

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA APLICADA

Parâmetros de Dominância em Grafos Regulares

por

Giovane Mansan

Trabalho submetido como requisito parcial
para a obtenção do título de
Doutor em Matemática Aplicada

Prof. Dr. Carlos Hoppen
Orientador

Porto Alegre, junho de 2019.

CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Mansan, Giovane

Parâmetros de Dominância em Grafos Regulares / Giovane Mansan.—Porto Alegre: PPGMAp da UFRGS, 2019.

151 p.: il.

Tese (Doutorado) —Universidade Federal do Rio Grande do Sul, Programa de Pós-Graduação em Matemática Aplicada, Porto Alegre, 2019.

Orientador: Hoppen, Carlos

Tese: Matemática Aplicada,
Grafos Aleatórios, Dominância, Método das Equações Diferenciais

Parâmetros de Dominância em Grafos Regulares

por

Giovane Mansan

Trabalho submetido ao Programa de Pós-Graduação em
Matemática Aplicada do Instituto de Matemática e Estatística da
Universidade Federal do Rio Grande do Sul, como requisito parcial
para a obtenção do título de

Doutor em Matemática Aplicada

Linha de Pesquisa: Matemática Discreta

Orientador: Prof. Dr. Carlos Hoppen

Banca examinadora:

Prof. Dr. Simon Griffiths
Pontifícia Universidade Católica do Rio de
Janeiro

Prof. Dr. Ricardo Misturini
Universidade Federal do Rio Grande do Sul

Prof. Dr. Vilmar Trevisan
Universidade Federal do Rio Grande do Sul

Prof. Dr. Esequia Sauter
Coordenador

AGRADECIMENTOS

Agradeço ao meu orientador, Carlos Hoppen, pela parceria e dedicação no desenvolvimento deste trabalho. Faço um agradecimento especial ao professor Nicholas Wormald pelo desenvolvimento do método que permitiu nossos avanços.

Agradeço também à CAPES pelo suporte financeiro e, sobretudo, aos meus familiares pelo incessante apoio. Também agradeço aos professores do IME/UFRGS que me guiaram durante várias etapas da minha jornada acadêmica.

Sumário

LISTA DE FIGURAS	viii
RESUMO	ix
ABSTRACT	x
1 INTRODUÇÃO	1
1.1 Noções preliminares	1
1.2 Objetivos	4
1.3 Metodologia	6
1.4 Estrutura da tese	9
2 PRELIMINARES	10
2.1 Probabilidade	10
2.2 Grafos e outras estruturas	11
2.3 Grafos aleatórios	13
2.4 Processos para grafos aleatórios	19
3 MOTIVAÇÃO E RESULTADOS	21
3.1 Noções de dominância	21
3.2 Dominância total	23
3.3 k -dominância	26
3.4 Resultados principais	29
3.4.1 Uma relação entre cintura e aleatoriedade em grafos d -regulares	35
4 O METODO DAS EQUAÇÕES DIFERENCIAIS	41

4.1	Condição de Lipschitz	42
4.2	O teorema da aproximação	42
4.3	Exemplo de aplicação do Teorema 4.2	46
5	DOMINÂNCIA TOTAL EM GRAFOS REGULARES	52
5.1	Uma heurística para produzir um conjunto totalmente dominante pequeno	52
5.2	A mudança esperada em cada passo	57
5.3	Algumas propriedades das soluções do sistema de equações diferenciais	62
5.4	Verificando a condição de Lipschitz	69
5.5	Aplicando o Teorema 4.2	70
5.6	Uma consequência da prova do Teorema 3.7	73
6	2-DOMINÂNCIA EM GRAFOS ALEATÓRIOS REGULARES: UM ALGORITMO SIMPLES	77
6.1	Uma heurística para produzir um conjunto 2-dominante pequeno	77
6.2	Analisando o desempenho do Algoritmo 5.1(b)	79
6.3	Formalizando os resultados	84
7	ALGORITMOS COM PRIORIDADE	89
7.1	Prioridade local	89
7.2	Prioridade global	91
7.2.1	Terminologias preliminares	91
7.2.2	Algoritmos despriorizados: uma forma de simular um processo com prioridade global	94

8	MELHORANDO AS COTAS PARA 2-DOMINÂNCIA EM GRAFOS ALEATÓRIOS REGULARES	100
8.1	Definindo as operações do nosso algoritmo	101
8.2	Encontrando as recorrências e as equações diferenciais associadas às operações Op_i do algoritmo dado	104
8.3	Definindo e analisando um algoritmo eficiente	108
8.4	Obtendo condições iniciais adequadas	110
8.5	Estabelecendo condições para a consistência da simulação	118
8.6	Prova do Teorema 3.12	126
9	CONCLUSÃO	128
	REFERÊNCIAS BIBLIOGRÁFICAS	132
	APÊNDICE A EXPERIMENTOS COMPUTACIONAIS	138

Lista de Figuras

Figura 2.1	Esquema de um grafo G com vértices v_1, v_2, \dots, v_6 e arestas $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}, \{v_2, v_5\}, \{v_5, v_6\}, \{v_6, v_4\}$ e $g(G) = 4$	12
Figura 3.1	Grafo de Petersen Generalizado de 16 vértices.	24
Figura 3.2	Grafo de Heawood	26
Figura 5.1	Gráfico das soluções do PVI no caso $d = 3$ com as curvas $z_0(x)$ em vermelho, $z_1(x)$ em azul, $z_2(x)$ em verde e $q(x)$ em preto. .	72

RESUMO

Este trabalho é dedicado ao estudo de cotas superiores para parâmetros de dominância em grafos d -regulares. Nossos resultados foram obtidos por meio da aplicação de um método conhecido e versátil proposto por Wormald que envolve equações diferenciais e aleatoriedade em grafos. Estão envolvidas as áreas de Combinatória, Probabilidade e Análise.

