_{l=1}^{\alpha} (t_x^l + 3 + \delta_x^l) - t_x^l = 3\alpha + \sum_{l=1}^{\alpha} \delta_x^l = 3\alpha$. Desta forma, temos que $\delta_x^i = 0$, para todo valor de i , o que significa que $t_x^{i+1} = t_x^i + 3 + \delta_x^i = t_x^i + 3$. Assim, fica claro que todo nó opera sempre a cada 3 passos do ERA. Desta forma, a orientação acíclica repete-se a cada 3 passos, garantindo que $m = 1$ e $p = 3$. \square

O Lema 3.5 mostra que, numa execução do ERA com concorrência $\frac{1}{3}$, o grafo construído apresenta a propriedade de os nós de W e os nós d_i de T_i , $i \in \{1, 2, \dots, n\}$, operarem sempre ao mesmo tempo.

Lema 3.5. *Considere um grafo $G = (N, E)$, como definido, a partir de uma instância I de $NAEQ3SAT_3$ e aciclicamente orientado segundo ω . Se*

a concorrência da orientação ω de G é $\gamma(\omega) = \frac{1}{3}$, temos que todos os nós $w_i \in W$, $i \in \{1, 2, \dots, n-1\}$, e $d_j \in T_j$, $j \in \{1, 2, \dots, m\}$, operam no mesmo passo.

Demonstração. Considere ω_1 , ω_2 e ω_3 como sendo orientações acíclicas consecutivas que formam o período na execução do ERA. Assuma que um nó $w_i \in W$, $i \in \{1, 2, \dots, n-1\}$, opera em ω_1 . Assuma por um momento que w_{i+1} não é um sumidouro em ω_1 . Então, nem a_{i+1} nem b_{i+1} são sumidouros em ω_1 . Se assim for, o triângulo formado pelos vértices a_{i+1} , b_{i+1} e w_{i+1} contradiz o Lema 3.4. Da mesma forma, se a_{i+1} , b_{i+1} não operam neste passo, então, pelo mesmo Lema 3.4, d_{i+1} obrigatoriamente estará operando em ω_1 . \square

Agora, a prova final.

Teorema 3.6. *CONMAX é NP-completo para grafos com grau máximo 4.*

Demonstração. Dado um grafo qualquer $G = (N, E)$ onde $n = |N|$ e uma orientação acíclica ω de G , existe em BARBOSA[8] um algoritmo polinomial com complexidade $O(n^6)$ que determina a concorrência $\gamma(\omega)$ associada. Assim, o problema pertence a NP, já que é possível verificar a validade de um certificado para a resposta SIM e tempo polinomial. Será mostrado que, dada uma instância de $I = (U, C)$ de NAEQ3SAT₃ e a instância especial de G de CONMAX, então I é satisfatível se, e somente se, G tem uma orientação acíclica ω associada a uma concorrência $\frac{1}{3}$. Primeiro, supõe-se I satisfatível através de uma atribuição de verdades η para U . A partir de η , a construção de uma orientação acíclica ω sobre G associada a uma quantidade de concorrência $\gamma(\omega) = \frac{1}{3}$ é definida pelos seguintes três passos:

- (i) Se o literal l em η tem um valor verdadeiro, então o nó x correspondente a l em G é um sumidouro em ω e o nó correspondente a \bar{l} é uma fonte em ω .

- (ii) Para cada $j \in \{1, 2, \dots, m\}$ selecione um nó de S_j conectado a um literal não satisfeito (ou seja, um literal que assume valor falso por conta da atribuição de valor, em η , à variável associada a este) e torne-o um sumidouro; selecione um nó y de S_j correspondente a um literal satisfeito e torne-o uma fonte. Neste passo, haverá um nó de S_j que não terá sido tornado nem sumidouro nem fonte e cujo nó de T_* do qual é vizinho pode corresponder tanto a um literal satisfeito quanto a um não satisfeito. As arestas incidentes a este nó, apesar disso, também estarão totalmente orientadas.
- (iii) Para todo $i \in \{1, 2, \dots, n\}$ os vértices a_i são tornados fontes e os vértices b_i são tornados sumidouros em ω (Figura 3.2b).

Está conluída a construção de ω . Agora, será provado que ω tem concorrência igual a $\frac{1}{3}$ através da definição das três orientações do período. Observe que, pelo item 1, todos os nós correspondentes aos literais verdadeiros operam em ω . Como η é uma atribuição de verdades satisfável de uma instância de $I = (U, C)$ de NAEQ3SAT_3 , pelo item 2, existe um nó operando em cada S_j , já que em cada cláusula existe pelo menos um literal não satisfeito. Pelo item 3, todos os vértices b_i operam neste passo. Quando as arestas dos sumidouros de ω são revertidas, uma orientação ω_1 é produzida (Figura 3.2c). Pelos itens 1 e 3, os nós w_i e d_i operam em ω_1 . Como todos os nós correspondentes a literais com valor verdadeiro foram sumidouros em ω , então fica claro que outro nó de cada S_j irá operar em ω_1 (este será o nó que não tiver se tornado fonte nem sumidouro, como destacado no item 2). É fácil observar que, em ω e ω_1 , todos os nós de S_* adjacentes a nós de T_* correspondentes a literais falsos por consequência de η operaram em ω_1 . Quando os sumidouros de ω_1 são revertidos, é obtida a orientação ω_2 (Figura 3.2a). O vértice em cada S_j que ainda não tiver operado (correspondente

a um literal verdadeiro) irá fazê-lo em ω_2 , já que a única aresta adjacente que, em ω_1 , o impedia de ser um sumidouro era interna a S_j (ou seja, era adjacente a outro nó de S_j) e já terá sido revertida. Também é claro que, já que os nós d_i operaram em ω_1 , os nós correspondentes aos literais não satisfeitos operam em ω

Assim, todos os nós de G operam uma única vez nessas três orientações e a orientação obtida a partir da reversão dos sumidouros de ω_2 é ω . Desta forma, temos $\gamma(\omega) = \frac{1}{3}$.

Agora, será mostrado que, se existe uma orientação acíclica ω sobre G com concorrência $\gamma(\omega) = \frac{1}{3}$, então I é satisfatível. Suponha que existe uma orientação acíclica ω sobre G com concorrência $\gamma(\omega) = \frac{1}{3}$. Esta orientação será utilizada para obter uma atribuição de verdade η para U .

Denotamos ω_1 , ω_2 e ω_3 as três orientações consecutivas na execução do ERA que caracterizam a concorrência $\gamma(\omega) = \frac{m}{p} = \frac{1}{3}$. Assumimos que, em ω_1 , todos os nós w_t , $t \in \{1, 2, \dots, n-1\}$, e d_r , $r \in \{1, 2, \dots, n\}$, operam conjuntamente, como na Figura 3.2c (o que é garantido, sem perda de generalidade, pelo Lema 3.5). Assim, nenhum nó de T_* correspondente a nenhum literal de I opera em ω_1 , pois o conjunto formado pelos vértices d_r são adjacentes a todos os nós de T_* .

No passo seguinte, com a orientação ω_2 formada pela operação do ERA, fica claro pelo Lema 3.4 que haverá, em cada T_i , $i \in \{1, 2, \dots, n\}$, exatamente 1 nó entre os dois correspondentes aos literais da variável u_i operando. Desta forma, atribuindo aos literais cujos nós correspondentes estão operando em ω_2 o valor verdadeiro, temos uma atribuição de verdades η para I que passamos a provar ser satisfatível.

Como já foi apresentado, a cada subgrafo S_j , $j \in \{1, 2, \dots, m\}$, corresponde uma cláusula $c_j \in C$. Em ω_2 , todas as arestas entre os nós de S_*

e os nós de T_* que representam os literais de I orientam-se na direção do nó do literal se este apresenta valor verdadeiro e na direção do nó de S_* caso contrário. Desta forma, precisamos apenas mostrar que, para cada S_j haverá pelo menos um nó com aresta orientada na direção de T_* e um nó com aresta orientada na direção de S_j . É fácil perceber, pelo Lema 3.4, que haverá pelo menos uma aresta direcionada de T_* para S_j , já que algum nó de S_j obrigatoriamente deverá estar operando em ω_2 . Para provar que haverá pelo menos uma aresta direcionada de S_j para T_* , considere o inverso, ou seja, que todas as arestas são direcionadas de T_* para S_j em ω_2 e assumamos t_1, t_2 e t_3 como os nós de T_* , vizinhos, respectivamente, dos nós s_1, s_2 e s_3 de S_j , nomeados de tal forma que s_k é sumidouro em ω_k , $k \in \{1, 2, 3\}$ (o que também é garantido pelo Lema 3.4). Porém, observe que, se s_1 só irá operar novamente na próxima vez que o ERA gerar ω_1 , então, em ω_2 , a aresta $s_1 t_1$ teria direção $s_1 t_1$ o que contradiz a hipótese inicial de que as arestas estarão direcionadas na direção de S_j em ω_2 . Assim, a atribuição de verdade definida a partir de ω é satisfável no sentido de $\text{NAE3SAT}_{\frac{2}{3}}$. \square

Na Figura 3.2 vemos uma seqüência de orientações derivadas da orientação ω com concorrência ótima $\gamma^*(\omega) = \frac{1}{3}$, produzida a partir de uma solução η da instância $I = (U, C) = (\{u_1, u_2, u_3, u_4\}, \{(u_1 \vee \overline{u_4}), (\overline{u_1} \vee \overline{u_2} \vee \overline{u_3}), (u_2 \vee \overline{u_3}), (u_3 \vee u_4), (\overline{u_1} \vee \overline{u_2} \vee \overline{u_4})\})$ de $\text{NAE3SAT}_{\frac{2}{3}}$, onde $\eta = \{V, F, F, V\}$, onde $V = \text{verdadeiro}$ e $F = \text{falso}$.

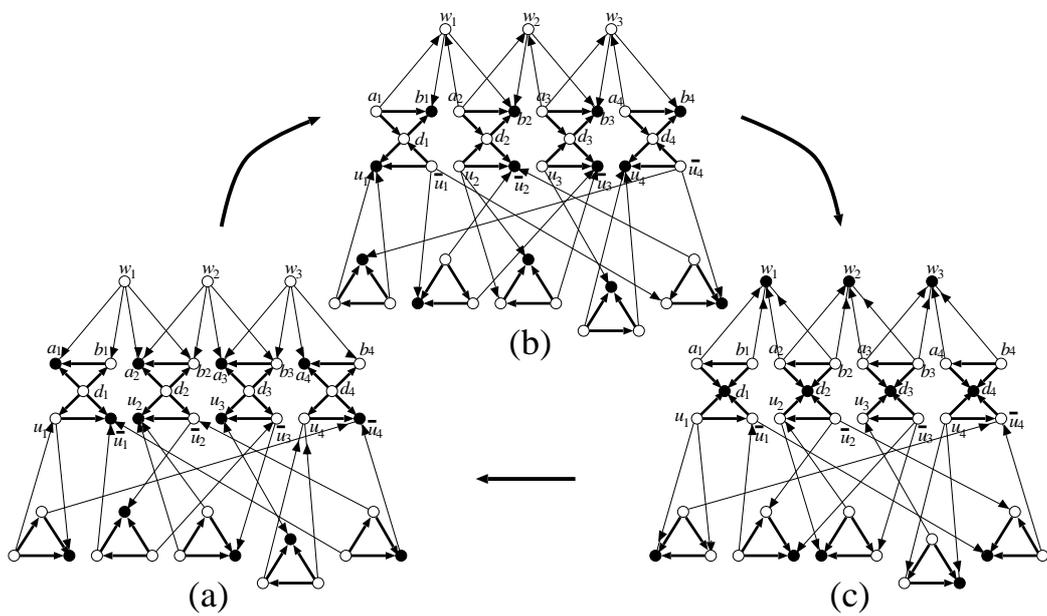


Figura 3.2: Orientações produzidas a partir de uma solução $\eta = \{V, F, F, V\}$ da instância de NAE3SAT dada por $I = (U, C) = (\{u_1, u_2, u_3, u_4\}, \{(u_1 \vee \bar{u}_4), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_3), (u_2 \vee \bar{u}_3), (u_3 \vee u_4), (\bar{u}_1 \vee \bar{u}_2 \vee \bar{u}_4)\})$.

3.5 Não-aproximabilidade de MININCON para Grafos em Geral

Sabe-se [10] que, se $P \neq NP$, então o problema de COLORAÇÃO não admite uma aproximação com razão menor que $n^{\frac{1}{7}-\epsilon}$ para qualquer $\epsilon > 0$. Nesta seção, é usado este resultado para provar que, dado um grafo qualquer $G = (N, E)$, o problema MININCON, a não ser que $P=NP$, não pode ser aproximado com uma razão menor que $n^{\frac{1}{7}-\epsilon}$ para qualquer $\epsilon > 0$. Tal resultado apresenta um limite para o tipo de algoritmo aproximativo que pode ser encontrado para a minimização do inverso da concorrência.

Teorema 3.7. *Dado um grafo $G = (N, E)$ com $n = |N|$ nós, então, a não ser que $P=NP$, MININCON não pode ser aproximado numa razão menor que $n^{\frac{1}{7}-\epsilon}$, para todo $\epsilon > 0$.*

Demonstração. É suficiente exibir uma L-redução [22] a partir do problema de coloração para MININCON porque, se um problema P_1 L-reduz para um problema P_2 e o problema P_2 tem um algoritmo de r-aproximação polinomial, então é possível provar que, descontadas as constantes, P_1 apresenta um algoritmo de r-aproximação polinomial.

De acordo com PAPADIMITRIOU[22], para L-reduzir COLORAÇÃO para MININCON, é necessário, dada uma instância $G = (N, E)$ do problema de COLORAÇÃO, obter um par f e g de algoritmos polinomiais no tamanho de G e um par de números reais positivos α e β , tais que o algoritmo f produz $f(G) = H = (N_H, E_H)$ uma instância de MININCON satisfazendo as seguintes condições:

$$(i) \text{ } Opt_{MININCON}(H) \leq \alpha Opt_{COLORACAO}(G) = \alpha \chi(G) \text{ e}$$

- (ii) dada uma solução viável de custo η_H para MININCON em H , o algoritmo g obtém uma solução viável de custo ξ_G para coloração em G tal que $|\chi(G) - \xi_G| \leq \beta |Opt_{MININCON}(H) - \eta_H|$.

O algoritmo f que produz a instância $H = f(G)$ de MININCON é definido pela adição de um nó universal v a G , $v \notin V(G)$ e para todo $u \in V(G)$, $vu \in E(H)$, isto é, $V(H) = V(G) \cup \{v\}$ e $E(H) = E(G) \cup \{vu : u \in V(G)\}$. Esta transformação torna possível a demonstração da primeira e mais simples parte da L-redução.

Dada uma coloração de G com as cores $1, 2, \dots, \chi(G)$, pode-se estender esta coloração para H pela atribuição de uma cor extra $\chi(G) + 1$ a v . Agora considere uma orientação de G que oriente cada aresta uw como $u \rightarrow w$ se u tem cor c_u , w tem cor c_w e $c_u < c_w$. Observe que o nó universal v será um sumidouro único em G numa determinada orientação acíclica no período do ERA e que a orientação de todas as arestas de todos os nós de G para v força que cada conjunto de sumidouro, respectivamente, com as cores $1, 2, \dots, \chi(G)$ operam numa seqüência em seguida à operação de v e recomeça novamente com uma nova operação de v . Isto define um período de tamanho $p = \chi(G) + 1$. Logo $Opt_{MININCON}(H) \leq p = \chi(G) + 1 \leq 2\chi(G) = 2Opt_{COLORACAO}(G)$. Desta forma $\alpha = 2$ é suficiente e conclui-se o primeiro passo da L-redução.

Se η_H é uma solução para MININCON em H com custo $\frac{p}{m}$, então necessariamente $m = 1$ [8]. Por conta do fato de que, no momento em que o nó universal é um sumidouro, nenhum outro nó pode operar também, tem-se que os nós de $V(H) \setminus \{v\}$ tornam-se sumidouros numa seqüência até v tornar-se um sumidouro novamente e nenhum nó opera duas vezes antes de v . Assim, os sumidouros de $H \setminus \{v\}$ definem uma partição de $V(G)$ em $p - 1$ conjuntos independentes e esta é a definição do algoritmo g . Logo,

$|\chi(G) - (p - 1)| \leq |\chi(G) + 1 - p| = |Opt_{MININCON}(H) - p|$ o que mostra que $\beta = 1$ é suficiente. \square

3.6 Algoritmo Aproximativo para MAXCON-MAX em Grafos com Grau Máximo Δ

Em 1941, BROOKS[12] provou que, se um grafo $G = (N, E)$ tem grau máximo Δ , número cromático $\chi(G)$, é conexo e não é nem um ciclo ímpar nem um grafo completo, então $\chi(G) \leq \Delta$. Em 1975, LOVÁSZ[18] mostrou um algoritmo polinomial que, neste caso, obtém uma Δ -coloração de G . O algoritmo a seguir é fortemente baseado no algoritmo citado de Lovász.

A

Entrada: Grafo $G = (N, E)$ com grau máximo Δ , não sendo G nem um grafo completo nem um ciclo com número ímpar de nós.

Saída: Uma orientação acíclica $A(G)$ em G associada à concorrência $\frac{2}{\Delta}$.

Execução:

- (i) Execute o algoritmo de Lovász obtendo a partição de N em conjuntos independentes $(N_1, N_2, \dots, N_\Delta)$.
- (ii) Para cada aresta $e = uv \in E$, onde $u \in N_i$ e $v \in N_j$ com $i < j$ oriente e na direção $v \rightarrow u$.

Em relação ao algoritmo A , vale a seguinte propriedade.

Lema 3.8. *Considere $G = (N, E)$ um grafo conexo com grau máximo Δ , onde G não é nem um ciclo ímpar nem um grafo completo. Se ω é uma*

orientação acíclica em G com concorrência associada $\gamma(\omega) = \frac{m}{p}$, então temos que $\frac{1}{\Delta} \leq \frac{m}{p} \leq \frac{1}{2}$.

Demonstração. A segunda desigualdade foi provada em BARBOSA[8] e é facilmente percebida pelo fato de que, no máximo, cada nó opera uma vez em cada 2 passos. Para a primeira desigualdade, basta perceber que a decomposição por sumidouros do grafo tem tamanho $\lambda = \Delta$. Portanto, na pior das hipóteses, um nó sairá sempre do nível mais alto da decomposição ($\Delta - 1$) e irá descendo de nível até tornar-se sumidouro no nível 0 da decomposição, retornando ao nível mais alto novamente. Desta forma, este irá operar 1 vez a cada Δ passos do algoritmo, perfazendo uma concorrência de, no mínimo, $\frac{1}{\Delta}$. \square

Desta forma, podemos produzir o seguinte teorema que versa sobre a razão de aproximação do algoritmo A .

Teorema 3.9. *A razão de desempenho do algoritmo A é, no máximo $\frac{2}{\Delta}$.*

Demonstração. Pelo Lema 3.8 temos que $\frac{1}{\Delta} \leq \frac{m}{p} \leq \frac{1}{2}$. Logo a razão de desempenho é limitada por $R_A = \frac{|A(G)|}{|Opt_{MAXCONMAX}(G)|} \geq \frac{\frac{1}{\Delta}}{\frac{1}{2}} = \frac{2}{\Delta}$. \square

3.7 Benchmark para Maximização de Concorrência

Um dos problemas para o desenvolvimento de heurísticas de maximização da concorrência do ERA é a dificuldade de obtenção das concorrências ótimas de um grafo de tal forma a permitir a avaliação dos resultados da heurística. Este problema é recorrente em qualquer problema NP-completo e, quando não resolvido, as heurísticas podem apenas ser comparadas em grafos com tamanho pequeno ou classe de grafos com resultado normalmente óbvio.

A solução para esse tipo de questão, normalmente, é a tentativa de estabelecimento de um *benchmark*, ou padrão de comparação, geralmente uma família de grafos com estrutura não óbvia, mas para a qual é possível estabelecer a solução ótima. Situação melhor ainda ocorre quando a família admite um algoritmo de construção que permite a geração de um número infinito de instâncias da família.

Os grafos de Mycielski [21] apresentam algumas destas características e são utilizados como *benchmarks* em problemas de coloração, pois têm número cromático conhecido. O estabelecimento da conjectura que é proposta a seguir viabilizaria sua utilização também como *benchmark* nos problema de maximização e minimização da concorrência do ERA. Abaixo, é apresentada a definição formal dos grafos de Mycielski.

Definição 3.10. *O grafo de Mycielski $M(k, n)$, $n \geq k \geq 2$ é o grafo completo em k vértices se $k = n$ ou é o grafo obtido a partir de $M(k, n - 1)$ tal que existe um conjunto independente (sem arestas entre cada par de vértices) adicional S , com $|S| = |V(M(k, n - 1))|$, de vértices gêmeos 1-1 aos vértices de $M(k, n - 1)$ em $M(k, n)$, existe ainda um vértice adicional w tal que a vizinhança de w é S , tal que um vértice v' de S gêmeo com v de $M(k, n - 1)$ possui a mesma vizinhança de v em $M(k, n - 1)$.*

Para os grafos de Mycielski, sabe-se que $\chi^*(M(2, n)) = n$.

Segue um teorema que mostra que o grafo de Mycielski $M(2, 4)$ tem concorrência ótima $\gamma^* = \frac{1}{4}$. Uma instância de $M(2, 4)$ tal que $\gamma = \frac{1}{4}$ é mostrada na Figura 3.3.

Teorema 3.10. *Dado o grafo de Mycielski $M(2, 4)$, temos que sua concorrência ótima é dada por $\gamma^*(M(2, 4)) = \frac{1}{4}$.*

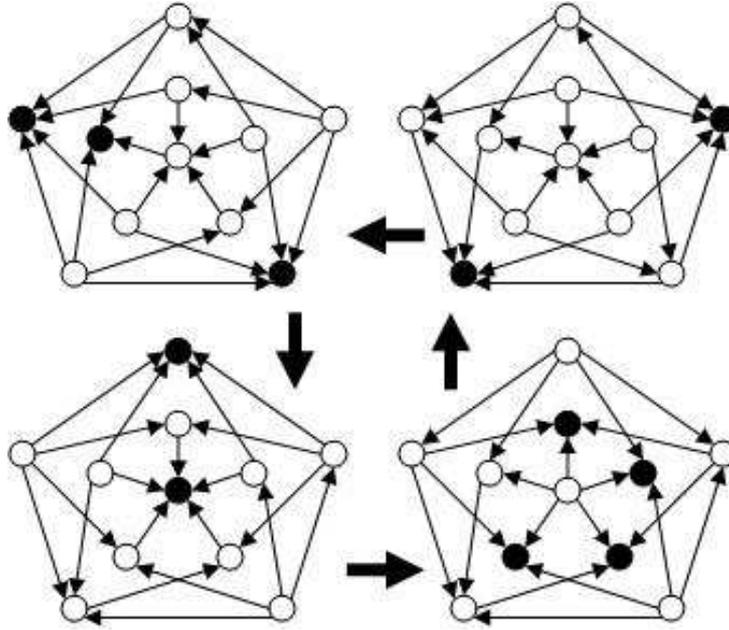


Figura 3.3: Grafo $M(2,4)$ com concorrência ótima $\gamma^* = \frac{1}{4}$.

Demonstração. Dado um grafo G não orientado, pode-se construir uma orientação a partir da coloração ótima $\chi^*(G)$, orientando cada aresta na direção do nó adjacente de menor cor. Com essa construção, o tamanho da decomposição por sumidouros é $\chi^*(G)$. A menor concorrência que será obtida com a orientação gerada por esta construção é aquela na qual o nó opera uma vez para cada $\chi^*(G)$ passos, ou seja, na qual um nó, após operar, é levado ao último nível na decomposição por sumidouros e, a partir daí, precisa esperar $\chi^*(G)$ passos até operar novamente. Como é sempre possível aplicar esta construção, temos que, para qualquer grafo $G = (N, E)$, $\gamma^*(G) \geq \frac{1}{\chi^*(G)}$. Portanto, para o grafo de Mycielski em questão, temos $\gamma^*(M(2,4)) \geq \frac{1}{\chi^*(M(2,4))} = \frac{1}{4}$.

Dessa forma, para provar que $\gamma^*(M(2,4)) = \frac{1}{4}$ é suficiente provar que para qualquer orientação ω de $M(2,4)$ existe um ciclo κ em 4 vértices de $M(2,4)$

tal que existam 3 arestas orientadas em um sentido e uma no sentido oposto, pois neste caso por (3.1) temos que $\gamma * (M(2, 4)) \leq \frac{1}{4}$.

$$\gamma(M(2, 4)) = \min_{\kappa \in M(2, 4)} \left\{ \min \left\{ \frac{n_{cw}(\kappa, \omega)}{|\kappa|}, \frac{n_{ccw}(\kappa, \omega)}{|\kappa|} \right\} \right\}. \quad (3.1)$$

Nós vamos assumir que um tal κ não exista em ω e vamos atingir uma contradição.

Esta prova usará a Figura 3.4. Assuma que ω é uma orientação acíclica para $M(2, 4)$ e que nessa orientação o vértice k , da Figura 3.4(c) é um sumidouro. Observe o C_5 definido pelos vértices a, b, c, d, e , nós observamos que em qualquer orientação existe um caminho orientado em 2 arestas consecutivas em a, b, c, d, e .

Nós assumimos que em ω estas arestas sejam \vec{de} e \vec{ea} .

As arestas aj e jd não podem ser orientadas \vec{aj} , \vec{jd} (Figura 3.4(b)) pois ω não seria acíclica; e não podem ser orientadas \vec{ja} , \vec{jd} (Figura 3.4(a)) e \vec{aj} , \vec{dj} (Figura 3.4(d)) pois ω definiria a contradição com o ciclo $\kappa = \{a, j, d, e\}$. Assim, resta a única orientação possível com \vec{ja} , \vec{dj} na Figura 3.4(c).

Primeiro de tudo, veja que na Figura 3.4(e), assumindo a contradição, as arestas \vec{dg} , \vec{ha} estão em ω pela hipótese de absurdo (que $M(2, 4)$ não possui um 4-ciclo com concorrência $\frac{1}{4}$) e, respectivamente, pelos vértices d, j, k, g e j, a, h, k .

Nós vamos dividir nossa redução ao absurdo em 2 partes:

- (i) \vec{cd} está em ω (Figura 3.4(k)). Esta é a parte fácil, pois pela hipótese de absurdo e por causa da aresta \vec{dg} necessariamente \vec{cb} e \vec{bg} estão em ω . Como \vec{bg} está em ω , então \vec{bi} está em ω , por causa das arestas \vec{bg} e \vec{gk} . Afirmamos que não é possível orientar a aresta ie sem que um 4-ciclo com concorrência $\frac{1}{4}$ seja definido. Se $\vec{ie} \in \omega$, então b, i, e, a define tal

ciclo. Se $\vec{ei} \in \omega$, então por causa da aresta \vec{ik} a aresta \vec{ef} está em ω e assim c, d, e, f define tal ciclo. A contradição procurada.

(ii) \vec{dc} está em ω (Figura 3.4(f)). Pela hipótese de absurdo e por causa da aresta \vec{dg} acontece uma das duas:

(a) \vec{bc} e \vec{bg} estão em ω (Figura 3.4(h)). Por causa da aresta \vec{gk} temos que \vec{bi} está em ω senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices i, b, g, k . Por causa da aresta \vec{ea} temos que \vec{ei} está em ω senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices b, i, e, a . Nesse caso por causa da aresta \vec{ei} e da aresta \vec{ik} temos que \vec{ef} está em ω senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices f, e, i, k . Assim, existe um ciclo em ω com concorrência $\frac{1}{4}$ nos vértices d, e, f, c , uma contradição.

(b) \vec{cb} e \vec{gb} estão em ω (Figura 3.4(g)). Antes de tudo, por causa da aresta \vec{ik} temos que \vec{ib} está em ω . Pela hipótese de absurdo e por causa das arestas \vec{fk} e \vec{hk} acontece uma das duas:

i. \vec{ch} e \vec{cf} estão em ω (Figura 3.4(j)). Neste caso, por causa da aresta \vec{ha} , temos que \vec{ba} está em ω , senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices c, h, a, b . Por causa da aresta \vec{ib} , temos que \vec{ie} está em ω , senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices e, i, b, a . Por causa da aresta \vec{ie} e da aresta \vec{fk} , temos que \vec{fe} está em ω , senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices i, e, f, k . Assim, existe um ciclo em ω com concorrência $\frac{1}{4}$ nos vértices d, c, f, e , uma contradição.

ii. \vec{hc} e \vec{fc} estão em ω (Figura 3.4(i)). Neste caso, por causa da aresta \vec{ha} , temos que \vec{ab} está em ω , senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices h, c, b, a . Por causa da aresta \vec{ea} ,

temos que \vec{ei} está em ω , senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices i, e, a, b . Por causa da aresta \vec{ik} , temos que \vec{ef} está em ω , senão temos o ciclo com concorrência $\frac{1}{4}$ nos vértices f, e, i, k . Assim, existe um ciclo em ω com concorrência $\frac{1}{4}$ nos vértices d, e, f, c , uma contradição.

□

A conjectura que fazemos é a seguinte.

Conjectura 3.1. *O grafo Mycielski $M(2, n)$, $n > 3$, tem concorrência ótima $\gamma^*(M(2, n)) = \frac{1}{n}$.*

Nossa esperança é que esta prova possa ser usada como base de indução na prova geral para a classe de Mycielski.

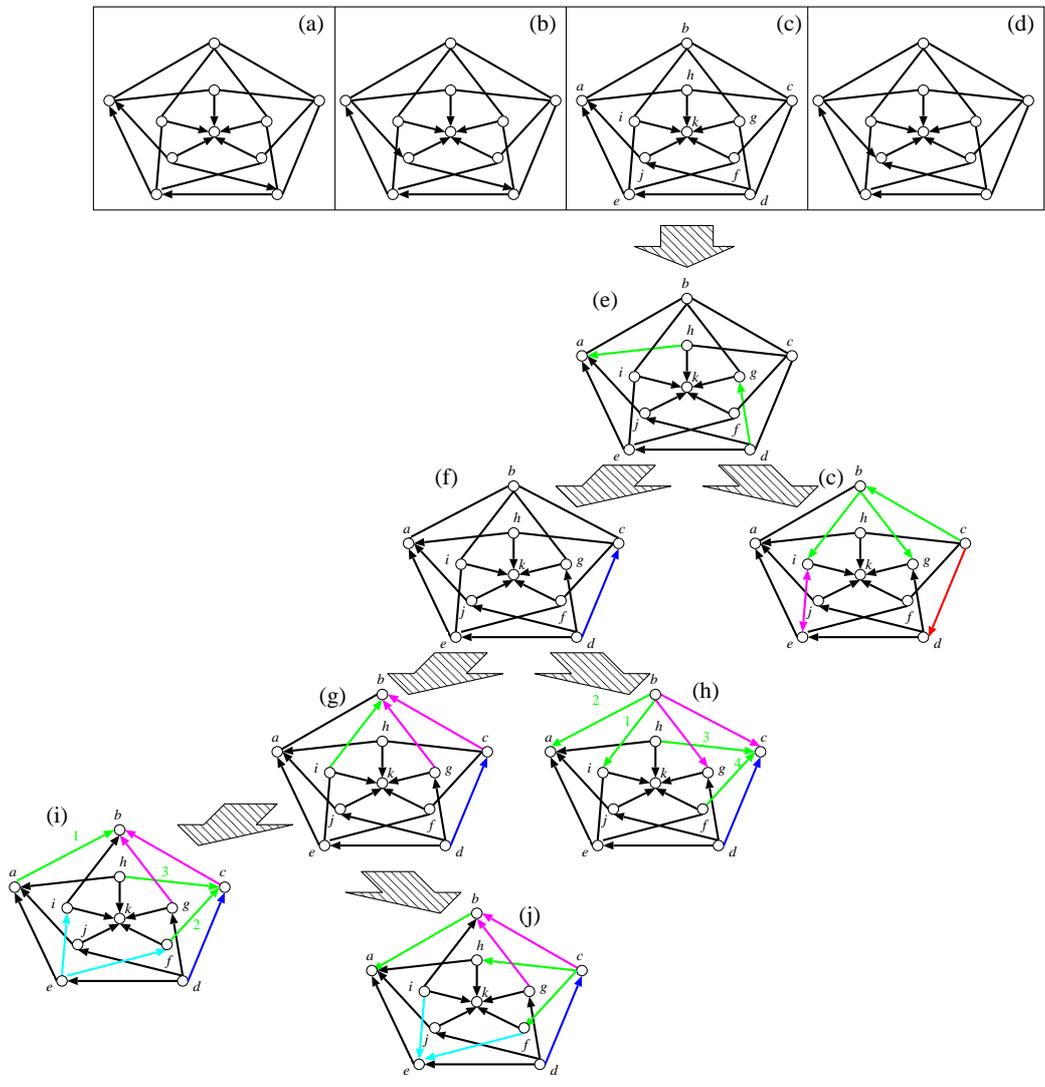


Figura 3.4: Grafo de Mycielski de ordem 2 e 4 para a prova do Teorema 3.10, $\gamma^*(M(2,4)) = \frac{1}{4}$.

Capítulo 4

Trilhas e Propriedades do ERA

4.1 Introdução

No Capítulo 3 - Otimização da Concorrência do ERA, foram apresentados resultados sobre o desempenho dos problemas de maximização e minimização da concorrência do ERA. Implicitamente, foi assumido um modelo centralizado de computação, onde há informações globais sobre o sistema, como número de nós, arestas, etc. O foco do presente capítulo se volta ao comportamento dinâmico do algoritmo e à sua natureza distribuída. Desta forma, é importante chamar a atenção para o fato de que o interesse se desloca dos algoritmos centralizados de otimização da concorrência para a caracterização do comportamento dinâmico do próprio ERA. Particularmente, foi investigado o fenômeno da modulação do sistema em execução pelos chamados ciclos principais.

Neste capítulo, são definidos vários conceitos, como os de trilha, ciclo principal, execução independente e execução canônica. São apresentadas provas de que toda trilha estabelece a monotonicidade do intervalo inoperante anterior e posterior. No caso do intervalo inoperante anterior, a monotonicidade

é de redução e, no caso do intervalo inoperante posterior, a monotonicidade é de crescimento. Também é mostrado que, na fase periódica do ERA, toda trilha pode ser considerada como tendo sido originada de um dos ciclos principais.

De posse destas ferramentas, é mostrado que o intervalo máximo inoperante e o intervalo mínimo inoperante do grafo são definidos pelos ciclos principais. Também, é apresentado um corolário que relaciona o número máximo de operações independentes ao máximo e o ao mínimo intervalo inoperante e, portanto, se estabelece que este também é determinado pelos ciclos principais. E, por fim, é apresentada uma prova de que, se todo nó de todos os ciclos principais operam uma vez a cada d passos, então todos os nós do sistema também irão operar uma vez a cada d passos.

4.2 Ciclo Principal (Senhor dos Anéis)

Como se sabe, a concorrência da execução do ERA em um grafo G qualquer orientado aciclicamente segundo ω é dada pela expressão [8]

$$\gamma(\omega) = \min_{\kappa \in K(G)} \left\{ \frac{\min \{n_{cw}(\kappa, \omega), n_{ccw}(\kappa, \omega)\}}{|\kappa|} \right\}$$

Seja $K(G)$ o conjunto dos ciclos de G . É fácil verificar que há um certo subconjunto $K'(G) \subseteq K(G)$ dos ciclos que satisfazem o mínimo mais externo desta equação. Foi provado em BARBOSA[8] que apenas os ciclos simples precisam ser considerados no cálculo de $\gamma(\omega)$, portanto definimos $K^*(G) \subseteq K'(G)$ como sendo o conjunto dos ciclos simples que satisfazem o mínimo desta expressão. Cada um dos ciclos de $K^*(G)$ é denominado de *Ciclo Principal* ou ainda, nesta tese, de *Senhor dos Anéis*.

Chamamos o conjunto $K^*(G)$ de *conjunto dos ciclos principais* de G . Estes ciclos têm propriedades especiais na execução do ERA que serão estudadas adiante.

4.3 Trilhas na Execução do ERA

A idéia de trilha é derivada do fato de que, no ERA, um nó só opera exatamente um passo após um subconjunto não vazio de seus vizinhos operar. Portanto, a partir de um sumidouro, podemos seguir uma seqüência de nós v_0, v_1, \dots de tal forma que v_i é sempre vizinho de v_{i+1} , sendo v_{i+1} sumidouro no passo seguinte em que v_i o foi.

Realmente, ao observarmos uma animação de um grafo executando o ERA, temos a impressão de estarmos vendo algo como “sumidouros percorrendo o grafo”, formando espécies de caminhos que chamamos aqui de trilhas. Segue a definição formal:

Definição 4.1. *Dado um grafo $G = (N, E)$ executando o ERA com um escalonamento dado por $\sigma = (\omega_1, \omega_2, \dots)$, uma **trilha** τ é uma seqüência não necessariamente finita de nós v_0, v_1, \dots , onde, para $t \in \mathbb{N}$, temos que $(v_t, v_{t+1}) \in E$ e $v_t \in \text{sinks}(\omega_t)$. Seja $\Gamma(G)$ o conjunto de todas as trilhas de G , tendo implícito o escalonamento a que se aplica.*

Uma *trilha infinita* é uma trilha em que a seqüência de nós é infinita (ver Figura 4.1). Estes conceitos são utilizados para algumas provas e análises nas próximas seções. Inicialmente, serão apresentadas algumas propriedades das trilhas.

Observe que podemos dividir as trilhas de uma execução do ERA de diversas formas. Não há, a princípio, nenhuma garantia de que uma trilha é maximal, por exemplo. Ou seja, eventualmente, poderemos considerar

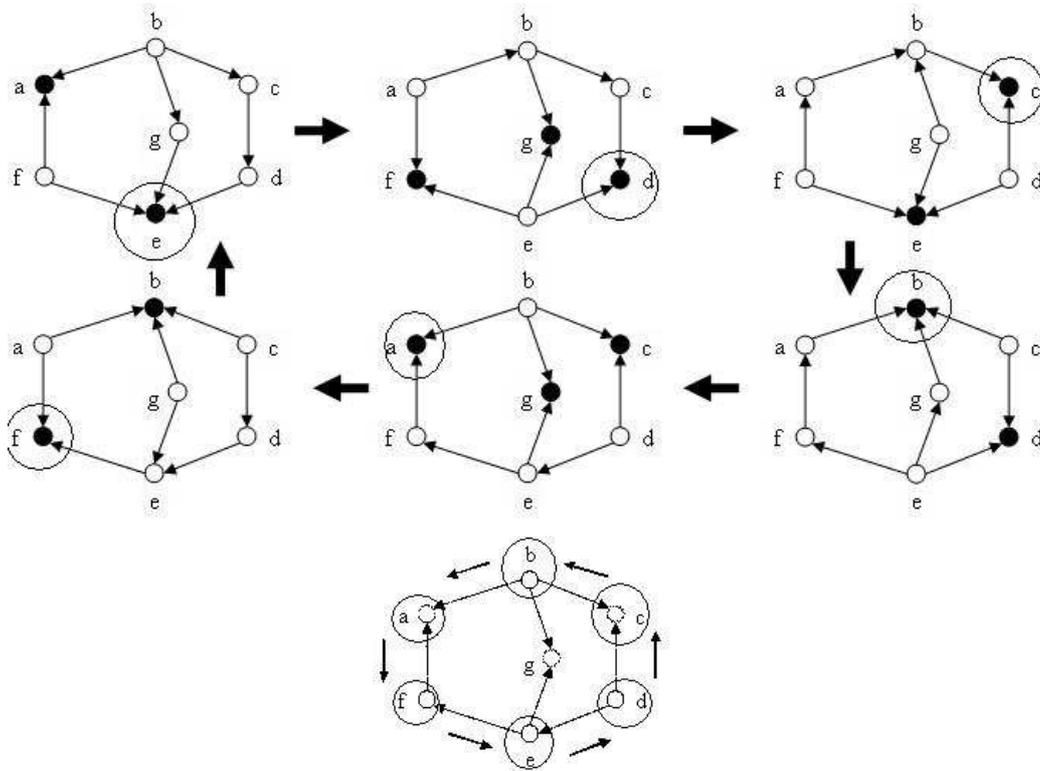


Figura 4.1: Exemplo de trilha infinita.

uma seqüência de operações dos nós v_0, v_1, \dots, v_k como 1, 2 ou mais trilhas, dependendo da necessidade. Assim, por exemplo, podemos considerar uma única trilha $\tau = v_0, v_1, \dots, v_k$ ou duas trilhas $\tau_1 = v_0, v_1$ e $\tau_2 = v_2, \dots, v_k$.

4.3.1 Propriedades das Trilhas

Antes de darmos prosseguimento, a seguinte definição é necessária.

Definição 4.2. *Dado um grafo $G = (N, E)$ executando o ERA, chamaremos t_v^i o instante de tempo (ou o passo) em que o nó $v \in N$ opera pela i -ésima vez ou pela $(i - 1)$ -ésima vez.*

Para facilitar os teoremas, assumimos sempre que, para um dado nó de referência $v \in N$, $t_v^i > t_u^i$ para todos os vizinhos $u \in viz(v)$ de v . Essa simplificação não resulta em perda de generalidade, mas pode causar confusão, já que, na verdade, t_v^i pode representar a $(i - 1)$ -ésima operação de v . Por exemplo, se um nó v opera pela quinta vez em t_v^5 e queremos considerar a última operação anterior de um vizinho u de v , dizemos que tal operação ocorreu em t_u^5 (ou seja, assumimos que $t_v^5 > t_u^5$) mesmo que no instante t_u^5 o nó u esteja operando pela quarta vez. Como sabemos que vizinhos operam alternadamente, temos certeza de que diferença a ser considerada no índice será de no máximo 1.

Como já foi dito, as próximas propriedades são utilizadas posteriormente em outras provas.

4.3.2 Trilha Estabelece Monotonicidade de Intervalo Inoperante Anterior

Considere uma trilha τ qualquer que alcança dois nós v e u , nesta ordem, porém não obrigatoriamente em tempos consecutivos. Dados δ_v^- e δ_u^- os intervalos de tempo (ou número de passos) em que v e u , respectivamente, ficaram sem operar até serem alcançados por τ , esta propriedade diz que $\delta_v^- \geq \delta_u^-$. Esta notação não é utilizada na prova, apenas aqui por simplicidade.

Ou seja, a monotonicidade para o intervalo inoperante anterior é de redução, o que significa que o intervalo inoperante anterior sempre reduz-se ou permanece igual ao longo dos nós que vão sendo alcançados pela trilha.

Teorema 4.1. *Considere um grafo $G = (N, E)$ qualquer executando o ERA e dois nós $v, u \in N$ alcançados por uma trilha τ nos tempos k e $k + \delta$, $k, \delta > 0$, ou seja $\tau(k) = v$ e $\tau(k + \delta) = u$. Se $t_v^i = k$ e $t_u^j = k + \delta$ (ou seja, τ*

alcança v e u , respectivamente, na i -ésima e j -ésima operação do nó), então temos que $t_u^j - t_u^{j-1} \leq t_v^i - t_v^{i-1}$.

Demonstração. A prova é por indução em δ . Provaremos que, se a propriedade é verdadeira para δ , então é válida para $\delta + 1$. Assim, provamos que, se vale a relação entre $\tau(k) = v$ e $\tau(k + \delta) = w$, então esta também vale para $\tau(k) = v$ e $\tau(k + \delta + 1) = u$. Pela hipótese de indução, temos que, para dois inteiros $z, i > 0$ quaisquer, $t_w^z - t_w^{z-1} \leq t_v^i - t_v^{i-1}$. Passamos a provar o passo de indução. Sabendo que há uma trilha que passa por w em $k + \delta$ e por u em $k + \delta + 1$, logo temos que $t_w^z + 1 = t_u^j$, para algum natural j . Como estes nós são obrigatoriamente vizinhos (já que τ passa por eles em tempos consecutivos), pela própria dinâmica do ERA é claro que eles operam alternadamente e, portanto, sabemos que $t_w^{z-1} < t_u^{j-1}$. Assim, podemos calcular $t_u^j - t_u^{j-1}$, fazendo as substituições necessárias:

$$t_u^j - t_u^{j-1} = (t_w^z + 1) - t_u^{j-1} < t_w^z + 1 - t_w^{z-1} \leq t_w^z - t_w^{z-1}$$

Portanto, prova-se o passo de indução, pois temos, pela transitividade da inequação, que $t_u^j - t_u^{j-1} \leq t_w^z - t_w^{z-1} \leq t_v^i - t_v^{i-1}$.

Para a base da indução, basta verificar que a propriedade vale para $\delta = 0$, o que é trivial, pois se trata de verificar a propriedade para o mesmo nó, o que é facilmente comprovado. \square

Um exemplo esquemático deste comportamento, pode-se ver na Figura 4.2. Observe que o nó v dá origem a uma trilha que alcança o nó u e que $\delta_v^- \geq \delta_u^-$. No exemplo da figura, existe a relação $t_v^i < t_u^{j-1}$, mas esta relação não é obrigatória, sendo casual deste exemplo.

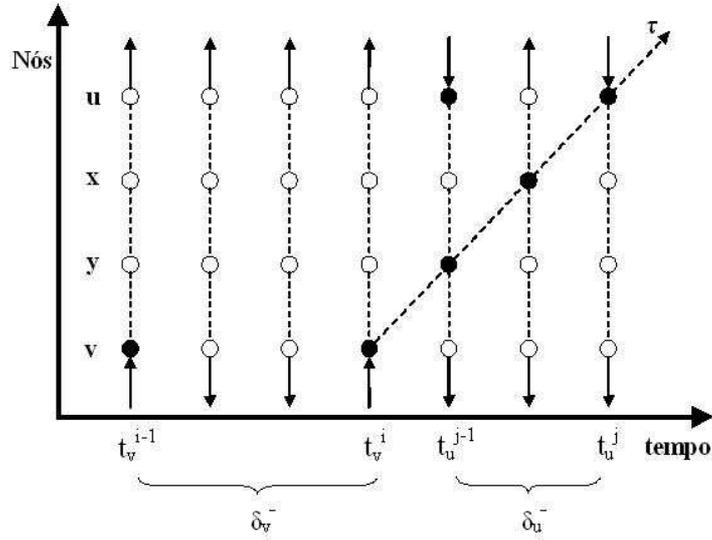


Figura 4.2: A trilha iniciada por v em t_v^i alcança u em t_u^j , garantindo que $\delta_v^- \geq \delta_u^-$.

4.3.3 Trilha Estabelece Monotonicidade de Intervalo Inoperante Posterior

Esta propriedade é, de certa forma, complementar à anterior. Considere inicialmente uma trilha τ qualquer que alcança dois os nós v e u , nesta ordem, porém não obrigatoriamente em tempos consecutivos. Dados os intervalos δ_v^+ e δ_u^+ de tempo que os nós ficam sem operar depois de τ tê-los alcançado, temos que $\delta_v^+ \leq \delta_u^+$.

Ou seja, a monotonicidade para o intervalo inoperante posterior é de crescimento, o que significa que o intervalo inoperante anterior sempre reduz-se ou permanece igual ao longo dos nós que vão sendo alcançados pela trilha.

Teorema 4.2. *Considere um grafo $G = (N, E)$ qualquer executando o ERA e dois nós $v, u \in N$ alcançados por uma trilha τ nos tempos k e $k + \delta$ ($k, \delta > 0$), respectivamente, ou seja $\tau(k) = v$ e $\tau(k + \delta) = u$. Se $t_v^i = k$ e $t_u^j = k + \delta$*

(ou seja, τ alcança v e u , respectivamente, na i -ésima e j -ésima operação do nó), então temos que $t_v^{i+1} - t_v^i \leq t_u^{j+1} - t_u^j$.