Um grafo G é dito d -regular se todo vértice tem grau d . Em um grafo G , um conjunto $D_T \subseteq V(G)$ é dito totalmente dominante se todo vértice de G possui um vizinho em D_T . Um conjunto $D_k \subseteq V(G)$ é dito k -dominante se todo vértice de $G \setminus D_k$ possui pelo menos k vizinhos em D_k . Os números de dominação total $\gamma_t(G)$ e de k -dominância $\gamma_k(G)$ são as menores cardinalidades possíveis de conjuntos D_T e D_k com essas propriedades, respectivamente, para o grafo G fixado. Neste trabalho, analisamos algoritmos capazes de construir conjuntos totalmente dominantes e 2-dominantes em grafos aleatórios d -regulares. Essas análises estabeleceram cotas superiores, válidas assintoticamente quase certamente, para os números $\gamma_t(G)/|V(G)|$ e $\gamma_2(G)/|V(G)|$ em função de d . As cotas probabilísticas produzidas aqui são melhores do que as cotas determinísticas encontradas na literatura para todos os valores de $d \geq 3$ para os quais a cota foi efetivamente calculada. Adicionalmente, a aplicação de um resultado recente de Hoppen e Wormald permite que essas mesmas cotas sejam válidas como cotas determinísticas para todos os grafos d -regulares G com cintura $g(G)$ suficientemente grande, onde a cintura $g(G)$ de um grafo G é o comprimento do menor ciclo contido em G .

ABSTRACT

This work is dedicated to the study of upper bounds on domination parameters in d -regular graphs. Our results were obtained through the application of a well-known and versatile method proposed by Wormald that involves differential equations and randomness in graphs. The areas of Combinatorics, Probability and Analysis are involved.

A graph G is said to be d -regular if every vertex has degree d . In a graph G , a set $D_T \subseteq V(G)$ is said to be totally dominating if every vertex of G has a neighbor in D_T . A set $D_k \subseteq V(G)$ is k -dominating if every vertex of $G \setminus D_k$ has at least k neighbors in D_k . The total domination number $\gamma_t(G)$ and the k -domination number $\gamma_k(G)$ are the smallest possible cardinalities of sets D_T and D_k with these properties, respectively, for a fixed graph G . In this work, we analyze the performance, in random regular graphs, of algorithms capable of constructing totally dominating sets and 2-dominating sets in d -regular graphs. This analysis leads to upper bounds, valid asymptotically almost surely, on $\gamma_t(G)/|V(G)|$ and $\gamma_2(G)/|V(G)|$ as a function of d . The probabilistic bounds produced here are better than the deterministic bounds found in the literature for all values $d \geq 3$ for which they have been computed. In addition to this, the application of a recent result of Hoppen and Wormald implies that essentially the same bounds are valid as deterministic bounds for all d -regular graphs G with girth $g(G)$ sufficiently large, where the girth $g(G)$ of a graph G is the length of the shortest cycle in G .

1 INTRODUÇÃO

Nas últimas décadas, houve um crescente interesse em certos parâmetros de grafos. Alguns dos parâmetros mais conhecidos estão relacionados a conceitos como *dominância*, *independência* e *número cromático*, por exemplo. Determinar qualquer um desses parâmetros para um grafo arbitrário é um problema *NP*-completo [39] de forma que esses parâmetros se tornaram objeto de muita pesquisa. Habitualmente, o que se faz é tentar estimar essas quantidades, isto é, obtêm-se cotas para esses parâmetros que valem para todos os grafos que satisfazem certas hipóteses. Essas cotas podem ser difíceis de obter e é natural que elas acabem dependendo excessivamente de casos especiais e, por isso, podem ficar muito afastadas dos valores típicos. Uma forma de tentar entender um grafo típico é por meio da teoria dos grafos aleatórios, na qual as cotas são válidas para todos os grafos considerados exceto por uma proporção desprezível do total. Além disso, para evitar a influência excessiva de casos especiais, é comum estudar o valor do parâmetro para alguma família de grafos pré-definida.

1.1 Noções preliminares

Duas formas tradicionais de investigar o comportamento de um parâmetro em um grafo d -regular evitando o efeito de casos especiais consistem em considerar seu valor para grafos que não contêm ciclos curtos e em considerar seu *valor típico*, isto é, seu valor para um grafo d -regular escolhido aleatoriamente de acordo com uma certa distribuição de probabilidade.

Os modelos probabilísticos mais frequentemente abordados na literatura são o modelo binomial $\mathbb{G}_{n,p}$, e o modelo uniforme $\mathbb{G}_{n,M}$ dos grafos com n vértices e M

arestas (ambos conhecidos como modelos de *Erdős-Rényi*) e o modelo uniforme dos grafos d -regulares, denotado por $\mathbb{G}_{n,d}$. Propriedades de grafos aleatórios regulares têm sido intensamente estudadas desde os anos 70, sobretudo envolvendo formas de gerar elementos desse espaço e estabelecendo relações entre seus elementos e parâmetros tradicionais da teoria dos grafos. Um dos resultados mais importantes é o Teorema 2.5, que responde várias questões sobre a presença de ciclos curtos em grafos d -regulares aleatórios arbitrariamente grandes. Para uma resenha de resultados sobre grafos regulares aleatórios, sugerimos Wormald [52].

A *cintura* de um grafo é o comprimento do menor ciclo de um grafo. O efeito da ausência de ciclos curtos em parâmetros de grafos também foi alvo de interesse pelo menos desde que Erdős [23] provou que, para quaisquer inteiros positivos k e g , existe um grafo com cintura pelo menos g e número cromático pelo menos k , o que evidencia a natureza global do número cromático de um grafo. Outro exemplo clássico é o Teorema de Grötzsch [26], que estabelece que todo grafo planar com cintura pelo menos 4 possui número cromático no máximo 3, enquanto que os grafos planares em geral podem ter número cromático igual a 4.

Particularmente, neste trabalho nos aprofundaremos em cotas superiores de dois parâmetros, conhecidos como *número de dominância total* e *número de 2-dominância*, para grafos aleatórios d -regulares e grafos com cintura suficientemente grande.

Seja $G = (V, E)$ um grafo, então dizemos que:

- (1) $D \subseteq V(G)$ é um *conjunto dominante* de G se todo vértice de $V(G) \setminus D$ possui um vizinho em D ;
- (2) $D_T \subseteq V(G)$ é um *conjunto totalmente dominante* de G se todo vértice de $V(G)$ possui um vizinho em D_T ;

- (3) $D_k \subseteq V(G)$ é um *conjunto k -dominante* de G se todo vértice de $V(G) \setminus D_k$ possui k vizinhos em D_k .

O estudo dos parâmetros relacionados à dominância é motivado, por exemplo, por questões de vigilância e cobertura de redes sem fio. Para ilustrar a diferença entre o conceito clássico de *dominância* e o de *dominância total*, consideraremos os seguintes problemas práticos.

Considere um conjunto de posições onde se deseja ter acesso à rede através da instalação de *pontos de acesso*, com roteadores wireless, em algumas dessas posições. Podemos modelar esse problema de modo que cada posição corresponda a um vértice e que duas posições estejam ligadas por uma aresta se o fato de uma delas ser um ponto de acesso implica que a outra tenha acesso à rede. Evidentemente, uma posição que é ponto de acesso também tem acesso à rede. Assim, o problema é escolher um conjunto de pontos de acesso de forma que todas as posições tenham acesso à rede, o que configura um conjunto dominante do grafo das posições. Um problema de otimização natural é permitir acesso universal à rede com o menor número possível de roteadores. Se adicionarmos a exigência de que a falha de até $k - 1$ roteadores não pode afetar o acesso universal à rede das posições que não são pontos de acesso, obtemos uma situação que configura um problema de otimização do número de k -dominância.

Um exemplo que ilustra a dominância total pode ser o seguinte. Considere um conjunto de locais que devam ser monitorados por câmeras de segurança através da escolha de alguns deles para serem pontos de monitoramento, isto é, para serem locais onde as câmeras serão instaladas. Suponhamos que cada local é um vértice e dois locais estão ligados por uma aresta se o fato de um deles ser ponto de monitoramento implica que o outro esteja monitorado por câmeras. Nesse caso, como as configurações de foco impedem que uma câmera filme objetos a distâncias

muito curtas, cada ponto de monitoramento não pode monitorar a si próprio, sendo necessária a existência de um ponto de monitoramento vizinho. Nessas condições, um conjunto de pontos de monitoramento que filma todos os locais é um conjunto totalmente dominante.

A construção de um conjunto dominante mínimo é um problema clássico já abordado em um grande número de trabalhos na literatura. O objetivo costuma ser controlar o tamanho desse conjunto $\gamma(G)$ em função do número de vértices do grafo G , o que sugere que se estude o parâmetro $\gamma(G)/|V(G)|$ conhecido como índice de dominância. Cotas superiores costumam surgir pela execução de algoritmos que servem para criar conjuntos dominantes e independentes, que não são necessariamente dominantes mínimos. De fato, um conjunto $I \subseteq V(G)$ é independente se, para todo par de vértices $\{u, v\} \subseteq I$, for verdade que $\{u, v\} \notin E(G)$. Portanto, um conjunto independente maximal é sempre dominante, e assim um conjunto independente maximal de cardinalidade mínima é uma cota superior para o número de dominação. Sendo assim, um algoritmo eficiente para dominância independente frequentemente é também eficiente para dominância. Por outro lado, um conjunto totalmente dominante não pode ser independente e os algoritmos podem diferir substancialmente dos anteriores.

1.2 Objetivos

Aqui, queremos estudar cotas superiores para algumas versões de dominância já bem exploradas, a dominância total e a 2-dominância, em grafos aleatórios d -regulares e grafos com cintura grande. Por isso, gostaríamos de obter resultados com a forma dos Resultados 1 e 2 abaixo. Os resultados são enunciados para a dominância total, mas também teremos resultados com o mesmo formato para a 2-

dominância. Para enunciar o resultado sobre grafos aleatórios regulares, dizemos que um evento E_n no espaço de grafos aleatórios d -regulares de n vértices vale *assintoticamente quase certamente* se, e somente se, $\mathbb{P}[E_n] \rightarrow 1$ quando $n \rightarrow \infty$.

Resultado 1. *Um grafo aleatório d -regular com n vértices assintoticamente quase certamente possui um conjunto totalmente dominante (ou 2-dominante) de tamanho no máximo $c_1(d)n$.*

Resultado 2. *Existe um inteiro $g > 0$ tal que, para todo n suficientemente grande, todo grafo d -regular de n vértices e cintura maior do que g possui um conjunto totalmente dominante (ou 2-dominante) de tamanho no máximo $c_2(d)n$.*

O objetivo geral é encontrar os menores valores de $c_1(d)$ e $c_2(d)$ para o qual as afirmações acima sejam verdadeiras. Veremos que é possível obter cotas superiores para $c_1(d)$ e $c_2(d)$ consideravelmente melhores do que as melhores cotas existentes para o caso geral.

Nos dois resultados, encontramos cotas que são essencialmente iguais, o que não é tão surpreendente. De fato, não é difícil de provar que qualquer cota $c_2(d) = f(d)$ para o Resultado 2 implica que $c_1(d) = f(d)$ também é válida para para o Resultado 1, conforme veremos na seção 3.4.1. Essa relação é válida para muitos parâmetros além da dominância.

Um resultado recente de Wormald e Hoppen [36] permite a direção contrária sob certas hipóteses. Mais precisamente, permite provar que uma cota $c_1(d) = f(d)$ para o Resultado 1 implica que $c_2(d) = f(d)$ também é válida para para o Resultado 2 desde que a cota $c_1(d)$ do Resultado 1 tenha sido obtida pela análise de um *algoritmo local*, que será tratado brevemente a seguir e com mais detalhes na Seção 5.6.

1.3 Metodologia

A estratégia para obter os resultados é analisar o desempenho médio de um algoritmo local que produz um conjunto com a propriedade desejada no grafo de entrada. Conhecendo esse valor médio, obtemos uma cota superior para a cardinalidade de um conjunto mínimo.

Existem diferentes contextos nos quais se pode definir *algoritmo local* de maneira rigorosa. Uma definição formal compatível com esse trabalho pode ser vista em Wormald e Hoppen [36]. Aqui, definiremos algoritmos locais de uma forma genérica, menos formal, como segue. Num contexto simplificado de coloração de vértices, um algoritmo é um procedimento (cujos passos podem envolver aleatoriedade) que decide a cor de cada vértice em um grafo de entrada G . Em outras palavras, dado um conjunto finito de cores C e um grafo de entrada $G = (V, E)$ (possivelmente com uma coloração inicial), a saída será uma função $f : V(G) \rightarrow C$, que estabelece cores para todos os vértices. Um algoritmo é dito local quando existe um inteiro positivo r tal que a decisão da cor de um vértice v_t específico depende apenas da coloração dos vértices que estão a distância no máximo r de v_t . Além disso, as decisões tomadas para cada vértice são invariantes por isomorfismo. A essa região daremos o nome de *vizinhança de profundidade r de v_t* . Essas definições são similares às de Gamarnik e Sudan [25].