Demonstração. A prova é por indução em δ . Provaremos que, se a propriedade é verdadeira para δ , então é válida para $\delta + 1$. Assim, provamos que, se vale a relação entre $\tau(k) = v$ e $\tau(k + \delta) = w$, então esta também vale para $\tau(k) = v$ e $\tau(k + \delta + 1) = u$. Pela hipótese de indução, temos que, para dois inteiros $z, i > 0$ quaisquer, $t_v^{i+1} - t_v^i \leq t_w^{z+1} - t_w^z$. Passamos a provar o passo de indução. Sabendo que há uma trilha que passa por w em $k + \delta$ e por u em $k + \delta + 1$, logo temos que $t_w^z + 1 = t_u^j$, para algum natural j . Como estes nós são obrigatoriamente vizinhos (já que τ passa por eles em tempos consecutivos), pela própria dinâmica do ERA é claro que eles operam alternadamente e, portanto, sabemos que $t_w^{z+1} < t_u^{j+1}$. Assim, podemos calcular $t_u^{j+1} - t_u^j$, fazendo as substituições necessárias:

$$t_u^{j+1} - t_u^j = t_u^{j+1} - (t_w^z + 1) > t_w^{z+1} - t_w^z - 1 \geq t_w^{z+1} - t_w^z$$

Portanto, prova-se o passo de indução, pois temos, pela transitividade da inequação, que $t_u^{j+1} - t_u^j \geq t_w^{z+1} - t_w^z \geq t_v^{i+1} - t_v^i$.

Para a base da indução, basta verificar que a propriedade vale para $\delta = 0$, o que é trivial, pois se trata de verificar a propriedade para o mesmo nó, o que é facilmente comprovado. \square

Um exemplo esquemático pode ser observado na Figura 4.3. Observe que o nó v dá origem a uma trilha que alcança o nó u e que $\delta_v^+ \leq \delta_u^+$.

4.4 Execução Independente

Intuitivamente, uma *execução independente* de um subgrafo é tal que, extraíndo-se o subgrafo do grafo no qual ele está incluído, sua execução

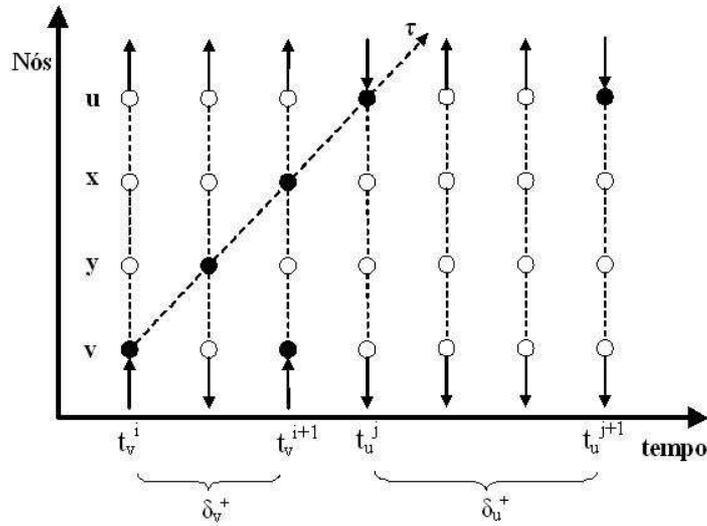


Figura 4.3: A trilha iniciada por v em t_v^i alcança u em t_u^j , garantindo que $\delta_v^+ \leq \delta_u^+$.

mantém-se intacta (ver Figura 4.4). Os subgrafos com execução independente apresentam propriedades que serão utilizadas para as provas seguintes.

Precisaremos de algumas notações. Considere uma orientação acíclica ω em um grafo G e um subgrafo G' de G . Chamamos $\omega^{G'}$ a *projeção da orientação* ω em G' de tal forma que cada aresta de G' assume a mesma orientação induzida por ω em G . Por exemplo, as arestas do subgrafo G' induzido pelos vértices $\{a, b, c\}$ na Figura 4.4 definem uma projeção $\omega^{G'}$ da orientação ω em G' .

Denotamos também o *escalonamento* ou a *execução* do ERA em um grafo G qualquer como $\sigma = (\omega_1, \omega_2, \dots)$, sendo uma seqüência de orientações acíclicas tal que, para todo inteiro $i > 0$, temos que $\omega_i = g(\omega_{i-1})$ (lembrar do O Capítulo 2 - Conceitos Básicos, onde é definido que a função $g(\omega)$ aplicada a uma orientação acíclica ω retorna a orientação gerada pela execução de 1 passo do ERA, ou seja, pela reversão das arestas de todos os sumidouros).

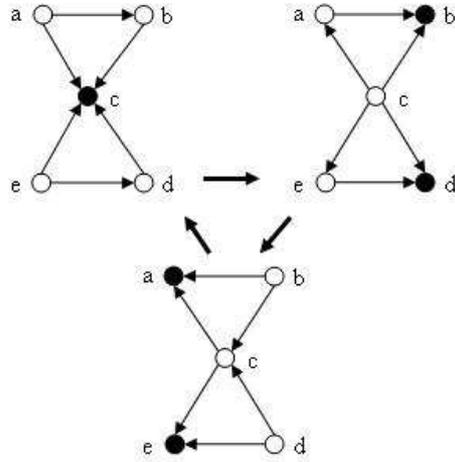


Figura 4.4: Exemplo execução independente. O comportamento do subgrafo induzido pelos nós a, b, c seria o mesmo se destacado do grafo induzido pelos nós a, b, c, d, e .

Para dois grafos $G = (N, E)$ e $G' = (N', E')$ dizemos $G' \subseteq G$ quando $N' \subseteq N$ e $E' \subseteq E$, ou seja, quando G' é subgrafo de G . Utilizamos as notações de conjunto de maneira livre, como $v \in \text{sinks}(\omega) \cap G'$, onde o nó v pertence ao conjunto de nós resultado da interseção do conjunto $\text{sinks}(\omega)$ com os nós de G' . Dados nós $v, u \in N$ ou uma aresta $e = (v, u)$ denotamos, por simplificação, $v \in G$ e $e \in G$ e $(v, u) \in G$. Tais notações incluem ciclos, assim, podemos denotar $\kappa \subseteq G$ e $v \in \kappa$ ou $e \in \kappa$.

Introduzimos a notação $\gamma_v(\omega, q) = \frac{m_v(\omega, q)}{q}$, representando a concorrência do nó v acumulado em q passos a partir da orientação acíclica ω , onde $m_v(\omega, q)$ denota o número de vezes que v opera nos q primeiros passos da execução do ERA iniciado com a orientação acíclica ω . Também denota-se $\gamma_v(\omega) = \lim_{q \rightarrow +\infty} \gamma_v(\omega, q)$. Lembre que, no O Capítulo 2 - Conceitos Básicos,

já havia sido introduzida a notação $\gamma(\omega)$ para denotar a concorrência do grafo como um todo.

4.4.1 Definição e Propriedades da Execução Independente

Começamos definindo formalmente uma execução independente de um subgrafo.

Definição 4.3. *Considere um grafo G e um subgrafo induzido G' de G . Dada uma execução $\sigma = (\omega_1, \omega_2, \dots)$ do ERA em G , dizemos que G' apresenta **execução independente** se, extraíndo-se G' de G e executando o ERA em G' a partir da orientação $\omega_1^{G'}$, obtivermos uma execução $\sigma' = (\omega'_1, \omega'_2, \dots)$ tal que, para todo inteiro $i > 0$, $\omega'_i = \omega_i^{G'}$, ou ainda, $g^i(\omega_1^{G'}) = g^i(\omega_1)^{G'}$.*

Por exemplo, considere apenas o subgrafo induzido pelos nós $\{a, b, c\}$ na Figura 4.4. Observe que os nós que se tornam sumidouros neste grafo também são sumidouros do grafo induzido pelos nós $\{a, b, c, d, e\}$, o que garante que sua execução é independente.

Apresentamos um lema que relaciona a execução independente com uma característica genérica da execução de um subgrafo. É mostrado que, dado um grafo G e um subgrafo G' de G e uma execução $\sigma = (\omega_1, \omega_2, \dots)$ do ERA em G com sua respectiva projeção em G' , se todos os nós sumidouros em $\omega_i^{G'}$ forem também sumidouros em ω_i^G , $i \in \mathbb{N}$, então teremos uma execução independente. Observe que, em uma execução do ERA desse tipo, um determinado nó v pode ser sumidouro em G' e não o ser em G , bastando que uma aresta adjacente a v esteja orientada na direção contrária a v e não pertença a G' . Este conceito e esta prova foram exploradas com pequenas modificações de notação em MALKA[19].

Lema 4.3. *Considere um grafo $G = (N, E)$ qualquer executando o ERA segundo $\sigma = (\omega_1, \omega_2, \dots)$ e um subgrafo $G' = (N', E')$ de G . Se, para todo ω_i , $i > 0$, e para todo $v \in N$, for verdade a relação $v \in \text{sinks}(\omega_i^{G'}) \rightarrow v \in \text{sinks}(\omega_i)$ então é verdade que G' apresenta execução independente.*

Demonstração. Pela Definição 4.3, temos que há execução independente em um subgrafo G' se, e somente se, para todo inteiro $i > 0$, é verdade que $g^i(\omega_1^{G'}) = g^i(\omega_1)^{G'}$.

Claramente, a relação $v \in \text{sinks}(\omega_i) \rightarrow v \in \text{sinks}(\omega_i^{G'})$, inversa do enunciado, é verdadeira, pois, na orientação inicial, um sumidouro de G também o será em G' . Além disso, todas as arestas invertidas em G também o serão em G' e, portanto, qualquer sumidouro formado em G , novamente, também será um sumidouro de G' .

Portanto, para provarmos que $g^i(\omega_1^{G'}) = g^i(\omega_1)^{G'}$, devemos considerar que $g^i(\omega_1)^{G'}$ é uma informação dada e verificar se $g^i(\omega_1^{G'})$ produz a mesma saída, assumindo que esta execução se dá de forma independente de G . Ou seja, é preciso provar que, considerando apenas os nós e arestas de G' , as orientações produzidas nas arestas de E' pela execução de $g(\omega_1^{G'})$, ou seja, a inversão das arestas de G' , são iguais às produzidas nas arestas de E' pela execução de $g(\omega_1)^{G'}$, ou seja, a inversão das arestas de G com posterior projeção em G' .

As arestas de E' invertidas pela aplicação de $g(\omega_1^{G'})$ são aquelas adjacentes a sumidouros de $\omega_1^{G'}$. Pelo enunciado, estes também são sumidouros de ω_1 e, portanto, as arestas de E' revertidas por $g(\omega_1)$ são as mesmas. Como são, por premissa, gerados os mesmos sumidouros, então o mesmo processo pode se repetir indefinidamente, logo, $g^i(\omega_1^{G'}) = g^i(\omega_1)^{G'}$. \square

Abaixo, apresentamos uma prova de que a concorrência de um subgrafo com execução independente é igual à concorrência do grafo no qual está inserido.

Pelo Capítulo 2 - Conceitos Básicos, sabe-se que, para um grafo qualquer $G = (N, E)$ executando o ERA segundo $\sigma = (\omega_1, \omega_2, \dots)$, tem-se que, para todo $v \in N$, $\gamma_v(\omega_1) = \gamma(\omega_1)$, ou seja, a concorrência de cada nó, no limite, é igual à do grafo como um todo. Também é possível encontrar prova semelhante ao Teorema 4.4 em MALKKA[19].

Teorema 4.4. *Considere um grafo $G = (N, E)$ executando o ERA segundo $\sigma = (\omega_1, \omega_2, \dots)$. Se existe um subgrafo G' de G com execução independente $\sigma' = (\omega_1^{G'}, \omega_2^{G'}, \dots)$, então temos que existe um ciclo simples $\kappa \in K^*$ tal que $\kappa \subseteq G'$.*

Demonstração. Considere um nó qualquer $v \in G'$, a execução $\sigma = (\omega_1, \omega_2, \dots)$ do ERA em G e a execução independente $\sigma' = (\omega_1^{G'}, \omega_2^{G'}, \dots)$ de G' que ocorre extraindo-se G' de G . Por definição, para qualquer inteiro $i > 0$, se $v \in \text{sinks}(\omega_i)$ então $v \in \text{sinks}(\omega_i^{G'})$ e, pelo Lema 4.3, se $v \in \text{sinks}(\omega_i^{G'})$ então $v \in \text{sinks}(\omega_i)$. Então, por consequência, $v \in \text{sinks}(\omega_i)$ se, e somente se, $v \in \text{sinks}(\omega_i^{G'})$.

Assim, é claro que, num mesmo conjunto de passos, cada nó de G' opera em σ' o mesmo número de vezes que opera em σ . Portanto, para todo $v \in G'$, temos que $\gamma_v(\omega_1^{G'}) = \gamma_v(\omega_1)$. Temos também [8] que $\gamma_v(\omega_1) = \gamma(\omega_1)$. Logo, por transitividade, temos $\gamma(\omega_1) = \gamma(\omega_1^{G'})$.

Desta forma, sabemos que $\gamma(\omega_1^{G'}) = \min_{c \in K(G')} \left\{ \min \left\{ \frac{n^+(c, \omega_1^{G'})}{|c|}, \frac{n^-(c, \omega_1^{G'})}{|c|} \right\} \right\}$. Assim, existe um ciclo κ de G' que satisfaz o mínimo da expressão. Portanto, $\gamma(\omega_1^{G'}) = \min \left\{ \frac{n^+(\kappa, \omega_1^{G'})}{|\kappa|}, \frac{n^-(\kappa, \omega_1^{G'})}{|\kappa|} \right\}$. Como κ também é um ciclo de G e $\gamma(\omega_1^{G'}) = \gamma(\omega_1)$, então κ também satisfaz o mínimo da expressão equivalente para $\gamma(\omega_1)$, o que o caracteriza como um ciclo printipal. \square

4.4.2 Execução Canônica

Em MALKA[19], é mostrado que qualquer grafo C_k , ou seja, todo ciclo com k nós, converge para um *escalonamento rotacional*, no qual o número de sumidouros é constante e estes se deslocam em um mesmo sentido em torno do ciclo. É fácil perceber que um ciclo κ , quando é um subgrafo de um grafo qualquer G , quando apresenta execução independente, executa o mesmo escalonamento rotacional, já que a própria definição de execução independente é que o subgrafo apresenta comportamento idêntico ao que teria se fosse isolado do resto do grafo. Neste texto, chamamos o escalonamento rotacional de *execução canônica do ciclo* e esta é definida abaixo.

Observe que assumimos uma notação de tal forma que (a) os nós de um ciclo são nomeados com sub-índices em ordem crescente e no sentido mais conveniente e (b) o uso do operador mod seja implícito nos sub-índices, tal que, para um ciclo κ e um índice $i \in \mathbb{N}$ qualquer, v_i representa $v_{i \bmod |\kappa|}$.

Definição 4.4. *Considere um grafo $G = (N, E)$, um ciclo $\kappa \subseteq G$ formado pelos nós v_0, v_1, \dots e uma execução $\sigma = (\omega_1, \omega_2, \dots)$ do ERA em G cuja fase periódica se inicia no tempo $t^p \geq 0$. Dizemos que κ está em **execução canônica** se existir um conjunto não vazio $\Gamma^*(G) = \{\tau_1, \dots, \tau_q\}$ de trilhas infinitas tal que, para quaisquer inteiros $t > t^p$ e $i > 0$ e para toda $\tau \in \Gamma^*(G)$, for verdade que $\tau(t) = v_i$ se, e somente se, $\tau(t-1) = v_{i-1}$ e, além disso, for verdade que $v \in \text{sinks}(\omega_t) \cap \kappa$ se, e somente se, existe $\tau \in \Gamma^*(G)$ tal que $v = \tau(t)$.*

A Figura 4.5 mostra um ciclo em execução canônica. Note que, neste caso, o ciclo não é subgrafo próprio de um grafo maior, mas, em uma execução independente, seu comportamento poderia ser idêntico.

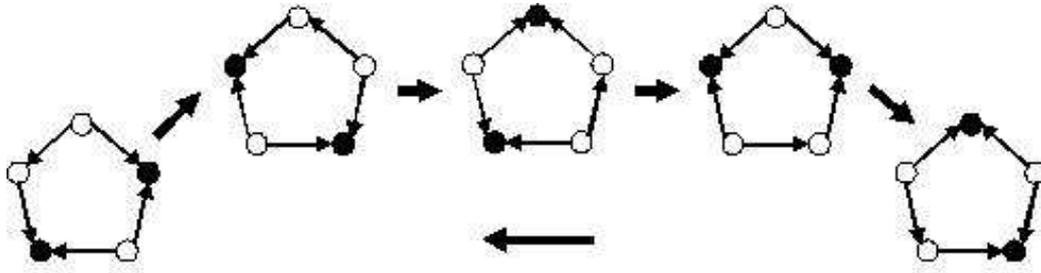


Figura 4.5: Exemplo de execução canônica.

A seguir, são apresentadas algumas propriedades das trilhas relacionadas aos ciclos principais.

4.5 Propriedades das Trilhas e Ciclos Principais

4.5.1 Ciclo Percorrido por Trilha Infinita

Em seguida, mostramos que todo ciclo que é percorrido por uma trilha infinita é um ciclo principal. A idéia é a seguinte: se uma trilha infinita τ percorre, a partir de um tempo t_0 , um ciclo κ sempre em um mesmo sentido, todas as arestas de κ orientadas, em t_0 , no mesmo sentido do deslocamento de τ darão origem a outras trilhas com características similares, ou seja, infinitas e percorrendo o ciclo no mesmo sentido.

Estas trilhas induzidas por τ passam a percorrer o ciclo na mesma direção, caracterizando uma execução independente, o que indica a existência de um ciclo principal (por definição, um ciclo simples) $\kappa^* \in K^*(G)$, contido em κ . Observe que κ não é obrigatoriamente um ciclo simples.

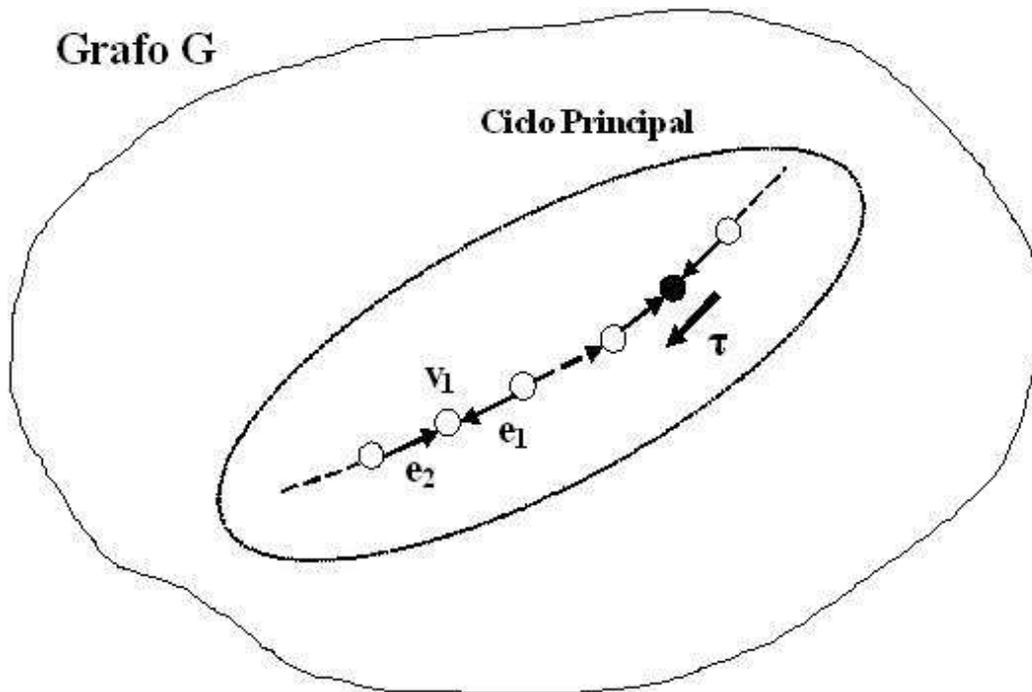


Figura 4.6: Antes de τ alcançar o nó v_1 , a aresta e_1 terá que ser revertida. Logo, v_1 irá operar antes disso. Pela operação de v_1 , e_2 ficará direcionado no sentido do deslocamento de τ .

Começamos com o Lema 4.5, que prova que uma trilha infinita em um ciclo qualquer não obrigatoriamente simples implica a existência de outras trilhas infinitas no mesmo ciclo. Uma ilustração sobre parte do que é apresentado neste lema pode ser verificado na Figura 4.6. Nesta, é destacado que, havendo uma trilha infinita e uma aresta orientada no mesmo sentido desta trilha, tal aresta deverá reverter sua orientação antes que a trilha a alcance. Este lema servirá de base para outras provas adiante.

Lema 4.5. *Dada um grafo $G(N, E)$ executando o ERA segundo $\sigma = (\omega_1, \omega_2 \dots)$, um ciclo κ de G (não obrigatoriamente simples) formado pelos nós $v_0, v_1, \dots, v_{|\kappa|-1}$ e uma trilha infinita τ para a qual tal que existe $t_0 \geq 0$ tal que $\tau(t_0 + i) = v_i$,*

para $i \in \mathbb{N}$. Assuma $d_{prox} > 1$ o menor inteiro positivo para o qual existe uma trilha τ_{prox} , possivelmente igual a τ , tal que $\tau_{prox}(t_0+i) = \tau(t_0+i+d_{prox})$, ou seja, τ_{prox} é a próxima trilha infinita percorrendo κ à frente de τ . Considerando o menor inteiro $d > 0$ para o qual exista a aresta orientada $\overrightarrow{v_d v_{d+1}}$, temos que, se $d \leq d_{prox} - 2$, então existirão dois inteiros $t'_0 \geq t_0$ e $d' \leq d$ para os quais haverá uma outra trilha infinita τ' tal que $\tau'(t) = \tau(t + d')$, para qualquer inteiro $t \geq t'_0$.

Demonstração. Inicialmente, verifica-se que sempre existe um d para o qual existe a aresta orientada $\overrightarrow{v_d v_{d+1}}$, já que, no máximo, $d = d_{prox} - 1$ e, portanto, esta será a aresta $\overrightarrow{v_{d-1} v_0}$ que precede o sumidouro da trilha τ_{prox} . Como dito na descrição do lema, este caso não é considerado, já que a trilha infinita já existe.

O restante da prova é por indução. Inicialmente, provamos que o lema é verdadeiro para a base de indução, ou seja, $d = 1$. Pela hipótese de indução, tem-se que existe a aresta orientada $\overrightarrow{v_1 v_2}$. Como, por premissa, temos que $\tau(t_0) = v_0$ e $\tau(t_0 + 1) = v_1$, então deverá haver uma aresta orientada $\overrightarrow{v_2 v_1}$ em $t_0 + 1$. Para que isso ocorra, o nó v_2 deverá operar em t_0 , ou seja, pode-se concluir que $v_2 \in sinks(\omega_{t_0})$. Porém, a operação de v_2 irá produzir a aresta orientada $\overrightarrow{v_2 v_3}$ em $t_0 + 1$. Já que $\tau(t_0 + 1) = v_1$, ou seja, v_1 é sumidouro em $t_0 + 1$, então temos reproduzida a situação inicial descrita. Como a mesma situação se repete para $t = t_0 + 2, t = t_0 + 3, \dots$, então a seqüência de operações de v_2, v_3, v_4, \dots nos tempos $t_0, t_0 + 1, t_0 + 2, \dots$ forma uma trilha infinita τ' para a qual $\tau'(t) = \tau(t + 2), \forall t \geq t_0$. Assim, $t'_0 = t_0$ e $d' = 2$.

Agora, para o passo de indução, assumimos que, se tivermos a prova verdadeira para $d = z$, podemos provar para $d = z + 1$. Pelo enunciado, em $t = t_0$, teríamos que $\tau(t_0) = v_i \in sinks(\omega_{t_0})$ e a aresta orientada $\overrightarrow{v_{z+1} v_{z+2}}$. Sabemos que o nó v_{z+2} deverá operar antes de a trilha alcançar o nó v_{z+1} , já

que é necessário que a existência da aresta orientada $\overrightarrow{v_{z+2}v_{z+1}}$ em $t = t_0 + z + 1$.

Ou seja, deve existir um inteiro $0 < w < z$ tal que $v_{z+2} \in \text{sinks}(\omega_{t_0+w})$.

Se $w > 1$, temos $\tau(t_0 + w) = v_w \in \text{sinks}(\omega_{t_0+w})$ e uma aresta orientada $\overrightarrow{v_{z+2}v_{z+3}}$ em $t = t_0 + w$. Como a distância entre v_w e v_{z+2} é dada por $z + 2 - w \leq z$, temos que, pela hipótese de indução, a propriedade estaria provada, já que a distância é menor ou igual a z .