Como exemplo de algoritmo local para construir conjuntos independentes, considere o seguinte algoritmo que utiliza duas cores, que chamaremos de vermelho e branco, isto é, $C = \{\text{vermelho}, \text{branco}\}$. Inicialmente, todos os vértices são brancos. Os vértices de G são ordenados aleatoriamente como v_1, \dots, v_n . Para cada $t \in \{1, \dots, n\}$, o algoritmo decide se o vértice v_t tem sua cor alterada para vermelho ou não. A essa decisão daremos no nome de passo t . O algoritmo

estabelece que essa alteração será feita se, e somente se, nenhum vizinho de v_t for vermelho. Assim, f é a coloração de G obtida após o algoritmo executar todos os seus passos na ordem crescente dada por $t = 1, 2, 3, \dots, n$. É fácil ver que, no final, o conjunto D de vértices vermelhos será um conjunto independente maximal do grafo G . Isso significa que D é independente, mas $D \cup \{v\}$ não é independente por todo $v \in V \setminus D$. Note que, em cada passo t , a decisão do algoritmo só depende da vizinhança de profundidade $r = 1$ de v_t , o que caracteriza um algoritmo local.

Em comparação com algoritmos não-locais, em cada passo, cada decisão depende apenas de uma quantidade limitada de informação, pois suas decisões não dependem da estrutura do grafo além de uma certa distância. Por esse motivo, em algoritmos locais, análises mais precisas são possíveis com métodos tradicionais.

É importante mencionar que a habilidade dos algoritmos locais para aproximar certos parâmetros tem atraído bastante atenção. Recentemente, Gamarnik e Sudan [25] mostraram que, para d suficientemente grande, algoritmos locais não podem aproximar o tamanho do maior conjunto independente para grafos d -regulares com cintura grande com um erro multiplicativo arbitrariamente pequeno. A estimativa de aproximação foi melhorada por Rahman e Virág [44], que provaram que, quando $d \rightarrow \infty$, o tamanho do maior conjunto independente produzido por algoritmos locais converge para a metade do tamanho do maior conjunto independente existente. Recentemente, Chen, Gamarnik, Panchenko e Rahman [16] provaram que os algoritmos locais também não fornecem as melhores cotas para alguns problemas de corte máximo em hipergrafos.

Nossa estratégia para obter os Resultados 1 e 2 da seção anterior é a seguinte. Usaremos o método devido a Wormald [55], conhecido como o Método das Equações Diferenciais (ou MED), para analisar o desempenho de algoritmos locais específicos que retornam conjuntos totalmente dominantes e conjuntos 2-

dominantes para qualquer grafo d -regular de entrada. Para cada algoritmo, isso produzirá um teorema com o formato do Resultado 1. Isto é formalizado pelos Teoremas 3.7, 3.9 e 3.11. Traduziremos, então, estes resultados para todos os grafos com cintura suficientemente grande usando [36, Theorem 5.5], de forma que obtaremos consequências do Resultado 1 com o formato do Resultado 2, o que produz os Teoremas 3.8, 3.10 e 3.12. Assim, nossas contribuições são obtidas pela análise dos Algoritmos 4.1(a), 5.1(a) e 8(b), que desenvolvemos, e pela aplicação dos métodos mencionados, já conhecidos na literatura. O Método das Equações Diferenciais é bastante geral e foi aplicado em muitos contextos envolvendo processos aleatórios. Por exemplo, em Fiz Pontiveros, Griffiths e Morris [43] temos uma aplicação desse método fora do contexto de grafos d -regulares. Por outro lado, esse método já foi usado para estudar parâmetros relacionados a dominância, veja [21, 22], e resultados para grafos com cintura grande usando a abordagem geral descrita acima foram provados em [37].

Cabe mencionar que grafos aleatórios regulares não são o único modelo probabilístico que permite obter resultados para grafos com cintura grande. Em Backhausz e Szegedy [7], processos em grafos d -regulares com cintura grande são comparados com processos na árvore d -regular infinita, que é única a menos de automorfismos. Isso exemplifica uma outra maneira de caracterizar grafos d -regulares com cintura grande.

Essencialmente, o método faz uso das teorias da probabilidade, da combinatoria e da análise. Por isso, optamos por incluir um capítulo inicial abordando alguns tópicos dessas teorias.

1.4 Estrutura da tese

Essa tese está organizada da seguinte forma.

No Capítulo 2 introduzimos a formalização de alguns conceitos básicos de probabilidade, teoria dos grafos e aleatoriedade em grafos a serem usados no decorrer da tese. Esse capítulo pode ser dispensado a critério do leitor. Depois disso, no Capítulo 3, faremos uma discussão preliminar dos parâmetros abordados aqui e também é onde enunciamos de maneira formal os resultados obtidos nesta tese.

Enunciaremos, no Capítulo 4, um teorema técnico conhecido como o *Método das Equações Diferenciais*, que permitiu a análise do nosso algoritmo. Neste momento, apresentamos um exemplo básico de aplicação direta do método. Esse exemplo será retomado no Capítulo 6.

O Capítulo 5 é onde usamos o método para provar nossos resultados. Aqui, como no exemplo da Seção 4.3 do Capítulo 4, descrevemos o algoritmo, deduzimos as recorrências necessárias ao teorema técnico e construímos o problema de valor inicial associado.

Abordamos, no Capítulo 6, a 2-dominância em grafos d -regulares com $d \geq 4$ por meio de um algoritmo que é praticamente uma extensão daquele usado na Seção 4.3 do Capítulo 4 e neste caso, o problema de valor inicial permitiu uma solução analítica, diferentemente do que ocorreu no Capítulo 5.

No Capítulo 7 são abordadas algumas questões relacionadas à *prioridade* em algoritmos locais para serem usadas no Capítulo 8, que será dedicado a analisar um algoritmo que apresenta cotas superiores melhores do que aquelas apresentadas no Capítulo 6.

2 PRELIMINARES

Nosso estudo envolve noções de probabilidade, teoria dos grafos e aleatoriedade em grafos. Neste capítulo, formalizaremos alguns conceitos básicos sobre essas áreas, que darão suporte à nossa argumentação posterior. Essa teoria pode ser encontrada em obras consagradas da literatura, assim, esse capítulo pode ser dispensado a critério do leitor.

2.1 Probabilidade

Dado um conjunto Ω , uma σ -álgebra \mathcal{A} sobre Ω e uma medida de probabilidade \mathbb{P} sobre \mathcal{A} , dizemos que uma tripla $(\Omega, \mathcal{A}, \mathbb{P})$ é um *espaço de probabilidades* ou um *modelo probabilístico*. Se Ω é um conjunto finito, então $(\Omega, \mathcal{A}, \mathbb{P})$ é dito um *espaço de probabilidade finito*. Chamamos de *variável aleatória* uma função $X : \Omega \rightarrow S$ onde S é um conjunto de valores reais.

Dados um espaço de probabilidade finito $(\Omega, 2^\Omega, \mathbb{P})$ e uma variável aleatória $X : \Omega \rightarrow S$, a *esperança* de X , denotada por $\mathbb{E}[X]$, será definida por

$$\mathbb{E}[X] = \sum_{\omega \in \Omega} \mathbb{P}[\omega] X(\omega) = \sum_{s \in S} s \mathbb{P}[X = s].$$

A *variável indicadora* do evento A , denotada por 1_A , é definida por

$$1_A(\omega) := \begin{cases} 1 & \text{se } \omega \in A \\ 0 & \text{se } \omega \notin A \end{cases}$$

Note que $\mathbb{E}[1_A] = \mathbb{P}[A]$.

Proposição 2.1 (Desigualdade de Markov). *Seja X uma variável aleatória não-negativa e t um número real positivo, então*

$$\mathbb{P}[X \geq t] \leq \frac{\mathbb{E}[X]}{t}.$$

Demonstração. Basta notar que

$$\mathbb{E}[X] \geq \mathbb{E}[X \cdot 1_{X \geq t}] \geq \mathbb{E}[t \cdot 1_{X \geq t}] = t \cdot \mathbb{P}[X \geq t].$$

□

Proposição 2.2 (Desigualdade de Chebyshev). *Seja X uma variável aleatória não-negativa e t um número real positivo, então*

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq t] \leq \frac{\sigma(X)^2}{t^2},$$

onde $\sigma(X) = \sqrt{\mathbb{E}[(X - \mathbb{E}[X])^2]}$.

Demonstração. Note que

$$\mathbb{P}[|X - \mathbb{E}[X]| \geq t] = \mathbb{P}[(X - \mathbb{E}[X])^2 \geq t^2].$$

Agora basta aplicar a Desigualdade de Markov em $Y = (X - \mathbb{E}[X])^2$.

□

2.2 Grafos e outras estruturas

Chamamos de *grafo* o par denotado por $G = (V, E)$ onde $V = V(G) = \{v_1, \dots, v_n\}$ é um conjunto finito e

$$E(G) \subseteq \{e \subseteq V(G) : |e| = 2\}.$$

Os elementos de V são os *vértices* de G e os elementos de E são as *arestas* de G . Dizemos que os vértices u e v são *adjacentes* se $\{u, v\} \in E(G)$.

Dado um grafo $G = (V, E)$, o *grau* de um vértice $v \in V$, denotado por $\deg_G(v)$, é definido por $\deg_G(v) := |\{u : \{u, v\} \in E\}|$. Fixe $d \in \{1, 2, \dots\}$, um grafo G é *d-regular* se todo vértice de G tem grau d . Dado $k \geq 3$, um *ciclo de comprimento k* em G é um conjunto $\{v_1, \dots, v_k\}$ de vértices de G tal que $\{v_k, v_1\} \in E$ e $\{v_i, v_{i+1}\} \in E$ para todo $1 \leq i \leq k - 1$. A *cintura* do grafo G , denotada por $g(G)$, é o menor comprimento de um ciclo de G . Um grafo G é dito *acíclico* ou *floresta* se não possui ciclos de nenhum comprimento. Neste caso, escrevemos $g(G) := \infty$.

O *grau máximo* e o *grau mínimo* de um grafo G , denotados, respectivamente, por $\Delta(G)$ e $\delta(G)$, são definidos por

$$\Delta(G) := \max\{\deg(v) : v \in V(G)\},$$

$$\delta(G) := \min\{\deg(v) : v \in V(G)\}.$$

Note que um grafo G é *d-regular* se, e somente, $\Delta(G) = \delta(G)$.

Um grafo pode ser esquematizado em um desenho com os vértices representados por pontos e as arestas por segmentos de reta ligando esses pontos. A Figura 2.1 ilustra como isso pode ser feito..

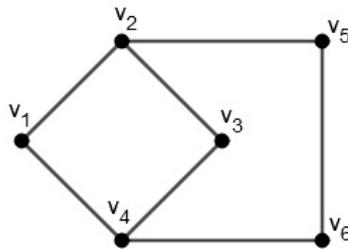


Figura 2.1: Esquema de um grafo G com vértices v_1, v_2, \dots, v_6 e arestas $\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}, \{v_2, v_5\}, \{v_5, v_6\}, \{v_6, v_4\}$ e $g(G) = 4$.

Um grafo $G = (V, E)$ de n vértices é dito *completo*, e denotado por K_n , se $E(G) = \{e \subseteq V(G) : |e| = 2\}$.

Um *multigrafo* $M = (V, E)$ é uma estrutura definida similarmente a um grafo, onde a principal diferença é que, em vez de conjuntos, usamos *multiconjuntos*, que diferem dos conjuntos por permitirem elementos repetidos. Como consequência, um multigrafo admite arestas repetidas e *laços* (uma aresta ligando um vértice a ele próprio). Fixado V finito com $|V| = n$, há uma quantidade finita $2^{\binom{n}{2}}$ de grafos distintos possíveis $G = (V, E)$, mas para multigrafos $M = (V, E)$ essa quantidade é infinita, a menos que se limite o número de arestas.