Porém, se $w = 1$, então $\tau(t_0 + 1) = v_1 \in \text{sinks}(\omega_{t_0+1})$ e $v_{z+2} \in \text{sinks}(\omega_{t_0+1})$. Então, nesse caso, teremos a mesma situação descrita no início, a saber: um sumidouro $\tau(t_0 + 1)$, participante de τ , e uma aresta orientada no sentido $\overrightarrow{v_{z+2}v_{z+3}}$ distantes $z + 1$ entre si. Desta forma, nos passos seguintes, será possível que ocorra uma situação em que $w > 1$, como já foi visto e a formação da trilha estará provada. Porém, se esta situação perdurar infinitamente, temos que os nós operando seqüencialmente com distância $z + 1$ do nó participante da trilha τ , ou seja, os nós $v_{z+1}, v_{z+2}, v_{z+3}, \dots$, comporão a nova trilha infinita operando nos tempos $t = t_0, t = t_0 + 1, t = t_0 + 2, \dots$. Em ambos os casos, a propriedade também está provada. \square

Em seguida, prova-se que a existência de uma trilha infinita percorrendo um subgrafo G' de G , implica a existência de um ciclo simples contido em G' tal que $\kappa^* \in K^*$.

A prova tem, na verdade, duas partes. A primeira diz que uma trilha τ' que percorre infinitamente um subgrafo G' a partir de um tempo t^p acaba passando por um mesmo nó de G' em tempos $t^p + k_1p$ e $t^p + k_2p$, distantes entre si um múltiplo do período p . Portanto, é definido um ciclo κ (não obrigatoriamente simples) que, pelas características periódicas do ERA, será percorrido infinitamente por uma trilha τ originada de τ' .

Na segunda parte, é utilizado o Lema 4.5 para dizer que cada aresta de κ orientada no sentido em que τ percorre κ dará origem a outra trilha infinita. Todas estas trilhas percorrerão κ infinitamente e no mesmo sentido de τ . A partir daí, utiliza-se o Lema 4.3 para provar que κ tem execução independente e, pelo Teorema 4.4, argumenta-se que o subgrafo G' contém pelo menos um dos ciclos principais, já que κ está contido em G' .

A Figura 4.7 mostra a idéia geral do teorema. Observe que τ percorre infinitamente os nós dentro da fronteira do subgrafo G' . Portanto, haverá a formação de um ciclo contido em G' que se tornará o ciclo principal. Note que outros ciclos principais podem existir fora da fronteira de G' . Também estamos assumindo na figura, por simplicidade, que κ é ciclo simples. Se κ não for um ciclo simples, podemos dizer que este apresentará execução independente e, portanto, uma parte de κ será o ciclo principal

O motivo pelo qual κ converge para uma execução independente é mostrado na Figura 4.8 e tem como base o Lema 4.5 que fala da formação de novas trilhas a partir de uma que percorre infinitamente um ciclo. Novamente, consideramos, por simplicidade, um ciclo simples.

Assume-se que a linha pontilhada é um ciclo qualquer de um grafo. Em (a) a trilha τ_1 se desloca no sentido horário e a aresta e_1 está orientada no mesmo sentido. Antes de τ_1 alcançar o nó v_1 tornando-o um sumidouro, é necessário que e_1 seja revertida por uma operação de v_2 . Em (b), τ_1 ainda se desloca na direção de v_1 e vemos que o nó v_2 tornou-se um sumidouro, pois outras arestas de fora do ciclo foram orientadas em sua direção. Com a operação de v_2 , a aresta e_1 reverte sua orientação, tornando possível que τ_1 alcance v_1 , o que pode ser visto em (c). Note que, neste exemplo, considera-se que a trilha τ_2 já foi formada, mas esta poderia surgir posteriormente. Em (d), vemos as trilhas τ_1 e τ_2 se deslocando na direção da aresta e_2 , reproduzindo a situação

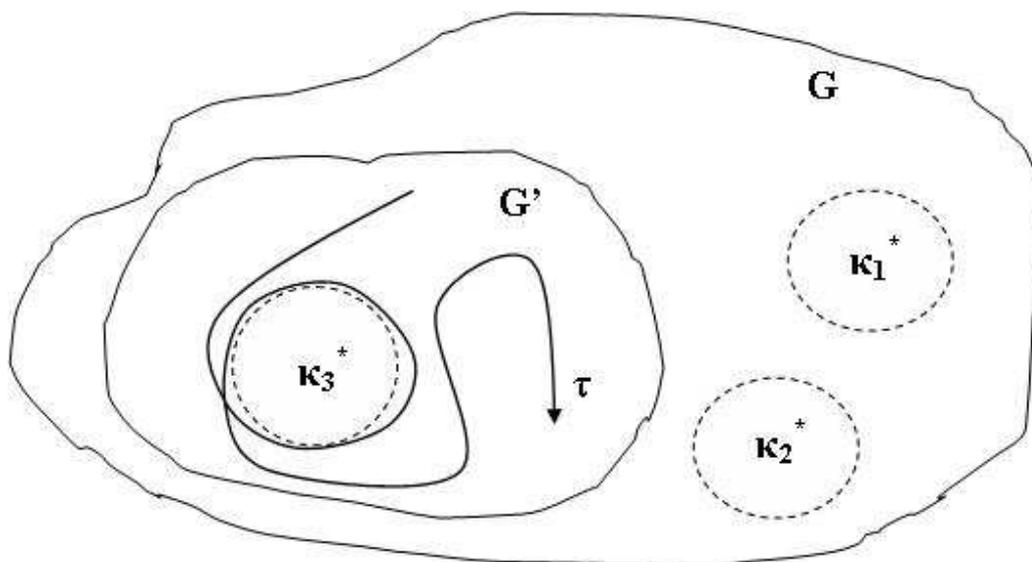


Figura 4.7: A trilha τ é infinita e contida em G' . Logo, G' contém um ciclo principal: κ_3^* .

anterior. Em (e), todas as arestas que inicialmente estavam orientadas no sentido do movimento de τ_1 , se tornaram trilhas infinitas que percorrem o ciclo.

As trilhas τ_1 , τ_2 e τ_3 são infinitas e definem completamente os sumidouros de κ (assumindo que, inicialmente, não havia outras arestas orientadas no mesmo sentido que τ_1). Além disso, estes sumidouros são sumidouros do grafo como um todo e, portanto, teremos uma execução independente. Como κ é simples, este é o próprio ciclo principal. Se não fosse, este conteria o ciclo principal.

A prova formal segue abaixo. Esta afirma que se, para qualquer tempo arbitrariamente grande escolhido, existe pelo menos uma trilha que percorre G' durante esse tempo, então um dos ciclos principais está em G' .

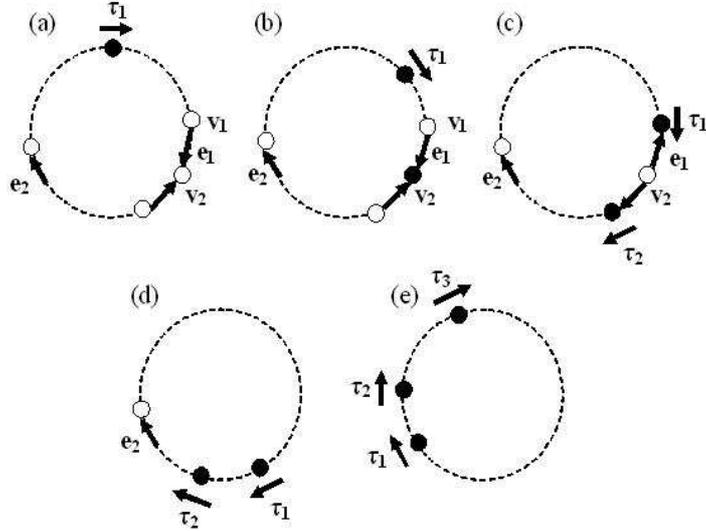


Figura 4.8: Exemplo esquemático onde uma trilha infinita leva o ciclo a executar de forma independente.

Teorema 4.6. *Considere um grafo $G = (N, E)$ qualquer executando o ERA na sua fase periódica de tamanho p iniciada no tempo t^p e um subgrafo G' de G . Se, para todo inteiro $\eta > 0$ arbitrariamente grande, existe uma trilha $\tau' \in \Gamma(G)$ tal que $\tau'(t) \in G'$, $t^p \leq t \leq \eta$, então existe $\kappa^* \in K^*(G)$ tal que $\kappa^* \subseteq G'$, ou seja, G' contém pelo menos um ciclo principal.*

Demonstração. Como G' tem tamanho finito, podemos escolher η suficientemente grande para que existam dois inteiros $k_1, k_2 \geq 0$ tais que $k_1 \neq k_2$, para os quais $\tau'(t^p + k_1p) = \tau'(t^p + k_2p)$. Desta forma, pelo comportamento periódico do ERA, é possível indentificar o ciclo $\kappa = (v_0, \dots, v_{|\kappa|-1})$ cujos nós correspondem àqueles percorridos por τ' desde de $\tau'(t^p + k_1p)$ até $\tau'(t^p + k_2p - 1)$ (todos percententes a G' , por premissa) e dizer que este é percorrido infinitamente e num mesmo sentido por uma trilha τ . Ou seja, para $t_0 = t^p + k_1p$, temos que $\tau(t_0 + i) = v_i$, para todo $i \geq 0$.

Passamos a provar que κ apresenta execução independente. Considere o ciclo κ , a trilha τ e o nó $\tau(t_0) = v_0$. Tome o menor valor d_1 , onde $1 < d_1 \leq |\kappa| - 1$, para o qual existe uma aresta orientada $\overrightarrow{v_{d_1} v_{d_1+1}}$. Se $d_1 = |\kappa| - 1$, então esta aresta é adjacente a $\tau(t_0)$ e, portanto, τ será a única trilha percorrendo κ infinitamente, não haverá outros sumidouros em κ e estará caracterizada uma execução independente, já que todos os sumidouros de κ também serão sumidouros de G .

Porém, se $1 < d_1 \leq |\kappa| - 2$, então, pelo Lema 4.5, teremos a formação, a partir de um certo tempo t_0^1 , de uma nova trilha tal que $\tau_1(t_0^1 + i) = \tau(t_0^1 + d_1 + i)$ para todo $i \geq 0$. Porém, em relação a τ_1 , poderemos agora ter outra aresta nas mesmas condições citadas, ou seja, a mais próxima aresta orientada no sentido do percorrimto de κ por τ_1 . Para esta aresta, a mesma situação ocorrerá, ou seja, a indução de outra trilha τ_2 à frente d_2 passos de τ_1 .

Pode-se utilizar este raciocínio reiteradamente até que, a partir de um tempo $t_0^* > t_0^x$ (onde x é o número de trilhas criadas nesse processo), nenhuma trilha mais seja criada em κ . Note que é possível garantir que as trilhas não são criadas indefinidamente, já que o número de nós de κ é finito. Haverá portanto, a partir de t_0^* , um conjunto $T = \{\tau, \tau_1, \dots, \tau_q\}$ de trilhas infinitas percorrendo o ciclo κ num mesmo sentido. Note que as únicas arestas orientadas no sentido do percorrimto de κ pelas trilhas de T são exatamente aquelas adjacentes a estas mesmas trilhas. Caso houvesse outra aresta orientada neste mesmo sentido, pelo Lema 4.5, outra trilha seria criada e teríamos um valor maior para t_0^* . Por isso, todos os sumidouros de κ serão sumidouros de G , já que um novo sumidouro em κ implicaria uma aresta orientada no sentido do percorrimto e esta também geraria outra trilha. Pelo Lema 4.3,

podemos afirmar que κ apresenta execução independente e, pelo Teorema 4.4, temos que existe κ^* tal que $\kappa^* \subseteq \kappa \subseteq G'$ e $\kappa^* \in K^*(G)$. \square

4.5.2 Comportamento das Trilhas no Período do ERA

Uma trilha não surge espontaneamente. Cada participante de uma trilha é sumidouro em um determinado passo do algoritmo porque pelo menos um outro nó, seu vizinho, foi sumidouro no passo anterior e, assim, sucessivamente. Na verdade, todas as trilhas do sistema (ou, por outro lado, todos os sumidouros) têm sua origem mais remota nos sumidouros que o grafo continha na orientação acíclica inicial da execução do ERA.

Para ser possível explicar o que o Teorema 4.7 diz, será necessária uma discussão prévia sobre trilhas gêmeas. Primeiramente, considere um grafo qualquer $G = (N, E)$ executando o ERA sob o escalonamento $\sigma = (\omega_1, \omega_2, \dots)$, cuja fase periódica de tamanho p inicia-se, sem perda de generalidade, em ω_1 . Assuma um nó qualquer $v \in N$ sumidouro em ω_t , para algum $t \geq 0$. Sabemos que, na fase periódica, se $v \in \text{sinks}(\omega_t)$ então $v \in \text{sinks}(\omega_{t+p})$.

Assim, se uma trilha τ_0 alcança os nós v_1, v_2, \dots nos tempos $1, 2, \dots$, então haverá outra trilha τ_1 que alcançará a mesma seqüência de nós nos tempos $1 + p, 2 + p, \dots$. Ou seja, se existe τ_0 tal que $\tau_0(t) = v_i, t > 0$, então existe τ_1 tal que $\tau_1(t + p) = v_i$. Assim, também haverá τ_2 tal que $\tau_2(t + 2p) = v_i$. Genericamente, haverá uma família de trilhas τ_i tais que $\tau_k(t + kp) = v_k, k \geq 0$. Vamos dizer que as trilhas τ_k são *trilhas gêmeas*.

Por isso, o Teorema 4.7 prova que, para toda operação de um nó $v \in N$ em um tempo t_r , vale uma das propriedades: (a) esta operação é resultado do alcance de uma trilha que se originou diretamente de um ciclo principal; ou (b) existe uma trilha gêmea à trilha que alcança v em t_r que se origina de um ciclo principal. Afirma-se que, para qualquer nó $v \in N$, sumidouro no

tempo t_r , existe pelo menos uma trilha τ que alcança um dos ciclos principais em um tempo $t > 0$ e, depois, alcança v em um tempo $t_r + kp > t$, $k \geq 0$.

Na verdade, no início da fase periódica, algumas trilhas são ainda remanescentes da fase pré-periódica e podem não ter sido originadas em nenhum ciclo principal. Porém, estas trilhas não são infinitas (se fossem, pelo Teorema 4.6, conteriam um ciclo principal) e, por serem finitas, estas vão terminando, até que restem apenas trilhas gêmeas destas, todas originadas dos ciclos principais.

Observe que várias propriedades são comuns a um conjunto de trilhas gêmeas e este fato é utilizado nas provas a seguir.

Na Figura 4.9 é apresentada uma visualização da situação mais simples descrita, onde todas as trilhas e operações são originárias do ciclo principal.

Na figura 4.10 é apresentada uma situação mais realista. Note que o eixo x representa o tempo e, neste, são marcados: o tempo t^p , onde é iniciada a fase periódica do ERA; o tamanho p do período; e os tempos t , $t + p$ e $t + 2p$, onde um determinado nó opera repetidamente com intervalos de p passos entre cada operação (de certa forma, poderíamos considerar tais operações como “gêmeas”, pois são o resultado da repetição natural da fase periódica do ERA). Acima do eixo x são plotadas as estruturas de interesse, que são repetidas em cada período para refletir o comportamento periódico: um ciclo principal κ^* qualquer; o nó que opera nos tempos t , $t + p$ e $t + 2p$; e as trilhas que alcançam este nó nestes tempos.

O que queremos destacar é que, dada a operação de um nó dentro do período, a trilha que o alcança pode simplesmente ter sua origem em um ciclo principal (situação simples da figura 4.9). Porém, pode ser que isso não ocorra, como no caso da primeira operação do nó da figura 4.10, já que nas primeiras operações, algumas trilhas podem ser originadas da fase

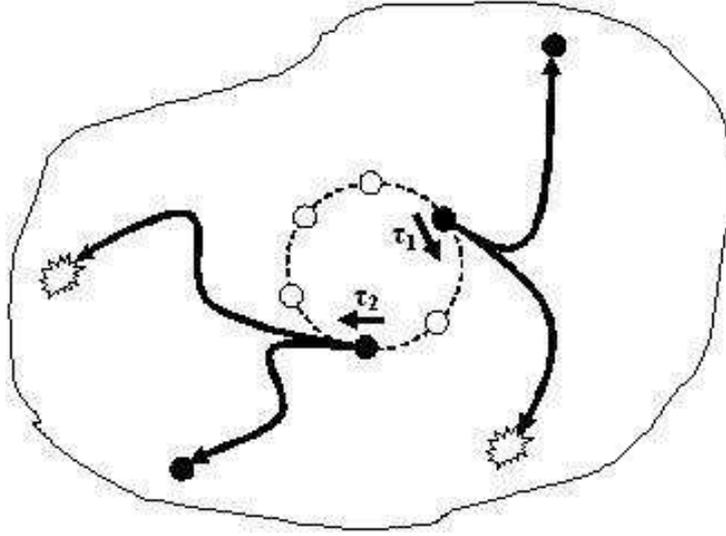


Figura 4.9: Exemplo esquemático onde, já na fase periódica do ERA, todas as trilhas e operações se originam no ciclo principal.

pré-periódica. Porém, mesmo nestes casos, outras trilhas alcançarão o nó nas suas operações “gêmeas”, de tal forma que a parte destas trilhas que ocorrem já dentro do período são idênticas, o que garante as propriedades que nos interessam.

Abaixo, é apresentada a prova formal.

Teorema 4.7. *Considere um grafo $G = (N, E)$ executando o ERA sob escalonamento $\sigma = (\omega_1, \omega_2, \dots)$ cuja fase periódica de tamanho p se inicia no tempo $t^p \geq 0$ e o conjunto $K^*(G)$ dos ciclos principais de G . Dados um inteiro qualquer $t_r \geq t^p$ e um nó qualquer $v \in N$, não pertencente a nenhum ciclo principal, temos que, se $v \in \text{sinks}(\omega_{t_r})$, então existem uma trilha $\tau \in \Gamma(G)$, um ciclo principal $\kappa \in K^*(G)$ e um inteiro $k > 0$ para os quais $\tau(t_r + kp) = v$ e $\tau(t) \in \kappa$ para algum $t < t_r + kp$.*

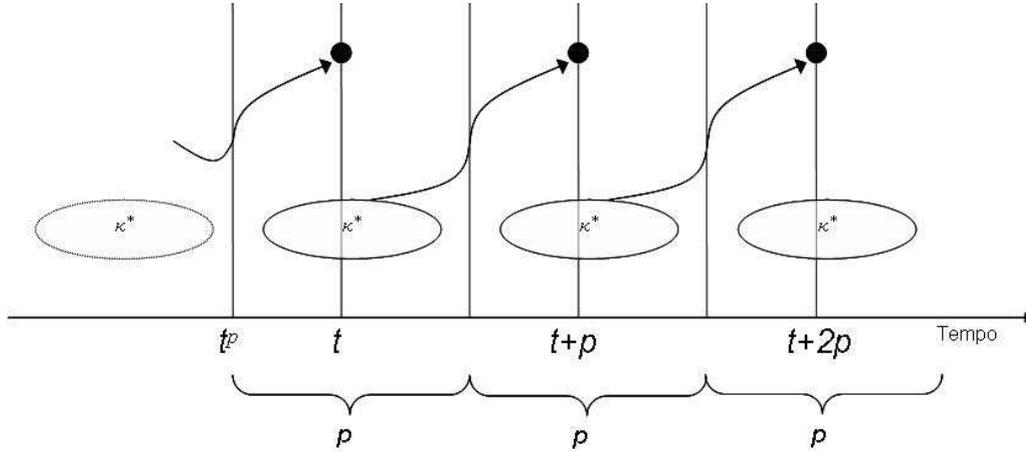


Figura 4.10: Exemplo esquemático sobre trilhas gêmeas e sua origem do ciclo principal.

Demonstração. Considere a família de trilhas gêmeas no formato τ_k tais que $\tau_k(t_r + kp) = v$, $k \geq 0$. É preciso, portanto, provar que, para pelo menos uma trilha desta família, existe um inteiro $t < t_r + kp$ e um ciclo principal $\kappa \in K^*(G)$ tais que $\tau_k(t) \in \kappa$.

A prova é por absurdo. Assuma por hipótese que existe um nó $v \in N$ para o qual não existem $\kappa \in K^*(G)$, $k \geq 0$ e $t < t_r + kp$ tais que $\tau_k(t) \in \kappa$.

Se nenhuma das trilhas da família τ_k tem interseção com os ciclos principais, então considere o grafo $\overline{G} = (\overline{N}, \overline{E})$ tal que $\overline{G} \subseteq G$, onde \overline{N} é composto dos nós que não estão em nenhum ciclo principal e \overline{E} das arestas adjacentes a 2 nós de \overline{N} . Podemos afirmar, para quaisquer $k \geq 0$ e $t \geq t^p$, que $\tau_k(t) \in \overline{G}$. Desta forma, para qualquer inteiro η , arbitrariamente grande, existe pelo menos uma trilha τ_k para a qual $\tau_k(i) \in \overline{G}$, para todo inteiro i tal que $t^p \leq i \leq \eta$. Assim, pelo Lema 4.6, temos que existe $\kappa \in K^*(G)$ tal que $\kappa \subseteq \overline{G}$, o que é uma contradição com a definição de \overline{G} . \square

4.6 Novas Propriedades do ERA

4.6.1 Ciclos Principais Determinam Máximo Intervalo Inoperante

Como vimos no Capítulo 2 - Conceitos Básicos, dado um grafo $G = (N, E)$ executando o ERA denotamos por $\lambda(G)$ o número do maior nível obtido na decomposição por sumidouros do grafo. Ou seja, se há 4 níveis na decomposição, consideramos $\lambda(G) = 3$, pois considera-se como 0 o rótulo do primeiro nível. Também sabemos que, em uma execução do ERA, cujo escalonamento seja representado pelas orientações acíclicas $\omega_0, \omega_1, \omega_2, \dots$, temos que $\lambda(\omega_i) \leq \lambda(\omega_{i-1}), \forall i > 0$. Também é verdade que, dado o menor inteiro $p \geq 0$, tal que ω_p encontra-se na fase periódica do ERA, temos que para todo inteiro $i \geq p$ é verdade que $\lambda(\omega_{i+1}) = \lambda(\omega_i)$.

Para um nó qualquer $v \in N$, definimos $\lambda(v)$ como o maior nível da decomposição por sumidouros que o nó v atinge (sempre assumindo o primeiro como nível 0). É claro que $\lambda(v) = \max_{i>0} t_v^i - t_v^{i-1}$, onde t_v^i corresponde à i -ésima operação de v . Portanto $\lambda(v)$ equivale ao maior tempo que o nó v fica sem operar, ou seja, o máximo intervalo inoperante de v .

Também será útil a definição a seguir, que torna o tratamento mais simples. Para um subgrafo qualquer G' de G , a grandeza $\lambda(G')$ pode ser definida assim: $\lambda(G') = \max_{v \in G'} \lambda(v)$. Imediatamente, deriva-se que, dado um nó $v \in G'$, temos que $\lambda(v) \leq \lambda(G')$. Por último, também será definido o mesmo tratamento para um conjunto de grafos ou subgrafos, como em $\lambda(K^*(G))$, por exemplo, onde $\lambda(K^*(G)) = \max_{\kappa \in K^*(G)} \lambda(\kappa)$.

Sabemos também que, para todo nó $v \in \kappa$ onde $\kappa \in K^*(G)$, temos que $\lambda(v) = \lambda(\kappa)$. Isso acontece porque, pela definição, $\lambda(v) \leq \lambda(\kappa)$ e em função das características da execução canônica, pois dado que as distâncias entre

as trilhas que percorrem os ciclos principais numa execução canônica não mudam e, claramente, a maior entre tais distâncias determinam $\lambda(\kappa)$. Além disso, como as trilhas passam por todos os nós do ciclo, é claro que cada um dos nós fica sem operar $\lambda(\kappa)$ passos em algum momento.

Teorema e Prova

Resumidamente, o que queremos provar é que $\lambda(v) \leq \lambda(K^*(G))$ para todo nó $v \in G$.