2.3 Grafos aleatórios

Um *grafo aleatório* é um grafo $G = (V, E)$ gerado segundo algum critério aleatório (como um sorteio, por exemplo). De forma análoga se define *multigrafo aleatório*.

Existem muitos modelos de grafos aleatórios, sendo que os mais famosos são $\mathbb{G}_{n,M}$ e $\mathbb{G}_{n,p}$. Dados inteiros M e n tais que $0 \leq M \leq \binom{n}{2}$, o espaço denotado por $\mathbb{G}_{n,M}$, é um espaço uniforme composto por todos os grafos de n vértices e M arestas.

Um modo natural de gerar um grafo de $\mathbb{G}_{n,M}$ pode ser descrito como um algoritmo. Comece com um grafo de n vértices, todos isolados, e, em cada passo, adicione uma aresta escolhendo um par que ainda não forma uma aresta com probabilidade uniforme. Após exatamente M passos, teremos gerado um elemento de $\mathbb{G}_{n,M}$. Esse modo de gerar grafos, passando por etapas, é parte de algo mais geral conhecido como *processo aleatório* de grafos, conforme apresentado, por exem-

plo, em Wormald [55]. Nele, o objeto construído evolui segundo alguma regra estocástica para, no final, produzir um grafo do espaço de probabilidade desejado.

Dado um número real p , com $0 \leq p \leq 1$, e $n \in \mathbb{N}$, um grafo aleatório do espaço $\mathbb{G}_{n,p}$, conhecido como o modelo de *Erdős-Rényi* [24], é um grafo de n vértices onde cada aresta está presente com probabilidade p de maneira independente. Evidentemente, quando $p = 1/2$, $\mathbb{G}_{n,p}$ é um espaço uniforme.

Sejam $n > d \geq 3$ inteiros não-negativos tais que nd é par, denotamos por $\mathbb{G}_{n,d}$ o espaço de probabilidade uniforme composto por todos grafos rotulados com conjunto de vértices $\{1, 2, \dots, n\}$ tais que cada vértice possui grau d . $\mathbb{G}_{n,d}$ é chamado de espaço dos grafos aleatórios *d-regulares* de n vértices.

Uma dificuldade que surge no estudo desse espaço é, por exemplo, a contagem exata do número de grafos *d-regulares* de n vértices. Não há uma fórmula explícita simples que forneça esse número em função de n e d , diferentemente do que ocorre com os espaços $\mathbb{G}_{n,p}$ e $\mathbb{G}_{n,M}$ descritos anteriormente. Sem essa quantia, não temos como saber a probabilidade de um grafo específico ser sorteado. Mesmo assim podemos estudar esse espaço acoplando-o a um espaço de probabilidade conveniente, apresentado por Bollobás [10] como *modelo das configurações* e, independentemente, por Wormald [52] como *modelo de emparelhamentos*. Faremos uma breve descrição desse modelo a seguir.

Considere um conjunto de nd pontos particionados em n células v_1, v_2, \dots, v_n com d pontos em cada célula. Chamaremos de *emparelhamento* um conjunto $P = \{p_1, \dots, p_{nd/2}\}$ tal que cada p_i é um par não-ordenado de pontos e cada ponto está precisamente em um par p_i . Seja Ω_{nd} o conjunto de todos os emparelhamentos distintos possíveis desse conjunto de nd pontos. Temos que

$$|\Omega_{nd}| = \frac{(nd)!}{(nd/2)!2^{nd/2}}.$$

Basta definir que cada um desses emparelhamentos tenha probabilidade $1/|\Omega_{nd}|$ de ocorrer, para completarmos a construção de $\mathcal{P}_{n,d}$ como um espaço de probabilidade uniforme, cujos elementos são todos os conjuntos $P = \{p_1, \dots, p_{nd/2}\}$.

Agora, vamos fazer uma correspondência entre emparelhamentos e multigrafos (com laços e arestas duplas permitidas) de uma forma natural. Cada par $\{i, j\}$ de pontos com $i \in u$ e $j \in v$ corresponde a uma aresta $\{u, v\}$ no multigrafo associado. Assim, cada emparelhamento está associado a um único multigrafo, mas o mesmo multigrafo pode estar associado a vários emparelhamentos. Dizemos que um multigrafo é *simples* se for um grafo (isto é, não possui laços nem arestas duplas). Neste caso, o multigrafo é um grafo d -regular de n vértices. Cada um desses grafos está associado a exatamente $(d!)^n$ emparelhamentos. Conseqüentemente, multigrafos simples são sorteados de maneira uniforme pelo modelo de emparelhamentos. Já os multigrafos não-simples não são gerados com a mesma probabilidade, já que o número de emparelhamentos associados a um multigrafo específico depende do número de laços e arestas presentes nesse multigrafo. Precisamente, para termos a quantidade de emparelhamentos associadas a um multigrafo específico, devemos tomar $(d!)^n$ e dividir:

- (a) por 2 para cada laço existente,
- (b) por $k!$ para cada aresta que é se repete k vezes.

Entretanto, ainda devemos estimar a probabilidade de que um multigrafo sorteado por $P_{n,d}$ seja grafo. Seu valor assintótico ($n \rightarrow \infty$) já foi calculado por Bender e Canfield [6]. Precisamente, esse cálculo fornece que, se um emparelhamento P for escolhido por $P_{n,d}$, então

$$\lim_{n \rightarrow \infty} \mathbb{P}[P \text{ gera um grafo simples}] = e^{(1-d^2)/4}.$$

Essa propriedade garante que, para d fixo, a proporção de emparelhamentos que são simples não tende a zero quando $n \rightarrow \infty$. Em outras palavras, a probabilidade de um elemento sorteado por $P_{n,d}$ corresponder a um grafo d -regular é maior do que o número positivo, digamos $e^{(1-d^2)/4}/2$, para todo n suficientemente grande. Portanto, para todo n suficientemente grande, um evento que ocorre com probabilidade tendendo a 1 em $P_{n,d}$ também ocorre com probabilidade tendendo a 1 quando condicionado ao evento $\{P : P \text{ é simples}\}$. A consequência prática disso é que se provarmos que um multigrafo de $P_{n,d}$ satisfaz certa propriedade assintoticamente quase certamente, então $\mathbb{G}_{n,d}$ satisfaz a mesma propriedade assintoticamente quase certamente.

Seja $(\mathbb{A}_n)_{n \in \mathbb{N}} = ((\Omega_n, \mathcal{A}_n, \mathbb{P}_n))_{n \in \mathbb{N}}$, uma sequência de espaços de probabilidade. Uma *propriedade* P de $(\mathbb{A}_n)_{n \in \mathbb{N}}$ é uma sequência (A_1, A_2, \dots) , tal que cada $A_n \in \mathcal{A}_n$. Geralmente, uma propriedade é declarada na forma de uma lei que seja natural em todos os espaços da sequência. Por exemplo, em $\mathbb{G}_{n,d}$, podemos declarar a propriedade: “não há vértices isolados”. Formalmente, ela corresponde a lista (A_1, A_2, \dots) , onde cada A_n é o conjunto de todos os grafos de n vértices que não possuem vértices isolados.

Definição 2.3. *Seja $(\mathbb{A}_n)_{n \in \mathbb{N}} = ((\Omega_n, 2^{\Omega_n}, \mathbb{P}_n))_{n \in \mathbb{N}}$ uma sequência de espaços de probabilidade, e seja $P = (A_1, A_2, \dots)$ uma propriedade de $(\mathbb{A}_n)_{n \in \mathbb{N}}$, então dizemos que a propriedade P é satisfeita **assintoticamente quase certamente** em \mathbb{A}_n se*

$$\lim_{n \rightarrow \infty} \mathbb{P}_n[A_n] = 1.$$

Particularmente, esse conceito é útil para caracterizar grafos aleatórios arbitrariamente grandes.

Por simplicidade, podemos nos referir a propriedades que ocorrem assintoticamente quase certamente apenas afirmando que ocorrem com *alta probabilidade*.

As observações abaixo explicam alguns fenômenos sobre a distribuição dos ciclos em grafos regulares grandes, caracterizando relações entre cintura e aleatoriedade. Isto será necessário para compreender a Seção 3.4.1.

Definição 2.4. Dizemos que uma variável aleatória X no conjunto $\{0, 1, 2, \dots\}$ tem Distribuição de Poisson com valor esperado $\lambda > 0$ se

$$\mathbb{P}[X = s] = \frac{e^{-\lambda} \lambda^s}{s!}, \quad s = 0, 1, 2, \dots$$

De forma simplificada, denotamos $X \sim \text{Poisson}(\lambda)$.

Teorema 2.5 (Bollobás [10], Wormald [53, 54]). Para um inteiro positivo d , seja $X_i(G) = X_{i,n}(G)$ ($i \geq 3$) o número de ciclos de comprimento i em $G \in \mathbb{G}_{n,d}$. Então, para cada $k \geq 3$ fixo e para $n \rightarrow \infty$, X_3, \dots, X_k convergem para variáveis aleatórias independentes com Distribuição de Poisson com valor esperado $\lambda_i = \frac{(d-1)^i}{2^i}$, para cada $3 \leq i \leq k$.

Estamos particularmente interessados em controlar a proporção de ciclos curtos em grafos aleatórios grandes, por isso formalizamos as seguintes quantidades.

Definição 2.6. Dado um grafo $G \in \mathbb{G}_{n,d}$ e $g_0 > 0$, então

- (a) $X_{n,d}^{g_0}(G)$ é definido como o número de ciclos de G com comprimento menor do que g_0 .
- (b) $A_{n,d}(g_0) := \{G \in \mathbb{G}_{n,d} : X_{n,d}^{g_0}(G) < \ln(n)\}$, isto é, $A_{n,d}(g_0)$ é o conjunto dos grafos de $\mathbb{G}_{n,d}$ com menos de $\ln(n)$ ciclos de comprimento menor do que g_0 .

Corolário 2.7. Para quaisquer $\epsilon, g_0 > 0$ existe $n_0 \in \mathbb{N}$ tal que, se $n \geq n_0$, então

$$|A_{n,d}(g_0)| > (1 - \epsilon)|\mathbb{G}_{n,d}|.$$

Demonstração. Pelo Teorema 2.5, existe $n_1 \in \mathbb{N}$ tal que, se $n \geq n_1$, então

$$\mathbb{E}[X_{n,d}^{g_0}] \leq 2 \sum_{i=3}^{g_0-1} \frac{(d-1)^i}{2^i} < (d-1)^{g_0}.$$

Isso, juntamente com a Desigualdade de Markov 2.1, fornece

$$\mathbb{P}[X_{n,d}^{g_0} \geq \ln(n)] \leq \frac{(d-1)^{g_0}}{\ln(n)}.$$

Agora, tomemos $n_2 \in \mathbb{N}$ tal que, se $n \geq n_2$,

$$\frac{(d-1)^{g_0}}{\ln(n)} < \epsilon.$$

Assim, tomando $n_0 = \max\{n_1, n_2\}$, podemos concluir que, para todo $n \geq n_0$,

$$\frac{|A_{n,d}(g_0)|}{|\mathbb{G}_{n,d}|} = \mathbb{P}[X_{n,d}^{g_0} < \ln(n)] \geq 1 - \epsilon$$

□

Corolário 2.8. Dados $g_0, d \geq 3$, existe n_0 tal que, para todo $n > n_0$ tal que dn é par, existe um grafo d -regular de n vértices com cintura pelo menos g_0 .