A idéia é que, se toda trilha origina-se de um dos ciclos principais e, no seu caminho, o intervalo entre duas operações consecutivas de cada nó alcançado é menor ou igual ao dos nós anteriores, então é claro que, para cada nó $u \in G$ que fique um tempo igual a $\lambda(u) = t_u^j - t_u^{j-1}$ sem operar, é possível descobrir um dos nós v de um ciclo principal do qual partiu uma trilha que alcançou u no tempo t_u^j para o qual $t_v^i - t_v^{i-1} \geq t_u^j - t_u^{j-1}$. Como $\lambda(v) \geq t_v^i - t_v^{i-1}$, logo $\lambda(v) \geq \lambda(u)$. Segue-se a prova formal.

Teorema 4.8. *Dado um grafo $G = (N, E)$ executando o ERA na fase periódica, temos que $\lambda(v) \leq \lambda(K^*(G))$, para qualquer nó $v \in N$.*

Demonstração. Considere um nó qualquer $v \in N$ e a definição de $\lambda(v) = \max_{j>0} t_v^j - t_v^{j-1}$. Tome um valor de j que maximiza tal expressão, ou seja, $\lambda(v) = t_v^j - t_v^{j-1}$. Por definição, em t_v^j , o nó v é um sumidouro. Logo, pelo Teorema 4.7 existe pelo menos uma trilha $\tau \in \Gamma(G)$, um ciclo principal $\kappa \in K^*(G)$ e um inteiro $k \geq 0$ tais que $\tau(t_v^j + kp) = v$ e $\tau(t) \in \kappa$ para algum $t < t_v^j + kp$.

Tomando-se o nó $u = \tau(t) \in \kappa$, assumamos $t_u^i = t$ para algum $i > 0$, ou seja, que a operação deste nó na sua participação em τ foi sua i -ésima operação. Assim, tomando-se t_u^{i-1} , o tempo da sua $(i-1)$ -ésima operação, temos pelo

Teorema 4.1 que $t_u^i - t_u^{i-1} \geq t_v^j - t_v^{j-1}$. Como, por definição $\lambda(K^*(G)) \geq t_u^i - t_u^{i-1}$ e $t_v^j - t_v^{j-1} \geq \lambda(v)$, por transitividade, temos que $\lambda(K^*(G)) \geq \lambda(v)$. \square

Por este teorema vemos que os ciclos principais, além de determinarem a concorrência do sistema, também determinam o tempo máximo que um nó fica sem operar.

Observe que, por outro lado, nem todo nó em um sistema representado por um grafo $G = (N, E)$ obrigatoriamente chega a ficar $\lambda(G)$ passos sem operar, sendo este um limite superior. Na Figura 4.11 temos um contra-exemplo.

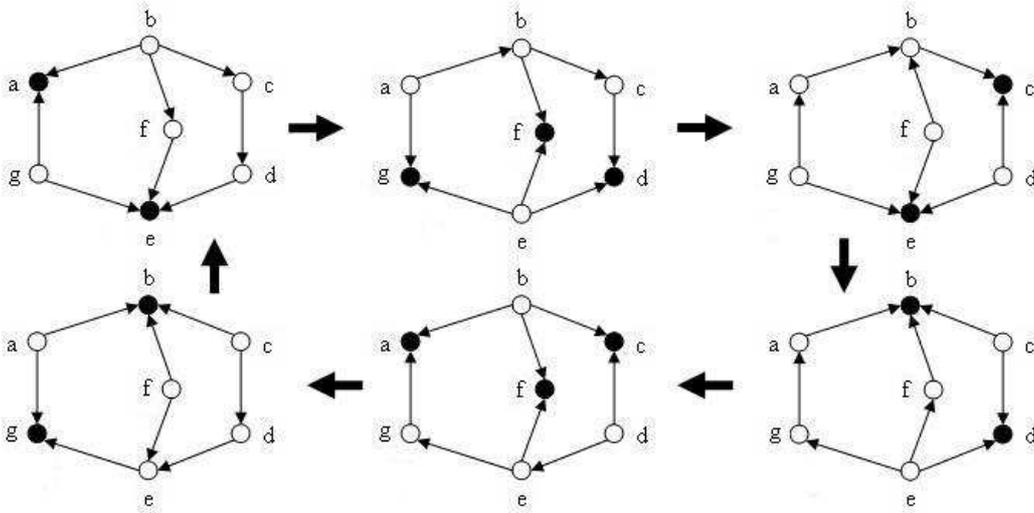


Figura 4.11: Exemplo em que o maior intervalo inoperante do grafo é diferente de um dos nós (nesse caso, o nó f). Ou seja, $\lambda(G) \neq \lambda(f)$.

Executando o ERA no sistema representado pelo grafo $G = (N, E)$ da Figura 4.11, verificamos que $\lambda(G) = 3$ e $\lambda(f) = 2$.

Uma consequência imediata do teorema acima é a relação entre o máximo intervalo inoperante com propriedades do maior ciclo principal de G , como

o número de arestas orientadas no sentido horário (ou anti-horário) e o tamanho do ciclo. Desta forma, é possível ter uma estimativa para o máximo intervalo inoperante a partir de características estáticas do sistema.

Corolário 4.9. *Considere um grafo $G = (N, E)$ executando o ERA na sua fase periódica e a orientação acíclica inicial ω_1 de G . Assuma $\xi(\kappa^*, \omega_1) = \min \{n^+(\kappa^*, \omega_1), n^-(\kappa^*, \omega_1)\}$. Dado $\kappa^* = \{\kappa \in K^*(G) \mid |\kappa| = \max_{x \in K^*(G)} (|x|)\}$, temos que $|\kappa^*| - 2\xi(\kappa^*, \omega_1) + 1 \geq \lambda(K^*(G)) \geq \left\lceil \frac{|\kappa^*|}{\xi(\kappa^*, \omega_1)} \right\rceil$.*

Demonstração. Basta notar que o menor valor que $\lambda(K^*(G))$ poderá alcançar é aquele que divide o maior ciclo principal em $\xi(\kappa^*, \omega_1)$ intervalos iguais, já que, pela expressão da concorrência, cada nó de κ^* irá operar $\xi(\kappa^*, \omega_1)$ vezes a cada $|\kappa^*|$ passos. Assim, para dividir o ciclo κ^* em intervalos iguais, temos $\lambda(K^*(G)) \geq \left\lceil \frac{|\kappa^*|}{\xi(\kappa^*, \omega_1)} \right\rceil$, onde o teto é acrescido para os casos em que não são múltiplos (ver Figura 4.12b).

Da mesma forma, o maior valor de $\lambda(K^*(G))$ é alcançado quando todas as trilhas do maior ciclo principal são localizadas consecutivamente no ciclo. A distância mínima entre 2 trilhas consecutivas é de 2 nós. Nesta seqüência de sumidouros, os nós não sumidouros localizados entre 2 sumidouros não contam para o intervalo máximo inoperante. Porém, um destes nós é contabilizado (ver Figura 4.12a). Portanto, temos $\lambda(K^*(G)) \leq |\kappa^*| - 2\xi(\kappa^*, \omega_1) + 1$.

Assim, pelo Teorema 4.8, temos que $\lambda(K^*(G)) = \lambda(G)$ e, portanto, $|\kappa^*| - \xi(\kappa^*, \omega_1) - 1 \geq \lambda(K^*(G)) \geq \left\lceil \frac{|\kappa^*|}{\xi(\kappa^*, \omega_1)} \right\rceil$. □

Analizando o Crescimento do Máximo Intervalo Inoperante

Em um sistema executando de forma distribuída, a execução do ERA depende da geração de uma orientação acíclica inicial. Alguns algoritmos

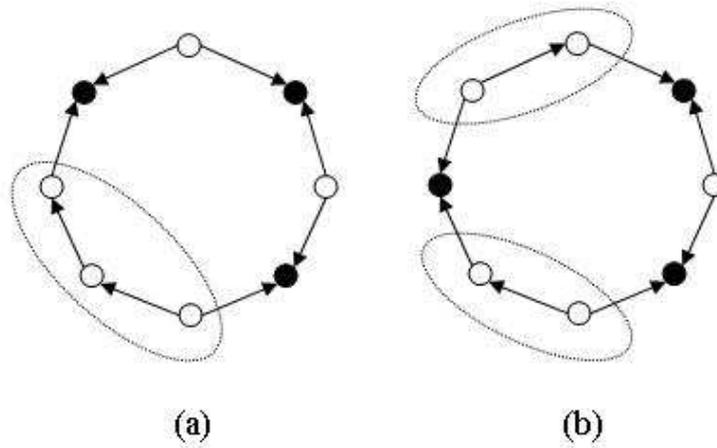


Figura 4.12: Exemplos de máximo intervalo inoperante assumindo ciclo principal κ com concorrência $\frac{3}{8}$. Em (a), as trilhas são consecutivas e $\lambda(G) = 3$, o valor máximo. Em (b), as trilhas estão dispostas o mais uniformemente possível e $\lambda(G) = 2$, o valor mínimo.

probabilísticos e distribuídos para geração de orientações acíclicas foram introduzidos em CALABRESE[13] e ARANTES[1]. Entre estes, destaca-se o algoritmo Alg-Cor, introduzido em ARANTES[1] e cuja velocidade de convergência é discutida no Capítulo 5 - Análise de Algoritmos Randômicos para Inicialização do ERA.

Não queremos aprofundar a discussão sobre as características do Alg-Cor neste trabalho. Por agora, basta saber que o Alg-Cor é um algoritmo probabilístico e distribuído que realiza uma $(\Delta + 1)$ -coloração no grafo, orientando as arestas da maior para a menor cor. E que, dentre os algoritmos para geração de orientações acíclicas comparados [1][2], este provou, experimentalmente, ser aquele que gera as maiores concorrências.

Inicialmente, o valor de $\lambda(G)$ é dado pela coloração gerada. Porém, até alcançar o período, este valor pode decrescer [8][7]. O máximo intervalo inoperante que nos interessa é aquele alcançado no período.

Assim, para que fosse possível ter uma idéia do crescimento de $\lambda(G)$ de acordo com o número de nós, denotado aqui como $n = |N|$ (considerando $G = (N, E)$), foram realizadas algumas simulações com grafos aleatórios. Para cada grafo construído aleatoriamente com n vértices, foi gerada uma orientação acíclica utilizando Alg-Cor. A partir desta orientação, foi executado o ERA e computou-se o valor de $\lambda(G)$ após a entrada no período. Assim, foi possível calcular $E[\lambda(G)|n = x]$, o valor esperado do máximo intervalo inoperante para grafos com número variado de nós, mostrado no eixo x do gráfico da Figura 4.13.

São apresentadas 3 seqüências diferentes: uma em que cada nó tem, em média, n arestas ($\bar{\Delta} = n$); outra com $\bar{\Delta} = \log_2 n$; e outra onde $\bar{\Delta} = 6$.

O interesse nestes diferentes padrões de conectividade se deve ao fato de que, em certas aplicações práticas, como as que distribuem os nós em áreas planas e cujo compartilhamento de recursos é fortemente determinado pela vizinhança física (como nas redes de sensores, por exemplo), o sistema tende a crescer segundo um regime mais ou menos constante de arestas por nó, similarmente a um grafo planar. O interesse é, portanto, na forma como o máximo intervalo inopante cresce nesse tipo de aplicação.

Vemos que o crescimento de $\lambda(G)$ é linear quando $\bar{\Delta} = n$. Porém, este é sub-linear não apenas para $\bar{\Delta} = 6$, como também para $\bar{\Delta} = \log_2 n$. Desta forma, para o tipo citado de aplicação, concluímos que o máximo intervalo inoperante tende a crescer sub-linearmente com o tamanho do sistema, um resultado que indica uma grande aplicabilidade do ERA em aplicações deste

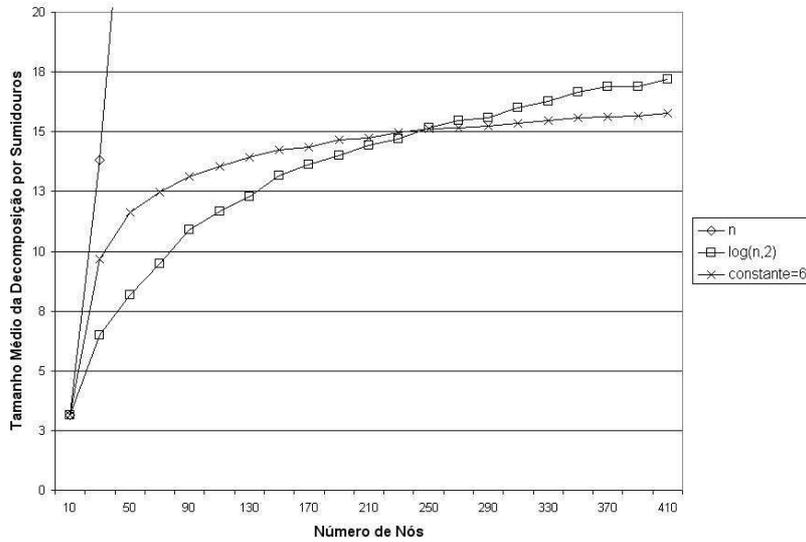


Figura 4.13: Avaliação do crescimento de $\lambda(G)$ em função de n para vários valores de Δ médio.

tipo, principalmente porque estamos assumindo uma estrutura totalmente distribuída, incluindo o algoritmo de geração inicial de orientações acíclicas.

Por outro lado, se for possível construir um algoritmo distribuído que limite a vizinhança aceita por um nó em um determinado número de arestas seria possível delimitar *a priori* e de forma totalmente distribuída o máximo intervalo inoperante do sistema. O desafio seria garantir a conexão do grafo.

4.6.2 Ciclos Principais Determinam Mínimo Intervalo Inoperante

A seguir é provado que os ciclos principais também determinam o menor intervalo inoperante de todos os nós do grafo. Dado um grafo $G = (N, E)$, de forma similar à seção anterior, chamamos o menor intervalo inoperante

de um nó $v \in N$ de $\lambda^-(v)$ e o definimos como $\lambda^-(v) = \min_{i>0} t_v^i - t_v^{i-1}$. Para qualquer subgrafo G' de G , definimos que $\lambda^-(G') = \min_{v \in G'} \lambda^-(v)$. A definição para conjuntos também é válida, logo $\lambda(K^*(G)) = \min_{\kappa \in K^*(G)} \lambda(\kappa)$.

Teorema 4.10. *Dado um grafo qualquer $G = (N, E)$ executando o ERA na sua fase periódica, temos que $\lambda^-(v) \geq \lambda^-(K^*(G))$, para qualquer nó $v \in N$.*

Demonstração. Esta prova é muito similar àquela apresentada para o Teorema 4.8. Basta verificar que $\lambda^-(v) = \min_{j>0} t_v^j - t_v^{j-1}$ e que, pelo Teorema 4.7 e pelo Teorema 4.2, este é limitado inferiormente por $\lambda^-(K^*(G))$. \square

4.6.3 Maior Número de Operações Independentes

Um corolário do Teorema 4.10 diz respeito ao maior número de vezes que um nó pode operar sem que outro opere. Dados dois nós quaisquer $v, u \in N$, não se trata aqui de comparar os números acumulados de operações entre v e u , mas de saber quantas vezes, no máximo, v opera sem que u tenha operado e vice-versa.

Esse tipo de preocupação é importante para aplicações em que é importante a garantia de operações alternadas. Basicamente, nestes casos, não seria útil operar várias vezes sem que cada outro nó sistema tenha operado. Menores números de operações independentes correspondem a altas taxas de alternância. Estas tendem a reduzir a probabilidade de existência de clusters temporais na operação dos nós, ou seja, que uma determinada região tenha vários nós operando enquanto outra se mantém menos ativa no intervalo.

Observe que também não se trata de reduzir a concorrência a um mínimo, como ter cada nó operando por vez no sistema. Estamos falando de priorizar a alternância e, assim, numa execução com concorrência $\frac{1}{2}$, por exemplo,

teríamos excelentes concorrência e alternância, pois metade do sistema opera de cada vez.

Apresentamos um corolário simples que limita superiormente esta grandeza pelas características dos ciclos principais.

Corolário 4.11. *Assuma um grafo qualquer $G = (N, E)$ executando o ERA na sua fase periódica e dois nós quaisquer $v, u \in N$. Considere as seqüências $T_v = t_v^0, t_v^1, \dots$ e $T_u = t_u^0, t_u^1, \dots$ dos tempos nos quais os nós v e u , respectivamente, operam. Dado um par de tempos consecutivos t_v^{i+1} e t_v^i , $i \geq 0$, em T_v e a seqüência $T'_u = t_u^j, t_u^{j+1}, \dots, t_u^{j+q-1}$, $j \geq 0$ e $q > 0$, dos tempos tais que, para $0 \leq x < j + q$, $t_v^i \leq t_u^x < t_v^{i+1}$, ou seja, os tempos em que u opera no intervalo compreendido entre $t_v^{i+1} - 1$ e t_v^i , temos que a quantidade q de operações neste intervalo é dado por $q \leq \left\lfloor \frac{\lambda(K^*)}{\lambda^-(K^*)} \right\rfloor$.*

Demonstração. É fácil perceber que o número q de vezes que u opera sem que v opere é menor ou igual ao total de vezes que u pode operar com distância mínima $\lambda^-(u)$ entre as operações, dentro do máximo intervalo $\lambda(v)$ que v fica sem operar. Logo, $q \leq \left\lfloor \frac{\lambda(v)}{\lambda^-(u)} \right\rfloor$. Pelo Teorema 4.8, temos que $\lambda^-(u) \geq \lambda(K^*(G))$ e, pelo Teorema 4.10, $\lambda(v) \leq \lambda^-(K^*(G))$. Logo, $q \leq \left\lfloor \frac{\lambda(v)}{\lambda^-(u)} \right\rfloor \leq \left\lfloor \frac{\lambda(K^*)}{\lambda^-(K^*)} \right\rfloor$. \square

4.6.4 Ciclos Principais com Trilhas Consecutivas Equidistantes

Uma propriedade interessante ocorre quando temos ciclos principais percorridos por trilhas consecutivas equidistantes. Este resultado é similar ao obtido em BARBOSA[8] que caracteriza as orientações cujas concorrência é dada por $\gamma = \frac{1}{p}$ ($m = 1$), ou seja, cada nó opera exatamente 1 vez no período.

A diferença é que, aqui, não caracterizamos a orientação acíclica como um todo, mas a orientação do ciclo principal no período.

No caso apresentado, consideramos q trilhas percorrendo os ciclos principais e com distância d entre si e provamos que o grafo se comporta de tal forma que todos os nós operam exatamente de d em d passos.

Teorema 4.12. *Considere um grafo qualquer $G = (N, E)$ executando o ERA na sua fase periódica de tamanho p . Assuma que cada ciclo $\kappa \in K^*(G)$, formado pelos nós $v_0, v_1, \dots, v_{|\kappa|-1}$, é percorrido por um conjunto $\Gamma = \{\tau_0, \tau_2, \dots, \tau_{q-1}\}$, $q > 0$, de trilhas consecutivas. Se existe $d > 1$ tal que, para todo par $\tau, \tau' \in \Gamma(G)$ de trilhas consecutivas e para todo $t, i \geq 0$, for verdade que $\tau(t) = v_i$ e $\tau'(t) = v_{i+d}$, então todos os nós de G operam 1 vez a cada d passos.*

Demonstração. Considere um nó $v \in G$ qualquer, não pertencente a nenhum ciclo principal, tal que v opera em um tempo qualquer t_v^i , $i > 0$. Sabemos pelo Teorema 4.7 que existe uma trilha τ , um ciclo principal $\kappa \in K^*(G)$ e um inteiro $k > 0$ tais que $\tau(t_v^i + kp) = v$ e $\tau(t) \in \kappa$ para algum $t < t_v^i + kp$, ou seja, uma operação equivalente ocorre quando v é alcançado por τ no tempo $t_v^i + kp$ e esta trilha origina-se de um ciclo principal. Assumimos, por simplificação que $t_v^{i+\delta} = t_v^i + kp$, ou seja, que o alcance de v por τ provoca a sua $(i + \delta)$ -ésima operação. Desta forma, pela periodicidade do ERA, temos que $t_v^{i+\delta+1} - t_v^{i+\delta} = t_v^{i+1} - t_v^i$.

Assuma um nó u , pertencente a um dos ciclos principais, tal que $\tau(t_u) = u$ para algum $t_u^j < t_v^{i+\delta}$ e o tempo $t_u^{j+1} > t_u^j + 1$ da próxima operação de u . Pela premissa da distância entre trilhas consecutivas dos ciclos principais, é fácil perceber que u opera a cada d passos, ou seja, $t_u^{j+1} - t_u^j = d$ e, além disso, que $\lambda(K^*(G)) = d$. Considere agora, o tempo $t_v^{i+\delta+1} > t_v^{i+\delta} + 1$ da próxima operação de v . Pelo Teorema 4.2, sabemos que $t_v^{i+\delta+1} - t_v^{i+\delta} \geq d$. Porém,

$t_v^{i+\delta+1} - t_v^{i+\delta} \leq \lambda(v) \leq \lambda(K^*(G)) = d$, logo $t_v^{i+\delta+1} - t_v^{i+\delta} = d$. Como esse mesmo raciocínio é aplicável para todos os nós e todas as operações de cada nó, então está claro que todos os nós operam 1 vez a cada d passos. \square

Capítulo 5

Análise de Algoritmos

Randômicos para Inicialização do ERA

5.1 Introdução

Em ARANTES[1] e ARANTES[2], foram estudados vários algoritmos randômicos e distribuídos para geração de orientações acíclicas. O objetivo principal era preencher o requisito inicial para a execução do ERA, a saber: a existência de uma orientação acíclica no grafo de dependências. Por isso, os algoritmos eram distribuídos. Foi também escolhido utilizar algoritmos para sistemas anônimos, o que garante grande abrangência, já que não é necessário nenhum conhecimento prévio da topologia da rede e de outras propriedades, como número de nós ou arestas. A utilização de sistemas anônimos levou à necessidade de utilização de algoritmos randômicos para quebra da simetria. Todos os algoritmos, em última análise, são extensões do algoritmo básico definido em CALABRESE[13].

Os estudos se concentraram em duas propriedades principais dos algoritmos: velocidade de convergência; e qualidade das orientações geradas.

Neste presente trabalho são apresentadas análises estendidas da velocidade de convergência dos algoritmos. Na Seção Algoritmo Calabrese/França, são reproduzidos o algoritmo básico Calabrese/França [13], assim como algumas análises sobre sua velocidade de convergência ARANTES[1]. Na Seção Alg-Viz, o algoritmo Alg-Viz [1][2] é reproduzido e são apresentadas novas análises sobre o seu tempo de execução, além de ser introduzida uma nova forma de polarização que é uma generalização para $f \geq 2$ daquela proposta por Calabrese/França para $f = 2$. Nesta seção, também é apresentado rapidamente o Alg-Cor [1][2], algoritmo utilizado para análise no Capítulo 4 - Trilhas e Propriedades do ERA e que apresenta a mesma velocidade de convergência de Alg-Viz. Os principais resultados são relativos à sub-exponencialidade em $n = |N|$ da convergência do algoritmo para grafos completos com utilização de qualquer número de faces $f \geq 2$. Também é demonstrado que, quando $f = n$, o algoritmo tem convergência linear. Estes resultados vêm se somar ao já conhecido anteriormente [13] de que a convergência destes algoritmos é $O(\log n)$ para grafos esparsos.

Na Seção Alg-Arestas são apresentadas análises detalhadas sobre a distribuição probabilística do tempo de execução do algoritmo Alg-Arestas, introduzido em ARANTES[1] e ARANTES[2]. A velocidade de convergência do problema é representada por $P[T(m) = x]$, ou seja, a probabilidade de o algoritmo convergir em tempo x quando executado em um grafo de m arestas. Esta análise inclui a utilização de uma expressão [17] que calcula $P[T(m) \geq u(m)+w]$, a probabilidade de uma instância específica de execução do algoritmo se afastar um número $w > 0$ qualquer de passos de um determinado valor mais provável dado pela função $u(m)$. As análises utilizando as

ferramentas introduzidas em KARP[17] foram publicadas em artigos anteriores do autor em conjunto com outros pesquisadores [3][4]. Também é introduzida uma nova expressão, baseada em ARANTES[1], que apresenta maior acurácia na previsão dos fenômenos para todos os casos, incluindo aqueles em que w e f são pequenos, além de permitir a obtenção de $P[T(m) = x]$ para qualquer valor inteiro de x .

5.2 Algoritmo Calabrese/França

Este algoritmo randômico distribuído, introduzido em CALABRESE[13], é a base para os outros algoritmos apresentados aqui. A idéia é utilizar em cada nó uma espécie de moeda que gera aleatoriamente um resultado 0 ou 1 em cada sorteio.

As análises deste algoritmo já foram realizadas em CALABRESE[13], ARANTES[1] e ARANTES[2] de forma diferente e são retomadas aqui para fins de entendimento.

Todos os algoritmos apresentados neste capítulo são assíncronos. Porém, as análises serão realizadas, por questões de simplicidade, como se estes fossem síncronos, situação na qual um *relógio global* marca o tempo em que todos os nós passam para o próximo passo do algoritmo. Da mesma forma, assim também serão apresentadas as próprias descrições dos algoritmos. Desta forma, conceitos como passos do algoritmo fazem sentido dentro deste contexto.

5.2.1 Notação

Dado um grafo $G = (N, E)$, quer-se definir uma orientação acíclica para G como saída. O algoritmo é guloso, no sentido de que este não volta atrás após a orientação de uma aresta.

Durante a execução do algoritmo, alguns nós podem orientar todas as suas arestas e não participar mais da execução do algoritmo, enquanto outros continuam executando com o objetivo de orientar as suas arestas ainda não orientadas. Por isso, diz-se que os nós que continuam executando o algoritmo e que têm alguma aresta não orientada são probabilísticos, enquanto aqueles que já executaram o algoritmo são chamados determinísticos. Define-se, portanto, para uma execução do ERA em um grafo $G = (N, E)$, o conjunto $N_k \subseteq N$ dos nós probabilísticos após k passos do algoritmo, para k inteiro positivo. Dado um nó $v \in N$, denota-se $viz_{N_k}(v)$ o conjunto dos vizinhos probabilísticos de v . Denota-se $G_k = (N_k, E_k)$ o subgrafo de G induzido por N_k .

Como já foi mencionado anteriormente, os algoritmos trabalham com sorteios e, no caso de Calabrese/França, de tais sorteios obtêm-se resultado 0 ou 1, representando uma espécie de moeda. Desta forma, denotamos c_i^k o resultado obtido pelo nó $v_i \in N_k$ no k -ésimo passo do algoritmo.

5.2.2 Algoritmo Não Polarizado

O algoritmo Calabrese/França apresenta duas versões: uma uniforme; e outra polarizada. Na segunda, a probabilidade de obtenção de 0 e de 1 no sorteio dos dados são iguais a $\frac{1}{2}$. No caso da polarizada, define-se uma função de cálculo da probabilidade de um nó obter 0 ou 1 no sorteio.

Portanto, o algoritmo pode ser definido da forma geral apresentada a seguir:

Calabrese/França

Entrada: Grafo não orientado $G = (N, E)$.

Saída: Orientação acíclica ω sobre G .

Execução:

- (i) Todos os nós $v \in N_k$ realizam um sorteio, obtendo c_i^k .
- (ii) Um nó $v_i \in N_k$ é um vencedor quando $c_i^k = 1$ e $c_j^k = 0$ para todo $v_j \in viz_{N_k}(v_i)$. Nós vencedores orientam as arestas adjacentes ainda não orientadas na sua direção e param de executar o algoritmo, ou seja, $v_i \notin N_{k+1}$.
- (iii) O algoritmo pára quando $N_k = \emptyset$.

Desta forma, as versões uniforme e polarizada do algoritmo são idênticas, excetuando-se a distribuição de probabilidades do sorteio no passo 1.

No caso do algoritmo uniforme, $P(c_i^k = 1) = P(c_i^k = 0) = \frac{1}{2}$.

Como será observado, tal algoritmo é, na verdade, muito ineficiente. A análise abaixo se dá sobre a execução do algoritmo para um grafo completo genérico $G = (N, E)$.

Considere o evento S_i^k como sendo aquele que ocorre sempre que $v_i \in N_k$ é o vencedor no k -ésimo passo do ERA. Considere também $S^k = \bigcup_{v_i \in N_k} S_i^k$, indicando a probabilidade de algum nó ser vencedor no passo k . É garantido que, no máximo, um nó é o vencedor em cada passo, já que G_0 é completo, assim como todos os G_k gerados nos passos subseqüentes, já que um nó é retirado junto de todas as arestas adjacentes a ele. Desta forma, os eventos S_i^k

para um mesmo k são disjuntos. Além disso, é fácil perceber que $\Pr(S^{k+1}) \geq \Pr(S^k)$, já que $|N_{k+1}| \leq |N_k|$, para todo $k \geq 0$. Então, temos:

$$\Pr(S^k) = \Pr\left(\bigcup_{v_i \in N_k} S_i^k\right) \geq \Pr\left(\bigcup_{i=1}^n S_i^0\right) = \sum_{i=1}^n \Pr(S_i^0) = \sum_{i=1}^n \left(\frac{1}{2}\right) \left(\frac{1}{2}\right)^{n-1} = \frac{n}{2^n}$$

O número de repetições necessárias para até que um nó torne-se um vencedor pode ser modelado por uma variável aleatória com distribuição geométrica com expectativa igual a $\frac{2^n}{n}$. Se considerarmos que são necessários que n nós tornem-se determinísticos, então poderíamos, assumindo $\Pr(S^k) = \frac{n}{2^n}$ para todo k , esperar que o algoritmos convergisse em $(n-1) \cdot \frac{2^n}{n} = 2^n - \frac{2^n}{n}$ passos (o último nó torna-se determinístico conjuntamente com o penúltimo). Obviamente, que se trata de uma aproximação relativamente grosseira, já que $|N_k|$ tende para 2 no final do algoritmo, mas dá uma idéia da ineficiência o algoritmo, principalmente se comparado à versão seguinte.

5.2.3 Algoritmo Polarizado

Para melhorar a eficiência do algoritmo, propõe-se a utilização de moedas polarizadas, de tal forma que $\Pr(c_i^k = 1) = \frac{1}{|\text{viz}_{N_k}(v_i)|+1}$ e $\Pr(c_i^k = 0) = 1 - \Pr(c_i^k = 1)$. Observe que a intuição envolvida é a de que nós com mais vizinhos têm menos chances vencer o sorteio. Portanto, a idéia é que em uma determinada região exista uma tendência de que apenas um nó obtenha 1 no sorteio, aumentando as chances de haver um vencedor.

Em grafos completos, mesmo que cada nó não conheça o número máximo de nós probabilísticos do sistema, já que este é anônimo, teremos $\Pr(c_i^k = 1) = \frac{1}{|N_k|+1}$ e $\Pr(c_i^k = 0) = 1 - \frac{1}{|N_k|+1}$ para todo $v_i \in N_k$ e $k \geq 0$. Então teremos, considerando $n_k = |N_k|$:

$$\Pr(S^k) = \Pr\left(\bigcup_{v_i \in N_k} S_i^k\right) = \sum_{i=1}^{n_k} \Pr(S_i^k) = \sum_{i=1}^{n_k} \left(\frac{1}{n_k}\right) \left(1 - \frac{1}{n_k}\right)^{n_k-1} \geq \frac{n_k}{n_k \cdot e} = \frac{1}{e}$$

Assim, aproximadamente $e \cong 2,71$ tentativas devem ocorrer até que um dos nós se torne vencedor. Portanto, teremos $2,17n$ passos até a convergência do algoritmo.

As idéias deste algoritmo são estendidas, assim como as análises para os próximos algoritmos.

5.3 Alg-Viz

A primeira extensão do algoritmo Calabrese/França é tão somente a utilização para o sorteio de dados com f faces em vez de moedas e a definição do vencedor como aquele que tem o valor sorteado maior do que todos os vizinhos probabilísticos. Ou seja, trata-se de uma generalização, já que, Calabrese/França é equivalente a Alg-Viz quando $f = 2$.

A corretude do algoritmo foi mostrada em [1].

5.3.1 Alg-Cor

Outro algoritmo desenvolvido em ARANTES[1] é chamado Alg-Cor. Neste algoritmo, quando um nó vence o sorteio dos vizinhos probabilísticos remanescentes, ele não simplesmente orienta as suas arestas. O nó escolhe a menor cor ainda não escolhida pelos vizinhos e, conforme os vizinhos vão sendo coloridos, a orientação se dá na direção da maior para a menor cor. Portanto, é gerada uma $(\Delta + 1)$ -coloração que dá a base inicial à orientação. Como já foi citado no Capítulo 4 - Trilhas e Propriedades do ERA, este é o algoritmo que gera as maiores concorrências dentre aqueles discutidos (Calabrese/França, Alg-Viz e Alg-Arestas, descrito posteriormente). Tal superioridade foi determinada de forma experimental e está relacionada à geração de menores caminhos orientados na orientação acíclica inicial. Em

ARANTES[5], foram exploradas algumas estratégias relacionadas à geração de orientações baseadas em $(\Delta + 1)$ -colorações e esquemas de vizinhança.

A velocidade de convergência de Alg-Cor é a mesma de Alg-Viz, já que, além da mesma estratégia de sorteio, a geração de orientações para as arestas de Alg-Viz ocorrida no momento em que o nó ganha de todos os vizinhos probabilísticos remanescentes é da mesma natureza que a escolha de cores de Alg-Cor.

5.3.2 Análise de Complexidade

Similarmente ao algoritmo Calabrese/França, denotaremos $d_i^k \in \{1, 2, \dots, f-1\}$ o valor do dado sorteado pelo nó $v_i \in N_k$ no k -ésimo passo do algoritmo. Aplica-se a mesma definição de S_i^k e S^k realizada na seção anterior.

Inicialmente, temos o seguinte lema para grafos arbitrários:

Lema 5.1. *Considere $G_k = (N_k, E_k)$ um grafo conexo com $|N_k| \geq 2$ e $k \geq 0$. Também considere dados não polarizados com $f \geq 2$ faces e o evento S_i^k um evento que ocorre quando $v_i \in N_k$ é o vencedor no k -ésimo passo. Assumindo $\Delta_k = \max\{|viz_{N_k}(v_i)| : v_i \in N_k\}$, então $\Pr(S_i^k) \geq h(\Delta_k, f)$ onde:*

$$h(\Delta_k, f) = \left(\frac{1}{\Delta_k + 1}\right) \left(1 - \frac{1}{f}\right)^{\Delta_k + 1} + \left(\frac{1}{2f}\right) \left(1 - \frac{1}{f}\right)^{\Delta_k}$$

Demonstração. Considere d_i^k o valor do dado associado ao nó $v_i \in N_k$. Pela definição de probabilidade condicional, temos que:

$$\Pr(S_i^k) = \sum_{\alpha=0}^{f-1} \Pr(d_i^k = \alpha) \cdot \Pr(S_i^k | d_i^k = \alpha), \forall v_i \in N_k \quad (5.1)$$

Associado a cada $v_i \in N_k$, considere $B_j^k(v_i, \alpha)$ um evento que ocorre toda vez que $d_i^k = \alpha > d_j^k$ para algum $v_j \in viz_{N_k}(v_i)$, ou seja, o nó ganha o sorteio contra algum vizinho probabilístico. Note que os eventos $B_j^k(v_i, \alpha)$

são independentes para todo $v_j \in \text{viz}_{N_k}(v_i)$. Além disso, $\Pr(B_j^k(v_i, \alpha)) = \frac{\alpha}{f}$.

Assim, para $v_i \in N_k$ e $\Delta_k = \max\{|\text{viz}_{N_k}(v_i)| : v_i \in N_k\}$, temos que:

$$\Pr(S_i^k | d_i^k = \alpha) = \Pr\left(\bigcap_{v_j \in \text{viz}_{N_k}(v_i)} B_j^k(v_i, \alpha)\right) = \prod_{v_j \in \text{viz}_{N_k}(v_i)} \left(\frac{\alpha}{f}\right) \geq \left(\frac{\alpha}{f}\right)^{\Delta_k} \quad (5.2)$$

Substituindo 5.2 em 5.1 e observando que $\Pr(d_i^k = \alpha) = \frac{1}{f}$, temos:

$$\Pr(S_i^k) \geq \sum_{\alpha=0}^{f-1} \left(\frac{1}{f}\right) \left(\frac{\alpha}{f}\right)^{\Delta_k} = \left(\frac{1}{f^{\Delta_k+1}}\right) \sum_{\alpha=0}^{f-1} \alpha^{\Delta_k}, \forall v_i \in N_k \quad (5.3)$$

Como $|N_k| \geq 2$ e G_k é conexo, temos que $\Delta_k \geq 1$ para todo $k > 0$. Como descrito em SPIEGEL[25], um limite inferior para a série encontrada na expressão anterior é:

$$\sum_{\alpha=0}^{f-1} \alpha^{\Delta_k} \geq \left(\frac{(f-1)^{\Delta_k+1}}{\Delta_k+1} + \frac{(f-1)^{\Delta_k}}{2}\right) \text{ onde } \Delta_k \geq 1 \quad (5.4)$$

Finalmente, substituindo 5.4 em 5.3, temos a expressão esperada. \square

A seguir é confirmada através de uma demonstração a intuição de que a probabilidade de haver um vencedor em um sorteio tende a 1 quando o número de faces de um dado é arbitrariamente grande. A demonstração é para um grafo completo qualquer $G = (N, E)$.

Teorema 5.2. *Se $G = (N, E)$ é um grafo completo, então $\lim_{f \rightarrow +\infty} \Pr(S^k) = 1$ para $k \geq 0$.*

Demonstração. Sendo G um grafo completo, temos que todos os G_k também o serão. Além disso, $\Delta_k = |N_k| - 1$ e todos os eventos S_i^k têm interseção vazia. Chamando $|N_k|$ de n_k , temos que $\Pr(S^k) = \Pr(\bigcup_{i=1}^{n_k} S_i^k) = \sum_{i=1}^{n_k} \Pr(S_i^k) = \sum_{i=1}^{n_k} h(\Delta_k, f)$. Considerando $\lim_{f \rightarrow +\infty} h(\Delta_k, f) = \frac{1}{n_k}$. Portanto, $\Pr(S^k) = \sum_{i=1}^{n_k} \frac{1}{n_k} = 1$. \square

O teorema abaixo mostra que o desempenho de Alg-Viz é sub-exponencial para um valor de $f \geq 2$ constante.

Teorema 5.3. *Considere $G = (N, E)$ um grafo completo com $|N| = n \geq 2$ e um dado não polarizado com f faces ($f \geq 2$). O algoritmo Alg-Viz usando este dado tem complexidade*

$$O\left(f \left(\frac{f}{f-1}\right)^{n-1}\right)$$

Demonstração. Considere $k = 0$ e $N = P_0$ como o conjunto inicial de nós probabilísticos. Sendo $G = (P_0, E)$ completo, segue pelo Lema 5.1 que, para todo $n_i \in P_0$, temos que:

$$\Pr(S^0) \geq \left(\frac{1}{n}\right) \left(1 - \frac{1}{f}\right)^n + \left(\frac{1}{2f}\right) \left(1 - \frac{1}{f}\right)^{n-1} = \left(\frac{2(f-1) + n}{2fn}\right) \left(1 - \frac{1}{f}\right)^{n-1}$$

Observe que todo G_k , $k \geq 0$, é também um grafo completo e que os eventos S_i^k (para todo $n_i \in N_k$) apresentam interseção vazia entre si. Assim, pelo Lema 5.1, um limite inferior para $\Pr(S^k)$ seria

$$\Pr(S^k) = \Pr\left(\bigcup_{n_i \in N_k} S_i^k\right) \geq \Pr\left(\bigcup_{i=0}^n S_i^0\right) = \sum_{i=1}^n \Pr(S_0) = \frac{1}{N}$$

$$\text{onde } N = \left(\frac{2f}{2(f-1) + n}\right) \left(\frac{f}{f-1}\right)^{n-1}$$

Agora, sem perda de generalidade, considere $\Pr(S^k) = \frac{1}{N}$ em todo passo $k \geq 0$ do algoritmo. Desta forma, deverá haver em torno de N tentativas até que um nó seja o vencedor em todo passo $k \geq 0$. Como já foi dito anteriormente, o número de repetições pode ser expresso por uma variável aleatória com distribuição geométrica e com probabilidade de sucesso igual a $\frac{1}{N}$. Como n nós precisam se tornar vencedores durante a execução do algoritmo, é possível concluir que Alg-Viz tem tempo de convergência expresso

da forma abaixo:

$$T(n) = \left(\frac{2fn}{2(f-1) + n} \right) \left(\frac{f}{f-1} \right)^{n-1} = O \left(f \left(\frac{f}{f-1} \right)^{n-1} \right)$$

o que completa a demonstração. \square

Em seguida, é apresentada um teorema que demonstra uma similaridade entre o desempenho de Alg-Viz com dados polarizados e $f = n$ faces e o algoritmo polarizado de Calabrese/França quando são considerados grafos completos, que são a pior instância do problema. Demonstra-se que, neste caso, o desempenho é $O(n.e)$, assim como no caso polarizado de Calabrese/França.

Teorema 5.4. *Se $G = (N, E)$ é um grafo completo e $f = n = |N|$, então $T(n) = O(n.e)$.*

Demonstração. É suficiente verificar que, segundo o Teorema 5.3, para $f = n$, temos que $\left(\frac{n}{n-1}\right)^{n-1} < e$ para todo $n \geq 2$. Logo, $T(n) = O(n.e)$. \square

O teorema acima considera uma execução de Alg-Viz que contém algum conhecimento global sobre o problema: o algoritmo sabe, *a priori*, o número de nós do sistema. Em sistemas anônimos tal conhecimento não é possível.

Na próxima seção, é avaliada uma forma de polarização para este algoritmo. Note que, para o caso de grafos completos, teremos $\Omega(n)$ passos, no mínimo, para a convergência do algoritmo. De qualquer forma, a intenção é garantir que o algoritmo rode apenas com conhecimento local, permitindo sua utilização por sistemas distribuídos anônimos e, ao mesmo tempo, usufruir de um melhor desempenho.

5.3.3 Alg-Viz Polarizado

Como já foi dito, Alg-Viz pode ser considerado uma versão generalizada de Calabrese/França. Portanto, a intenção é generalizar também o conceito

de polarização. Para isso, a expressão da polarização deve ser uma variante natural do caso das moedas utilizadas em Calabrese/França e deve manter algumas características básicas importantes, como por exemplo, ser uma distribuição de probabilidades nas faces do dado, ou seja, $\sum_{i=0}^{f-1} \Pr(d_i^k = i) = 1$.

Assim, para um nó $v_i \in N_k$ e uma constante $f \geq 2$, considere a seguinte polarização das faces do dado num passo qualquer $k \geq 2$ do algoritmo:

$$\Pr(d_i^k = \alpha) = \frac{1}{f} \left(1 - \left(\frac{\alpha}{f-1} \right) \left(\frac{|viz_{N_k}(v_i)|-1}{|viz_{N_k}(v_i)|+1} \right) \right), \alpha > 0$$

$$1 - \sum_{\alpha=1}^{f-1} \Pr(d_i^k = \alpha), \alpha = 0$$

Note que, para $f = 2$, esta polarização é idêntica àquela utilizada em Calabrese/França. Também é fácil perceber que, se $v_i \in N_k$ e $|viz_{N_k}(v_i)| = 1$, para algum $k \geq 0$, então $\Pr(d_i^k = \alpha) = \frac{1}{f}$. Assim, para o caso especial em que $|N_k| = 2$, a probabilidade de empate entre os dois nós é $\frac{1}{f}$, o que significa que a probabilidade de haver um vencedor no k -ésimo passo é igual a $\Pr(S^k) = 1 - \frac{1}{f} \geq \frac{1}{2}$ e, portanto, a probabilidade de haver vencedores aumenta com o valor de f .

Outra indicação de que a polarização é adequada advém da observação de que $\Pr(\alpha = f-1) = \frac{1}{f} \frac{2}{|viz_{N_k}(v_i)|+1} \leq \frac{1}{|N_{N_k}(v_i)|+1}$, para todo $f \geq 2$. Isto significa que a probabilidade de um nó sortear o maior valor do dado é sempre menor do que a de Calabrese/França e se reduz com o valor de f . Desta forma, explora-se a redução na probabilidade de empate para valores maiores de f .

Na verdade, de forma geral, esta polarização equilibra bem os fatores de desempate. Para valores pequenos de f , utiliza-se o número de vizinhos como fator polarizador. Para valores grandes de f , este tende a dominar, tornando a polarização mais uniforme.

Na Figura 5.1 são apresentadas algumas simulações onde são comparados o número de iterações necessárias para a convergência do algoritmo nos casos

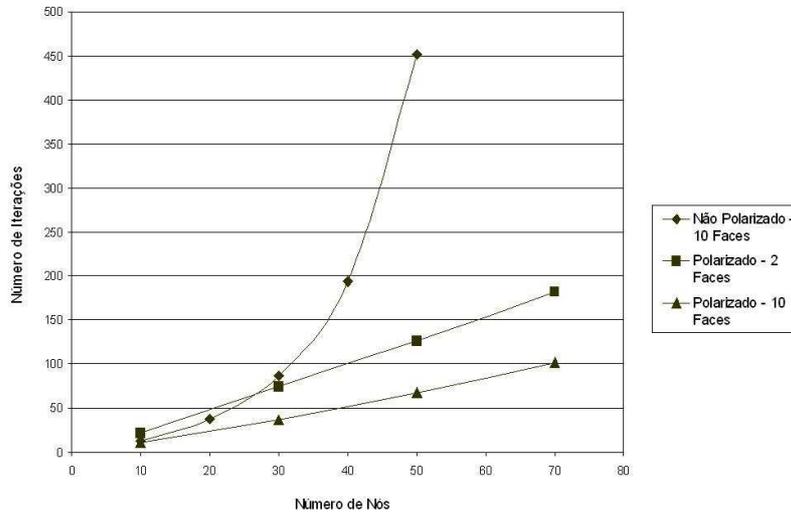


Figura 5.1: Comparação de dados polarizados e não polarizados.

com moedas polarizadas e com dados polarizados e não polarizados de 10 faces.

As simulações foram realizadas com versões sequenciais do algoritmo implementado em linguagem C. Cada ponto do gráfico é o valor médio da simulação do algoritmo para 1000 grafos gerados aleatoriamente.

É fácil notar que a polarização torna a convergência do algoritmo mais rápida. Além disso, consistentemente, o número de faces maior resulta em melhor desempenho, validando a expressão utilizada para a polarização.

5.4 Alg-Arestas

A diferença deste algoritmo para os anteriores é também bastante pequena, em princípio. Novamente, todos os nós realizam um sorteio com um dado de f faces. Porém, o procedimento que orienta as arestas é ainda mais

simples: a aresta entre dois nós é orientada na direção daquele que tiver obtido o maior número no sorteio. Em caso de empate, a aresta permanece não orientada e seus nós adjacentes permanecem probabilísticos, participando da próxima iteração do algoritmo, onde é realizado um novo sorteio. O processo se repete até que todas as arestas tenham sido orientadas. Introduzida em ARANTES[1], uma demonstração de corretude do algoritmo é reproduzida a seguir.

5.4.1 Corretude de Alg-Arestas

Teorema 5.5. *O algoritmo Alg-Arestas produz orientacoes acíclicas no grafo alvo $G(N, E)$.*

Demonstração. Definimos K como sendo o conjunto de todos os ciclos simples (que não apresentam nós repetidos) do grafo $G(N, E)$. A idéia é demonstrar que, para qualquer ciclo simples $\kappa \in K$, o algoritmo *Alg-Arestas* não gera ciclos. κ é composto dos nós $u_0, u_1, \dots, u_{|\kappa|-1}$, considerados em ordem de vizinhança no ciclo, de tal forma que, para todo $0 \leq i \leq |\kappa|$, u_i é conectado por uma aresta de κ a $u_{(i+1) \bmod |\kappa|}$. Os números sorteados no primeiro passo para os nós do ciclo são dados por u_i^s . Podemos ter duas possibilidades:

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

(b) $\exists a$ tal que $u_a^s < u_{(a+1) \bmod |\kappa|}^s$ - Nesse caso, tomemos uma nova ordenação dos nós $v_0, v_1, \dots, v_{|\kappa|-1}$ tal que $v_i = u_{(a+i) \bmod |\kappa|}$ e, por conseqüência, dos números sorteados $v_0^s, v_1^s, \dots, v_{|\kappa|-1}^s$. Já sabemos que $v_0^s < v_1^s$, portanto a aresta (v_0^s, v_1^s) terá a direção $\overrightarrow{v_0^s v_1^s}$. Desta forma, se $\forall j, 1 \leq j < |\kappa| - 2$, temos $v_j^s \leq v_{j+1}^s$, então $v_{|\kappa|-1}^s > v_0^s$ e, portanto, a aresta $(v_{|\kappa|-1}^s, v_0^s)$ terá direção $\overrightarrow{v_{|\kappa|-1}^s v_0^s}$, impossibilitando que seja formado um ciclo orientado em κ nos passos seguintes do algoritmo. Ao contrário, se $\exists j, 1 \leq j < |\kappa| - 2$, tal que,

$v_j^s > v_{j+1}^s$, então a aresta (v_j^s, v_{j+1}^s) terá direção $\overrightarrow{v_j^s v_{j+1}^s}$, impedindo do mesmo modo a formação de um ciclo. \square

5.4.2 Análise de Tempo de Execução

Em relação ao tempo de execução, é fácil perceber a superioridade de Alg-Arestas em relação aos algoritmos anteriores. Particularmente, nota-se que, mesmo em um grafo completo, por exemplo, há a possibilidade de não haver empates logo no primeiro sorteio e, portanto, de o sistema tornar-se orientado em apenas 1 passo. E, certamente, com o número suficientemente grande de faces no dado, é até provável que isto ocorra [1]. Neste contexto, a necessidade de polarização é muito menor, já que o algoritmo em si apresenta eficiência superior.

A convergência de Alg-Arestas foi analisada em ARANTES[1] e, assumindo um dado de f faces, o tempo $T(m)$ de convergência do algoritmo apresenta o seguinte valor esperado:

$$E[T(m)] = \sum_{i=1}^{+\infty} i \left[\left(1 - \frac{1}{f^i}\right)^m - \left(1 - \frac{f}{f^i}\right)^m \right]$$

Foi demonstrado experimentalmente que tal expressão resulta em medidas bastante próximas das obtidas via simulações. Também experimentalmente, pode-se verificar a relação $\log_f m \leq E[T(m)] \leq \log_f m + 2$.

Porém, informações mais detalhadas acerca da distribuição dos resultados podem ser interessantes, já que o valor esperado pode esconder resultados bastante díspares de experimentos individuais.

Análise Através da Expressão de Karp

Um dos instrumentos utilizados para essa análise é um trabalho apresentado em KARP[17] que aborda problemas com características de recorrência probabilística como este.

Para analisar a convergência de Alg-Arestas, observe que a probabilidade de empate entre nós adjacentes é igual a $\frac{1}{f}$. Mais ainda, podemos afirmar que $\frac{1}{f}$ é a razão de empates em todo o grafo, sugerindo que $\frac{1}{f}$ de todas as arestas remanescentes não são orientadas em cada passo (e, claro, $\frac{f-1}{f}$ destas o são). Este processo se repete até que não existam arestas remanescentes.

Desta forma, seguindo a abordagem de KARP[17] podemos descrever o tempo de execução de um algoritmo com a recursão $T(x) = a(x) + T(h(x))$, onde x é uma variável real não negativa representando o tamanho do problema em um determinado passo da recursão, $h(x)$ é uma função randômica assumindo valores no intervalo $[0, x]$ e $a(x)$ é uma função real não negativa de x que modela o esforço necessário para quebrar um problema de tamanho x em um problema de tamanho $h(x)$.

Considere uma função $g(x)$ tal que $E(h(x)) \leq g(x) \leq x$, onde $g(x)$ e $\frac{g(x)}{x}$ são funções reais não negativas e não decrescentes de x . Em KARP[17] são apresentados limites para a distribuição de $T(x)$ (dado que $h(x)$ é uma função randômica, $T(x)$ também o é) quando $h(x)$ obedece certas restrições.

A equação $\tau(x) = a(x) + \tau(g(x))$ pode ser considerada uma versão determinística da relação recursiva $T(x)$. Nesse caso, para cada valor de x , a variável randômica $h(x)$ será igual ao limite superior do seu valor esperado, ou seja, $E(h(x)) = g(x)$, para todo x . Karp descreve a solução $u(x)$ que é dada pela fórmula abaixo:

$$u(x) = \sum_{i=0}^{+\infty} a(g^{[i]}(x)) \quad (5.5)$$

onde $g^{[i]}(x)$ é definido como $g(g^{[i-1]}(x))$ e $g^{[0]}(x) = x$.

Com todas estas definições, em KARP[17] é demonstrado o teorema abaixo:

Teorema 5.6. *Considere d uma constante tal que $a(x) = 0$ para $x < d$ e $a(x) = 1$ para $x \geq d$. Tome $c_t = \min\{x : u(x) \geq t\}$. Então, para todo real positivo x e todo inteiro positivo w , temos que*

$$\Pr[T(x) \geq u(x) + w] \leq \left(\frac{g(x)}{x}\right)^{w-1} \frac{g(x)}{c_{u(x)}} \quad (5.6)$$

O teorema acima define a probabilidade de o valor de $T(x)$ (que, como já foi dito, é uma variável aleatória) ser maior ou igual w unidades do valor de referência $u(x)$. Resumindo, o teorema dá uma medida para a variação de $T(x)$ acima desta referência.

Este teorema, portanto, pode ser utilizado para a avaliação da convergência de Alg-Arestas e é expresso no seguinte teorema.

Teorema 5.7. *Considere um grafo não direcionado qualquer $G = (N, E)$ com m arestas e um dado não polarizado com f faces. Sobre o tempo de convergência $T(m)$ do algoritmo Alg-Arestas podemos afirmar que*

$$\Pr[T(m) \geq \lfloor \log_f m \rfloor + 1 + w] \leq \left(\frac{1}{f}\right)^{w-1} \frac{m}{f^{\lfloor \log_f m \rfloor + 1}} \quad (5.7)$$

Demonstração. Para aplicar o Teorema 5.6, precisamos fazer algumas definições. Primeiramente, consideramos $a(m)$ como a função que mede o número de sorteios necessários para reduzir o número de arestas a serem orientadas para $h(m)$. Logicamente, temos que $d = 1$, já que $a(m) = 1$ para $m \geq 1$ e $a(m) = 0$ para $m = 0$, já que, nesse caso, o problema já foi resolvido.

Deve-se calcular $E(h(m)) = g(m)$, a função que represente o valor esperado para a variável aleatória $h(m)$, sendo esta o número de arestas não

orientadas em um sorteio em que participam m arestas. Considerando cada aresta individualmente, a probabilidade de esta não ser orientada é a probabilidade de empate no sorteio, ou seja, $\frac{1}{f}$. Desta forma, $g(m) = \frac{m}{f}$. Observe que $g(m)$ e $\frac{g(m)}{m}$ são funções não negativas e não decrescentes de m .

A solução $u(m)$ de $\tau(x) = a(x) + \tau(g(x))$ é dada por $u(m) = \sum_{i=0}^{\infty} a(g^{[i]}(m))$. Por definição, temos que $g^{[0]}(m) = m$ e, portanto, $a(g^{[0]}(m)) = 1$. Para $i > 0$, temos que $g^{[i]}(m) = \frac{m}{f^i}$, logo: (a) $a(g^{[i]}(m)) = 0$ se $g^{[i]}(m) = \frac{m}{f^i} < 1$, ou seja, quando $i \geq \lfloor \log_f m \rfloor + 1$ e (b) $a(g^{[i]}(m)) = 1$ caso contrário, ou seja, $0 < i \leq \lfloor \log_f m \rfloor$. Tomando

$$u(m) = a(g^{[0]}(m)) + \dots + a(g^{\lfloor \log_f m \rfloor}(m)) + a(g^{\lfloor \log_f m \rfloor + 1}(m)) + \dots$$

Sabemos que $a(g^{[i]}(m)) = 1$, para $i = 0, \dots, \lfloor \log_f m \rfloor$, e $a(g^{[i]}(m)) = 0$, para $i = \lfloor \log_f m \rfloor + 1, \dots$. Logo

$$u(m) = \lfloor \log_f m \rfloor + 1$$

Para terminar a demonstração, começamos provando que $c_t = f^{t-1}$. Segundo a definição, $c_t = \min\{m \mid u(m) \geq t\}$. Logo:

$$u(m) \geq t \rightarrow \lfloor \log_f m \rfloor + 1 \geq t \rightarrow \log_f m \geq t - 1 \rightarrow f^{\log_f m} \geq f^{t-1} \rightarrow m \geq f^{t-1}$$

Logo, temos $c_t = \min\{m \mid m \geq f^{t-1}\} = f^{t-1}$. E, portanto, $c_{u(m)} = f^{\lfloor \log_f m \rfloor + 1 - 1} = f^{\lfloor \log_f m \rfloor}$. Assim:

$$\begin{aligned} \Pr[T(m) \geq u(m) + w] &\leq \left(\frac{g(m)}{m}\right)^{w-1} \frac{g(m)}{c_{u(m)}}, \\ \Pr[T(m) \geq \lfloor \log_f m \rfloor + 1 + w] &\leq \left(\frac{\frac{m}{f}}{m}\right)^{w-1} \frac{\frac{m}{f}}{f^{\lfloor \log_f m \rfloor}}, \\ \Pr[T(m) \geq \lfloor \log_f m \rfloor + 1 + w] &\leq \left(\frac{1}{f}\right)^{w-1} \frac{m}{f^{\lfloor \log_f m \rfloor + 1}} \end{aligned}$$

como queríamos demonstrar. \square

Interpretação dos Resultados

Uma idéia clara do funcionamento do Alg-Arestas surge pela avaliação da Expressão 5.7. Por exemplo, é possível perceber que, para $m = f^k$, onde $k \in \mathbb{N}$, temos que $\Pr[T(m) \geq u(m) + 1] \leq \frac{1}{f}$. Isso significa que a probabilidade de a convergência do algoritmo se afastar apenas 1 unidade do centro do valor esperado é, na pior das hipóteses, inversamente proporcional ao número de faces do dados. Para dados com número grande de faces, isso significa que a grande maioria dos experimentos irá convergir em até $\lfloor \log_f m \rfloor + 2$ passos. Além disso, quando $m = f^k$, temos que $\Pr[T(m) \geq u(m) + w] \leq \left(\frac{1}{f}\right)^w$, o que deixa ainda mais claro que a probabilidade de um experimento ultrapassar uma unidade a mais o valor de $u(m)$ decresce exponencialmente.

Os gráficos 5.2 e 5.3 apresentam os resultados da aplicação da Expressão 5.7. Em ambos, o número de arestas dos grafos utilizados é dado pelo eixo x . Em 5.2, é fixado o valor $w = 2$ e são apresentadas seqüências com número de faces 2, 5 e 10. Assim, procura-se mostrar a influência de m e de f na probabilidade calculada. Em 5.3, a intenção é entender principalmente a influência de w , já que são mostradas seqüências para vários valores de w e mantendo-se $f = 2$.

Nota-se que o comportamento da expressão é bastante incomum. Porém, é fácil perceber o motivo deste comportamento pela presença de $\lfloor \log_f m \rfloor$ na expressão plotada no gráfico. Como já foi dito, para $m = f^k$, temos $\Pr[T(m) \geq u(m) + w] \leq \left(\frac{1}{f}\right)^w$. Isso significa que o resultado da expressão é idêntico para todo valor de m que seja igual a uma exponencial inteira positiva de f . Porém para valores de m no intervalo $f^i < m < f^{i+1}$ (i é algum inteiro positivo), a única parte da expressão que tem seu valor alterado com o crescimento de m (eixo x nos gráficos) é o valor de m no numerador da segunda fração da Expressão 5.7 (reproduzida aqui para facilitar:

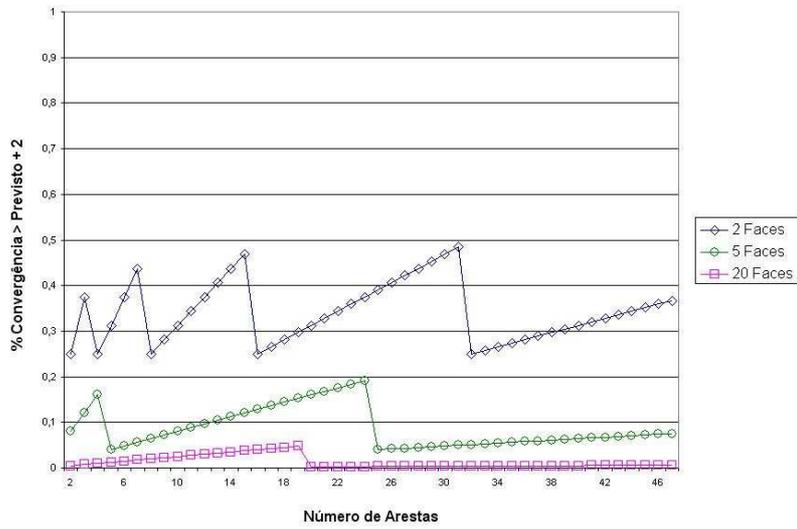


Figura 5.2: Análise da Expressão Karp com $w = 2$.

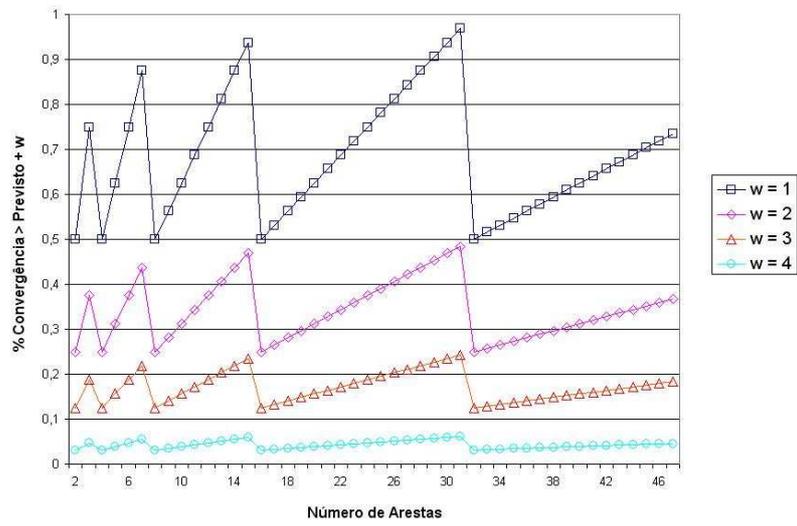


Figura 5.3: Análise da Expressão Karp com w variável.

$\left(\frac{1}{f}\right)^{w-1} \frac{m}{f^{\lfloor \log_f m \rfloor + 1}}$). Portanto, temos uma reta crescente entre os valores de m correspondentes a exponenciais inteiras de f .

Uma questão que pode ser colocada é se o aumento de valor da Expressão 5.7 nos intervalos entre as exponenciais positivas de f representa uma redução da capacidade de previsão da fórmula ou reflete a realidade do sistema. O gráfico da Figura 5.4 compara os resultados dessa expressão usando $w = 2$ com o resultado de simulações. No eixo x são utilizados vários valores de m e no eixo y são plotados os valores da probabilidade $P[T(m) \geq u(m) + 2]$, ou seja, usando $w = 2$.

As seqüências “Karp (f=2)” e “Karp (f=5)” apresentam o cálculo de $P[T(m) \geq u(m) + 2]$ segundo a Expressão 5.7 para valores de f iguais a 2 e a 5, respectivamente.

Nas seqüências “Simulado (f=2)” e “Simulado (f=5)” são calculados, através de simulação, os valores reais de $P[T(m) \geq u(m) + 2]$. Tal cálculo se dá da seguinte forma: para cada ponto do gráfico, temos definidos os valores de w (que é fixo em 2), de f (que é 2 ou 5 de acordo com a seqüência) e de m (definido pelo eixo x). Para cada tupla (w, f, m) , portanto, são executadas 10000 simulações do Alg-Arestas e, em cada simulação, é comparado o tempo de convergência do algoritmo com o valor de $u(m) + 2$ (ou seja, $\lfloor \log_f m \rfloor + 1 + 2$). São contados, para cada tupla, o número b de vezes em que o tempo real de convergência foi igual ou maior a $u(m) + 2$. Assim, a expressão $\frac{b}{1000}$ representa o valor simulado de $P[T(m) \geq u(m) + 2]$.

É interessante notar que o comportamento da expressão equivale ao comportamento real do sistema. Ou seja, o valor real de $P[T(m) \geq u(m) + w]$ realmente aumenta quando m está em um intervalo entre exponenciais inteiras de f , ou seja $f^i < m < f^{i+1}$ (no restante dessa seção, i representa qualquer inteiro positivo), e apresenta valor fixo quando $m = f^i$. Isso significa que

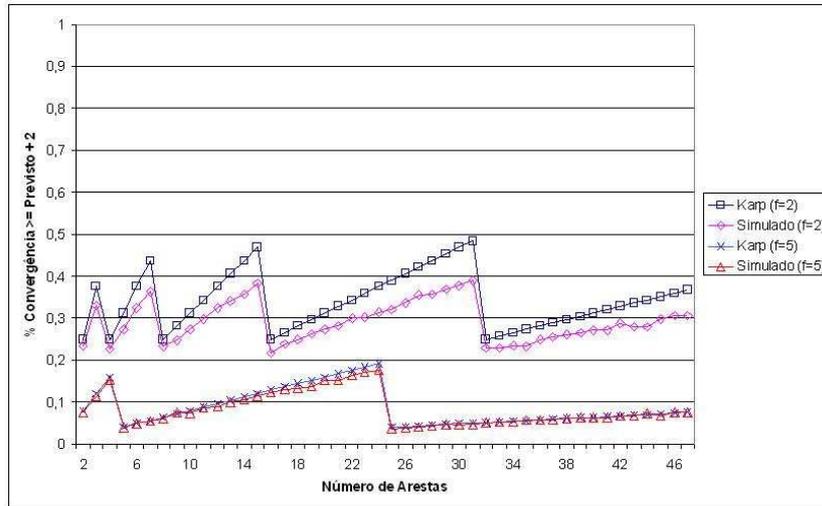


Figura 5.4: Comparando resultados de simulação com a Expressão Karp.

não é a expressão que perde precisão nesse intervalo, mas o sistema em si que fica menos previsível. Dessa forma, o aumento de número de arestas causa uma espécie de perturbação na previsibilidade do sistema até que esta atinja um valor igual a f^i , quando há uma acomodação. Essa acomodação se dá porque a Expressão 5.7 tem variáveis dos dois lados da desigualdade, já que $u(m)$ também altera seu valor com o crescimento de m . Em termos intuitivos acerca da dinâmica do sistema, também não é estranho que este aumento de imprecisão ocorra, já que, quando $f^i \leq m < f^{i+1}$, medimos a probabilidade de $T(m)$ ser maior que uma constante $c = \lceil \log_f f^i + \delta \rceil + 1 + w = i + 1 + w$, $\delta < f^{i+1} - f^i$, ou seja, a expressão torna-se $P[T(m) \geq c]$. E é claro que esta probabilidade tende a aumentar com o valor de m já que a variável aleatória $T(m)$ obviamente é influenciada por este parâmetro.

Um ponto que pode ser examinado é quanto imprecisa pode se tornar a expressão para valores de m nos intervalos entre exponenciais positivas de f . Tal verificação é motivada uma característica da expressão, a saber:

pode-se ver no gráfico 5.4 que as probabilidades (simulada ou calculada) assumem valores máximos, ou picos, exatamente quando $m = f^i - 1$. Um detalhe é que estes picos aumentam juntamente com o valor de i . Assim, na seqüência “Karp (f=2)”, temos os seguinte pontos de pico no gráfico: $(2^3 - 1; 0, 437)$, $(2^4 - 1; 0, 468)$ e $(2^5 - 1; 0, 484)$. Na seqüência “Karp (f=5)” (não faz muita diferença utilizar uma seqüência simulada ou calculada, já que estas apresentam o mesmo comportamento), temos os seguintes picos (alguns não aparecem no gráfico, mas foram simulados): $(5^1 - 1; 0, 152)$, $(5^2 - 1; 0, 175)$ e $(5^3 - 1; 0, 182)$. Estes dados comprovam, numericamente, o que já era verificável visualmente: os picos aumentam com o valor de i . Portanto, gostaríamos de descartar a hipótese de ocorrerem grandes imprecisões para valores grandes de i . Temos a seguinte expressão para estes casos (substituindo m por $f^i - 1$):

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

Pode-se verificar facilmente que

$$\lim_{i \rightarrow +\infty} \left(\frac{1}{f}\right)^{w-1} \frac{f^i - 1}{f^{\lfloor \log_f (f^i - 1) \rfloor + 1}} = \lim_{i \rightarrow \infty} \left(\frac{1}{f}\right)^{w-1} \frac{f^i - 1}{f^{(i-1)+1}} = \left(\frac{1}{f}\right)^{w-1}$$

Assim, nota-se que, para casos em que desejamos considerar $w = 1$, há bastante imprecisão para grandes valores de m tais que $m = f^i - 1$. O gráfico 5.3 confirma esta suspeita, bastando notar a seqüência em que $w = 1$ (em todas as seqüências deste gráfico, foi usado $f = 2$). Porém, para outros valores de w a imprecisão converge para um valor limitado e menor $\frac{1}{2}$.

O que podemos verificar, pelas análises e gráficos é que, para valores de f não muito pequenos (por exemplo $f > 3$) e para $w > 1$, a Expressão 5.7 é bastante precisa na previsão do comportamento do sistema. Além disso, vimos que existe uma redução da previsibilidade (e não da precisão)

que é devida às próprias características do sistema para valores de m como $m = f^{i-1}$. A imprevisibilidade cresce juntamente com o valor de i , porém, não se torna muito significativa nos casos em questão ($w > 1$ e $f > 3$).

Porém, ainda não está claro se, para $w = 1$, o sistema apresenta realmente o nível de imprevisibilidade demonstrado na fórmula, que é muito grande, já que a probabilidade para $m = f^i - 1$ aproxima-se rapidamente de 1. Adiante, em comparação com simulações, verificaremos que, com $w = 1$, não é a previsibilidade do sistema que reduz e, sim, a precisão da Expressão 5.7.

Por estes e outros motivos, introduzimos outra análise.

5.4.3 Análise Através de Outra Técnica

A análise do tempo de convergência através do uso do teorema de Karp apresenta alguns limitantes. Primeiramente, os valores de w devem ser inteiros positivos. Logo, se temos $\lfloor \log_f m \rfloor + 1 = 4$, não é possível saber, por exemplo, qual é a probabilidade de um experimento convergir em 1, 2, 3 ou 4 passos. Além disso, para valores pequenos de f e para $w = 1$, a precisão da expressão de Karp não é tão boa, como poderemos verificar.

Abaixo, é apresentada uma expressão que tenta expressar, através de uma análise probabilística baseada nas características específicas do problema, a probabilidade de um experimento individual convergir em exatamente i passos (i inteiro positivo).

Teorema 5.8. *Considere um grafo não direcionado qualquer $G = (N, E)$ com m arestas e um dado não polarizado com f faces. Sobre o tempo de convergência $T(m)$ do algoritmo Alg-Arestas dentro destes parâmetros, podemos afirmar que*

$$P[T(m) = i] = \left(1 - \frac{1}{f^i}\right)^m - \left(1 - \frac{f}{f^i}\right)^m \quad (5.8)$$

Demonstração. Cada aresta no algoritmo Alg-Arestas é orientada de forma independente e a probabilidade de empate (ou fracasso) em cada tentativa é de $q = \frac{1}{f}$, pois temos f^2 configurações possíveis para o sorteio dos dois dados e em f possibilidades há empates. Portanto a probabilidade de sucesso é de $p = 1 - \frac{1}{f}$. Assim, podemos modelar o número de passos que uma aresta $m_k \in E$ leva até ser orientada como uma variável aleatória geométrica X_k . Pela definição de variável aleatória geométrica, temos que a probabilidade de uma aresta ser orientada em exatamente i passos (i inteiro positivo) é dado por

$$P(X_k = i) = pq^{i-1} = \left(1 - \frac{1}{f}\right) \left(\frac{1}{f}\right)^{i-1} = \frac{f-1}{f^i} \quad (5.9)$$

Ou seja, a probabilidade de uma aresta se orientar em exatamente i passos é igual à de fracassar na tentativa $i - 1$ vezes e obter sucesso na i -ésima tentativa. A probabilidade de todo o sistema convergir em i passos é igual à probabilidade de as últimas arestas a serem orientadas demorarem exatamente i passos e pode ser representada por $P[\text{Max}\{X_0, X_1, \dots, X_{m-1}\} = i]$. Ou seja, mede-se a probabilidade de o máximo entre o número de passos necessários para orientar cada aresta ser igual a i .

Considerando, para fins de notação, que $P[X_k = i] = x_k^*$ e $P[X_k < i] = x_k$, temos que $P[\text{Max}\{X_0, X_1, \dots, X_{m-1}\} = i]$ pode ser calculado por

$$x_0^*x_1x_{m-1} + x_0x_1^*x_{m-1} + \dots + x_0^*x_1^*x_2 \dots x_{m-1} + x_0^*x_1x_2^* \dots x_{m-1} + \dots + x_0^*x_1^*x_2^* \dots x_{m-1}^*$$

onde são consideradas as $\binom{m}{1}$ configurações em que apenas uma aresta é a última a ser orientada, somadas às $\binom{m}{2}$ configurações em que duas arestas são as últimas a serem orientadas e, assim, sucessivamente, até a única configuração (já que $\binom{m}{m} = 1$) em que todas as arestas terminam ao mesmo tempo em i passos. Assim, temos que

$$P[\text{Max}\{X_0, X_1, \dots, X_{m-1}\} = i] = \sum_{j=1}^m \left[\binom{m}{j} \cdot P(X = i)^j \cdot P(X < i)^{m-j} \right]$$

Note que nos desfizemos dos índices das variáveis aleatórias, já que todas são geométricas com os mesmos parâmetros. Pode-se verificar que esta expressão é igual a um binômio de Newton sem a parcela $j = 0$. Logo, temos

$$P[\text{Max}\{X_0, X_1, \dots, X_{m-1}\} = i] = [P(X = i) + P(X < i)]^m - P(X = i)^m$$

Sabemos também que

$$P(X < i) = 1 - P(X \geq i) = 1 - \sum_{j=i}^{\infty} P(X = j) = 1 - \sum_{j=i}^{\infty} \frac{f-1}{f^j}$$

Usando a fórmula da PG e substituindo, temos

$$P(X < i) = \frac{f^i - f}{f^i}$$

Substituindo 5.9 e 5.4.3 em 5.4.3, temos

$$P[\text{Max}\{X_0, X_1, \dots, X_{m-1}\} = i] = \left(\frac{f-1}{f^i} - \frac{f^i - f}{f^i} \right)^m - \left(\frac{f^i - f}{f^i} \right)^m$$

Após as simplificações, temos o resultado esperado. \square

Esta demonstração é fortemente baseada em ARANTES[1] e é, na verdade, uma reatuação de algumas idéias a esse contexto. Importante notar que se trata de uma igualdade. Ou seja, a expressão nos informa exatamente a probabilidade de um determinado experimento individual convergir em um número qualquer de passos.

Claramente, é possível calcular o valor de $P[T(m) \geq u(m) + w]$, bastando para isso calcular $\sum_{i=w}^{\infty} P[T(m) = u(m) + i]$.

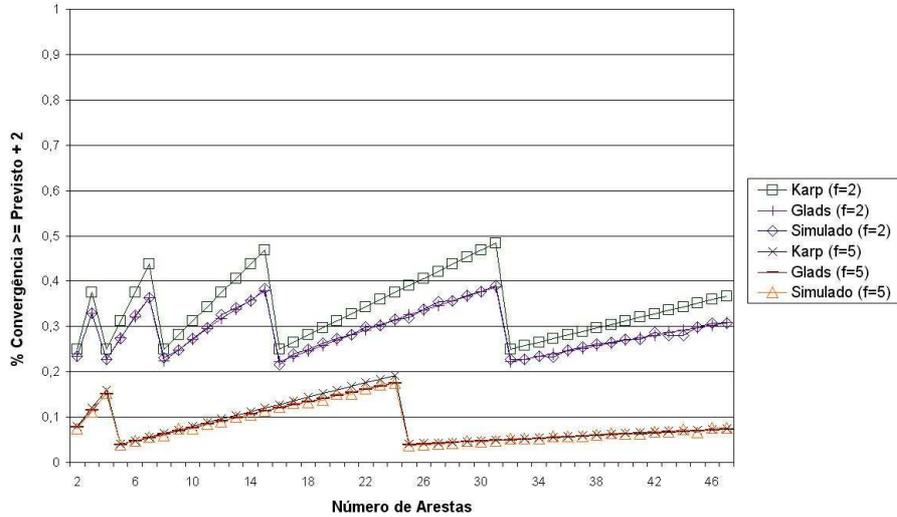


Figura 5.5: Comparando resultados de simulação com as Expressões Karp e Glads com $w = 2$.

5.4.4 Comparação dos Resultados

O gráfico 5.5 mostra uma comparação entre as Expressões 5.7 e 5.8 e uma simulação. No eixo x são utilizados vários valores de m e no eixo y são plotados os valores de $P[T(m) \geq u(m) + 2]$.

As seqüências “Karp ($f=2$)” e “Karp ($f=5$)” apresentam o cálculo de $P[T(m) \geq u(m) + 2]$ segundo a Expressão 5.7 para valores de f iguais a 2 e a 5, respectivamente. Da mesma forma, são as seqüências “Glads ($f=2$)” e “Glads ($f=5$)” com valores gerados pela Expressão 5.8. As seqüências “Simulação ($f=2$)” e “Simulação ($f=5$)” calculam os valores reais de $P[T(m) \geq u(m) + 2]$ com o método já discutido anteriormente. É importante destacar que, para as seqüências “Glads ($f=2$)” e “Glads ($f=5$)”, o valor de $P[T(m) \geq u(m) + 2]$ foi calculado pelo somatório $P[T(m) \geq u(m) + 2] = \sum_{i=u(m)+w+1}^{+\infty} P[T(m) = i]$

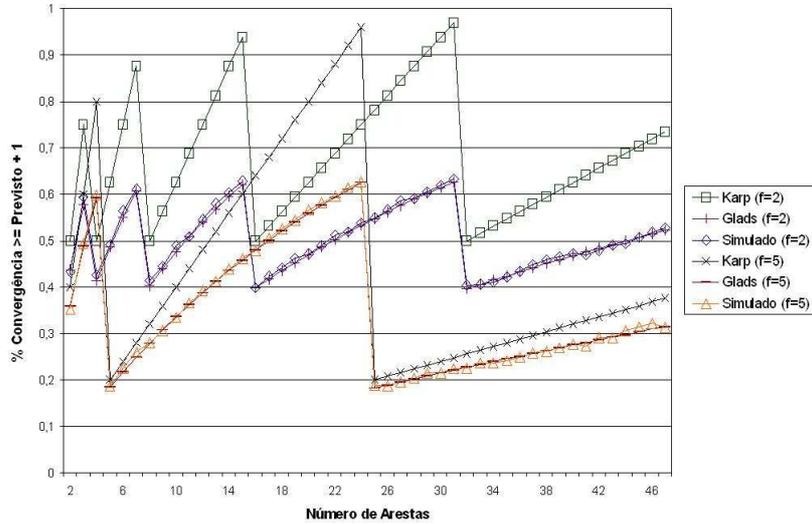


Figura 5.6: Comparando resultados de simulação com as Expressões Karp e Glads com $w = 1$.

(obviamente, foram somadas apenas as 20 primeiras parcelas da expressão - a expressão converge rapidamente e seria possível utilizar até menos parcelas).

Pode-se notar que a Expressão 5.7 é bastante fiel à realidade, principalmente quando $f = 5$. De uma maneira geral, há uma precisão menor na Expressão 5.7 para valores pequenos de f . Na comparação, porém, nota-se que quase não é possível diferenciar a Expressão 5.8 da simulação, confirmando a precisão da igualdade que a expressão representa, mesmo nesse caso em que nem está sendo mostrada a igualdade e, sim, uma soma infinita da expressão básica.

Também há muita perda de precisão da Expressão 5.7 quando $w = 1$. Esta perda pode ser verificada no gráfico 5.6. Novamente, mesmo nesse caso, a Expressão 5.8 é extremamente fiel à simulação.

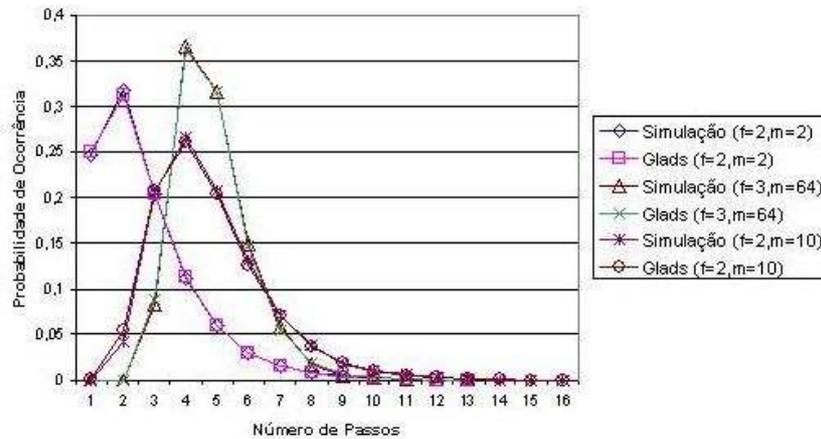


Figura 5.7: Comparando resultados de simulação com a Expressão Glads para vários valores de convergência.

Na Figura 5.7, vemos simulações e resultados da Expressão 5.8. Cada seqüência considera um único par de valores de f e m e mostra a probabilidade (no eixo y) de, dados estes valores, um experimento ocorrer em um certo número de passos (no eixo x). Mais uma vez os resultados são totalmente coerentes para os valores simulados e calculados.

Por tudo isso, entendemos que a utilização da Expressão 5.8 é interessante em vários aspectos, pela sua precisão em quaisquer combinações de f e m e pela capacidade de prever $P[T(m) = i]$ para qualquer inteiro positivo i . Em contrapartida, há também vantagens significativas na utilização da Expressão 5.7, principalmente para casos onde f e w têm valores maiores do que 3 e 1 e quando há necessidade de manipulação algébrica de uma desigualdade do tipo $P[T(m) \geq x]$, já que, nesses casos, a Expressão 5.8 precisa de uma soma infinita para chegar ao resultado. A experiência mostra que a expressão converge rápido, mas não há estudos conclusivos sobre esse tema.

Capítulo 6

Conclusão

Neste trabalho, vários aspectos do Escalonamento por Reversão de Arestas foram pesquisados.

Inicialmente, o foco voltou-se para o estabelecimento da complexidade dos algoritmos de otimização de concorrência.

Foi investigado se o problema de encontrar a concorrência máxima era menos complexo para uma classe bastante simples de grafos. A prova da NP-completude para o problema de decisão associado ao problema de encontrar a concorrência máxima para grafos com grau máximo 4 coloca um parâmetro importante para o problema. A dificuldade da maximização da concorrência torna-se ainda mais clara com o resultado da não existência, a não ser que $P = NP$, de um algoritmo de aproximação com razão menor que $n^{\frac{1}{7}-\epsilon}$, $\epsilon > 0$, para o problema de minimizar o inverso da concorrência.

É sabido que o problema de COLORAÇÃO é polinomial para grafos com grau máximo 3 e NP-completo para grau máximo 4. Para a concorrência, sabe-se agora que o problema é NP-completo para grau máximo 4. Porém, não há resultado para o caso em que o grau máximo é 3.

Outra característica interessante é que a minimização da concorrência também é um problema NP-completo. O problema de maximização de cores tem resultado óbvio igual a $|N|$. Neste ponto, parece que a concorrência tem características comuns ao de ciclo hamiltoniano.

Também foram abordadas algumas características dinâmicas do ERA, onde o fenômeno da modulação do sistema pelos ciclos principais foi investigado e algumas propriedades do ERA foram estabelecidas em função destes ciclos. Novos conceitos foram utilizados para este estudo e novas propriedades foram descobertas: o máximo e o mínimo intervalo inoperante e o número de operações independentes. Também foi mostrado que se um nó do ciclo opera 1 vez a cada d passos, o sistema entra no período se comportamento exatamente da mesma forma.

Essas propriedades reforçam a idéia de que os ciclos principais modulam o comportamento de todo o sistema e, portanto, podem ser utilizados em aplicações futuras para controlar o sistema como um todo. Além disso, alguns estudos futuros podem utilizar esse conhecimento sobre os ciclos principais para tentar derivar outras propriedades do ERA, com algumas que já o foram aqui.

Outro ponto também abordado foi a velocidade de convergência dos algoritmos probabilísticos e distribuídos para inicialização do ERA, que geram orientações acíclicas no sistema. Foram mostradas análises para os algoritmos Alg-Viz e Alg-Cor, este último sendo o algoritmo que gera maiores concorrências entre os algoritmos distribuídos e randômicos estudados. Basicamente, ficou demonstrado que o problema é sub-exponencial para grafos completos e linear quando o número de faces do dado é igual ao número de nós.

Para o algoritmo Alg-Arestas, foi possível estender o conhecimento da velocidade de convergência do algoritmo além da média anteriormente introduzida em ARANTES[1] e ARANTES[2]. Foi utilizado um resultado bem estabelecido em KARP[17] para calcular a probabilidade de a convergência de um experimento individual se afastar um certo número de passos acima do valor esperado, representado por $P[T(m) > U(m) + w]$, $w > 0$. Foi, também, proposta uma nova forma de análise que permite conhecer a probabilidade de um determinado experimento convergir em exatamente $i > 0$ passos, representado por $P[T(m) = i]$, obtendo esta análise valores mais precisos em determinados casos.

6.1 Trabalhos Futuros

Algumas das perguntas que motivaram parte do trabalho, principalmente o Capítulo 4 - Trilhas e Propriedades do ERA ainda permanecem sem resposta. Com o conceito de trilha esperava-se ser possível responder questões importantes e abertas como a determinação do momento em que o ERA entra no período e se a composição por sumidouros de uma concorrência ótima é uma coloração ótima.

Em relação à entrada no período do ERA, há algumas indicações que não puderam ser totalmente desenvolvidas. Por exemplo, é claro que o sistema só está no período quando os ciclos principais estão em execução canônica. Portanto, se forem estabelecidas as condições a partir das quais as trilhas que percorrem os ciclos da execução canônica não tenham sua execução perturbada, mesmo que o sistema não esteja no período, este poderá apresentar características similares ao período, com várias propriedades similares. Uma conjectura nesse caminho é a de que, se todas as trilhas do sistema já forem

originárias do ciclo principal, então este obrigatoriamente estará no período, o que parece bastante provável, mas não foi mostrado.

Um conceito não introduzido neste trabalho, mas que foi utilizado em conjecturas e podem ser parte de trabalhos futuros são as *trilhas com espera*. As trilhas com espera são seqüências de nós vizinhos em que a trilha só passa para o próximo nó quando o nó corrente opera. Por exemplo, considere os nós, n_1, n_2, n_3 e n_4 , onde n_i é vizinho de n_{i+1} . Considere a seguinte seqüência de eventos iniciados em t_0 : n_1 opera em t_0 ; n_2 opera em $t_0 + 1$ (portanto, a aresta (n_2, n_3) é revertida na direção $\overrightarrow{n_2n_3}$); n_3 não opera em $t_0 + 2$ e opera em $t_0 + 3$; e n_4 não opera em $t_0 + 4$ nem em $t_0 + 5$ e opera em $t_0 + 6$. Temos uma trilha com esperas que sai de n_1 e vai a n_4 com 1 espera em n_3 e 2 esperas em n_4 . Trilhas com espera são generalizações das trilhas e apresentam propriedades também generalizadas.

Espera-se utilizar estas propriedades, por exemplo, no caso em que o ciclo principal tem 2 trilhas (normais) o percorrendo e distando d e D entre si, com $d < D - 1$ (ver Figura 6.1). E analisar, utilizando trilhas com espera, se atrasarmos a trilha τ_2 (fazendo com que o nó não opere em um determinado passo) não afetaremos o caminho da trilha τ_1 , voltando o sistema para o período. Observe que a concorrência não teria sido afetada neste caso. Com esta prova, poderíamos fazer algumas observações importantes acerca da relação entre concorrência e coloração associada, já que, neste tipo de sistema, podemos afirmar que $\chi(G) = D$, já que esta é definida pelo ciclo principal e, através do atraso de τ_2 , estaríamos reduzindo $\chi(G)$ sem alterar $\gamma(G)$.

Outra conjectura importante que está aberta é a relativa aos grafos de Mycielski, onde afirma-se que para um grafo $M(k, n)$, $n > k \geq 2$, tem-se que $\gamma^*(M(2, n)) = \frac{1}{n+1}$. Este resultado é muito importante para o desen-

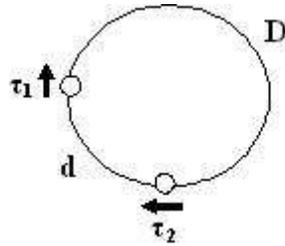


Figura 6.1: A conjectura é que, atrasando τ_2 , τ_1 não é afetada e o sistema converge para o período com o ciclo mantendo a nova execução canônica.

volvimento de heurísticas para maximização da concorrência, já que é uma classe de grafos de estrutura complexa para a qual haveria a informação da concorrência ótima com cujos resultados da heurística poderiam ser comparados. Hoje é difícil validar heurísticas porque a obtenção da concorrência ótima para comparação é muito cara ou a estrutura do grafo é muito pobre.

Em relação à convergência dos algoritmos de inicialização probabilística, existe muito espaço para novos estudos. A expressão 5.8 foi utilizada aqui para modelar a convergência de um algoritmo específico. Mas, a princípio, pode ser utilizada para modelar quaisquer sistemas que apresentem a relação de recorrência $T(x) = a(x) + T(h(x))$ discutida na seção 5.4.2 - Análise de Tempo de Execução, desde que $g(x) = E(h(x)) = \frac{1}{c}$, ou seja, o sistema decresça de tamanho de acordo com um fator constante probabilístico e também que sejam respeitadas as condições impostas por KARP[17]. Desta forma, mesmo um algoritmo que não apresente as mesmas características do Alg-Arestas, mas que possa ser modelado desta forma, pode utilizar a mesma expressão 5.8 para calcular a probabilidade de convergência em um número de passos qualquer.

Uma investigação bastante relevante é se é possível a aplicação da mesma técnica utilizada nesta tese para as outras situações descritas em KARP[17], onde é possível utilizar funções $g(x)$ mais complexas, desde que sejam respeitados alguns limites. Se for possível introduzir a técnica utilizada aqui, teríamos um resultado bastante importante.

Referências Bibliográficas

- [1] ARANTES JR, G. M. *Orientações Acíclicas em Sistemas Distribuídos Anônimos e suas Aplicações no Compartilhamento de Recursos*. Master's thesis, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 1999.
- [2] ARANTES JR, G. M., FRANÇA, F. M. G. “Geração quase-instantânea de orientações acíclicas em sistemas distribuídos anônimos”. In *Anais do II Workshop em Sistemas Computacionais de Alto Desempenho*, pp. 55–62, Pirenópolis, GO, Brasil, 2001.
- [3] ARANTES JR, G. M., FRANÇA, F. M. G., MARTINHON, C. A. J. “Algoritmos randômicos para a geração de orientações acíclicas em sistemas distribuídos”. In *Anais do XXXIV Simpósio Brasileiro de Pesquisa Operacional*, Rio de Janeiro, RJ, Brasil, 2002.
- [4] ARANTES JR, G. M., FRANÇA, F. M. G., MARTINHON, C. A. J. “Gerando orientações acíclicas com algoritmos probabilísticos distribuídos”. *Pesquisa Operacional*, v. 25, n. 3, pp. 301–312, Setembro a Dezembro 2005.
- [5] ARANTES JR, G. M., FRANÇA, F. M. G., MARTINHON, C. A. J., FARIA, L. “A multistart approach to near-optimal concurrency dynamics in neighborhood-constrained systems”. In *Proc. of the Inter-*

- national Conference on Optimization and Control with Applications*, Tunxi, China, 2002.
- [6] BANDYOPADHYAY, S., COYLE, E. J. “An energy efficient hierarchical clustering algorithm for wireless sensor networks”. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, v. 3, pp. 1713–1723, 2003.
- [7] BARBOSA, VALMIR CARNEIRO. *Concurrency in Systems With Neighborhood Constraints*. PhD thesis, University of California Los Angeles, Los Angeles, California, USA, 1986.
- [8] BARBOSA, VALMIR CARNEIRO. “Concurrency in heavily loaded neighborhood-constrained systems”. *ACM Transactions on Programming Languages and Systems*, v. 11, n. 4, pp. pp. 562–584, 1989.
- [9] BARRIÈRE, L., FLOCCHINI, P., FRAIGNIAUD, P., SANTORO, N. “Capture of an intruder by mobile agents”. In *Proceedings of the fourteenth annual ACM symposium on Parallel algorithms and architectures*, pp. 200–209, Winnipeg, Manitoba, Canada, 2002.
- [10] BELLARE, M., GOLDRICH, O., SUDAN, M. “Free bits, PCPs and non-approximability - towards tight results”. *SIAM J. Comp*, v. 27, pp. 804–915, 1998.
- [11] BIN, G., ZHE, L., ZE-JUN, W. “A dynamic-cluster energy-aware routing algorithm based on neural structure in the wireless sensor networks”. In *Proceedings of the The Fifth International Conference on Computer and Information Technology*, pp. 401–405, Washington, DC, USA, 2005. IEEE Computer Society.

- [12] BROOKS, R. L. “On colouring nodes of a network”. In *Proceedings of Cambridge Philosophy Society*, v. 37, pp. 194–197, 1941.
- [13] CALABRESE, A., FRANÇA, F. M. G. “Randomised distributed primer for the updating control of anonymous ann”. In *Proceedings of ICANNR94*, Sorrento, Italy, 1994.
- [14] FRANÇA, F. M. G., ALVES, V. C., GRANJA, E. P. “Edge reversal-based asynchronous timing synthesis”. In *Proc. of IEEE International Symposium on Circuits and Systems*, pp. 45–48, Monterey (CA), 1998.
- [15] FRANÇA, F. M. G., PRADO, C. B., DINIZ, P.S.R. “Implementation of overlapped block filtering using scheduling by edge reversal”. In *Proceedings of the IEEE 2000 International Symposium on Circuits and Systems*, v. 5, pp. 97–100, Piscataway, NJ, USA, 2000.
- [16] GAREY, M.R., JOHNSON, D.S., TARJAN, R.E. “The planar hamiltonian circuit problem is np-complete”. *SIAM Journal on Computing*, v. 5, pp. 704–414, 1976.
- [17] KARP, R. “Probabilistic recurrence relations”. *Journal for Association for Computing Machinery*, v. 41, n. 6, pp. 1100–1136, 1994.
- [18] LOVÁSZ, L. “Three short proofs in graph theory”. *Journal of Combinatorial Theory B*, v. 19, pp. 111–113, 1975.
- [19] MALKA, Y., MORAN, S., ZAKS, S. “A lower bound on the period length of a distributed scheduler”. *Algorithmica*, v. 10, pp. 383–398, 1993.
- [20] MALKA, Y., RASJBAUM, S. “Analysis of distributed algorithms based on recurrence relations”. In *Proc. 5th International Workshop on Dis-*

- tributed Algorithms (WDAG)*, pp. 242–253, Delphi, Greece, October 1991.
- [21] MYCIELSKI, J. “Sur le coloriage des graphes”. In *Colloquim Mathematiques*, v. 3, pp. 161–162, 1955.
- [22] PAPADIMITRIOU, C., YANNAKAKIS, M. “Optimisation, approximation, and complexity classes”. *Journal of Computer and System Sciences*, v. 43, pp. 425–440, 1991.
- [23] SCHAEFER, T. J. “The complexity of satisfiability problems”. In *Proc. 10th Ann. ACM Symp. on Theory of Computing*, pp. 216–226, New York, 1978. Association for Computing Machinery.
- [24] SCHURGERS, C., TSIATSI, V., GANERIWAL, S., SRIVASTAVA, M. “Optimizing sensor networks in the energy-latency-density design space”. *IEEE Transactions on Mobile Computing*, v. 1, pp. 70–80, 2002.
- [25] SPIEGEL, M. R. *Manual de fórmulas, métodos e tabelas de matemática*. Makron, MacGraw-Hill, 1992.