Demonstração. Note que, em geral, qualquer evento com probabilidade não nula deve possuir pelo menos um elemento. Pelo Teorema 2.5,

$$\lim_{n \rightarrow \infty} \mathbb{P}[X_{n,d}^{g_0} = 0] = \prod_{i=3}^{g_0-1} \frac{e^{-\lambda_i} \lambda_i^0}{0!} = \exp\left(-\sum_{i=3}^{g_0-1} \lambda_i\right),$$

onde $\lambda_i = \frac{(d-1)^i}{2^i}$. Agora, basta tomar n_0 tal que para todo $n > n_0$ tenhamos

$$\mathbb{P}[X_{n,d}^{g_0} = 0] > \frac{1}{2} \exp\left(-\sum_{i=3}^{g_0-1} \lambda_i\right) > 0.$$

□

2.4 Processos para grafos aleatórios

Olharemos para espaços de grafos aleatórios através de processos aleatórios utilizados para gerá-los. Iniciamos com um grafo G_0 de n vértices, geralmente todos isolados, e vai-se acrescentando arestas repetidamente de forma aleatória segundo alguma regra estocástica. Isso determina um espaço de probabilidade de sequências G_0, G_1, G_2, \dots onde G_t é o t -ésimo grafo do processo. Se nenhuma aresta é adicionada em um certo instante M , exigiremos que $G_{i+1} = G_i$ para todo $i \geq M$ e diremos que G_M é o grafo final do processo. Esse M será dito o *tempo de parada natural do processo*. Trataremos de grafos finitos. Neste caso, o tempo de parada é finito, pois o número de arestas acrescentadas em cada passo é positivo e o número total de arestas é limitado por $n(n-1)/2$, onde n é o número de vértices do grafo. Como consequência, o espaço das listas G_0, G_1, G_2, \dots também será finito. Além disso, vamos descartar as listas G_0, G_1, G_2, \dots que forem impossíveis segundo a respectiva regra: por exemplo, se a regra diz que em cada passo adicionamos uma aresta, não há como o grafo G_t ter $2t$ arestas. Assim, esses casos, que teriam probabilidade nula, não estão inclusos no espaço de probabilidade das listas G_0, G_1, G_2, \dots . Portanto, cada lista contida no espaço tem probabilidade maior do que zero.

Um dos aspectos que será particularmente interessante para nós é entender a sequência de graus do grafo ao longo do processo. Para tanto, vamos definir as variáveis $Y_i(t)$ como o número de vértices de grau i em G_t . Consideraremos sequências que iniciam com um grafo G_0 com todos os vértices isolados, isto é, com $E(G_0) = \emptyset$. Como as variáveis Y_i dependem de n , estas poderão, em certos casos, serem estudadas quando $n \rightarrow \infty$. Eventualmente outras variáveis podem ser definidas. Sempre que nos referirmos a ordem de crescimento $O(\cdot)$, esta será

com respeito a n e uniforme com relação a outras quantidades. Assim, vamos usar a mesma notação usada em [38]:

Em particular, estamos interessados em gerar um grafo d -regular, com n vértices, de maneira uniforme, por meio de algum processo adequado. Como vimos na Seção 2.3, o espaço dos grafos d -regulares não se parece muito com os outros espaços definidos na Seção 2.3. Na verdade, gerar um grafo d -regular aleatório de alguma maneira, mesmo que não seja necessariamente um processo, já não é uma tarefa tão trivial quanto gerar um grafo aleatório nos espaços $\mathbb{G}_{n,p}$ e $\mathbb{G}_{n,M}$.

Em termos de “número de tentativas”, o modelo dos emparelhamentos não é a forma mais eficiente de gerar grafos aleatórios regulares, já que quanto maior for d , menor a chance de o multigrafo gerado ser um grafo d -regular. Apesar disso, o modelo dos emparelhamentos já atende nossos objetivos, e, por isso, será o que usaremos neste trabalho, mas cabe mencionar que há formas mais eficientes de gerar grafos aleatórios d -regulares. Temos, por exemplo, um algoritmo conhecido como *d-process* estudado em Ruciński e Wormald (1992) [46]. Ele é um processo aleatório de grafos bem simples, como segue. Dados d e n , com dn par, comece com n vértices isolados e adicione repetidamente arestas ligando vértices de grau estritamente menor do que d . Em cada instante uma aresta é adicionada em uma das posições vagas com probabilidade uniforme. O processo acaba quando todos os vértices têm grau d ou quando todos os vértices que têm grau menor que d já estão ligados por aresta. Em [46] foi provado que este algoritmo gera um grafo d -regular assintoticamente quase certamente. Ainda é uma questão em aberto se esse algoritmo gera grafos aleatórios d -regulares com probabilidade uniforme.

3 MOTIVAÇÃO E RESULTADOS

3.1 Noções de dominância

Há uma variedade de parâmetros que estão relacionados à noção geral de dominação em grafos que já foram abordados na literatura. A noção mais estudada é a *dominância*, a qual nos referimos como *dominância clássica*. Ela é definida da seguinte forma. Dado um grafo $G = (V, E)$ e um subconjunto de vértices qualquer $V' \subseteq V(G)$, dizemos que um vértice $v \in V(G)$ é *dominado* pelo conjunto V' se for adjacente a algum vértice de V' . Dizemos que $\tilde{V} \subseteq V(G)$ é um conjunto dominante de G se todo $v \in V \setminus \tilde{V}$ é dominado por \tilde{V} , isto é, cada vértice de $V(G)$ deve ter um vizinho em \tilde{V} ou estar em \tilde{V} . O *número de dominação*, denotado por $\gamma(G)$, é definido por

$$\gamma(G) = \min\{|\tilde{V}| : \tilde{V} \text{ é um conjunto dominante de } G\}.$$

Dado um grafo $G = (V, E)$, dizemos que $D_T \subseteq V(G)$ é um *conjunto totalmente dominante* de G se todo $v \in V(G)$ for dominado por D_T . Certamente, qualquer conjunto totalmente dominante é também um conjunto dominante clássico, e um grafo G admite um conjunto totalmente dominante se, e somente se, G não possui vértices isolados.

Naturalmente, o *número de dominação total*, denotado por $\gamma_t(G)$, é definido por

$$\gamma_t(G) = \min\{|D_T| : D_T \text{ é um conjunto totalmente dominante de } G\},$$

isto é, $\gamma_t(G)$ é o menor tamanho possível de um conjunto totalmente dominante de G .

Em Hedetniemi e Laskar [29] temos uma revisão bibliográfica de artigos sobre a dominação em grafos. Ali, mostra-se que as publicações sobre essa teoria surgiram nos anos 50, com um crescimento significativo por volta dos anos 70. A partir dessa data, o ritmo se manteve até os dias atuais. A *dominância total* foi introduzida por Cockayne, Dawes e Hedetniemi [17], e desde então vem sendo amplamente explorada como parte da teoria dos grafos.

Determinar o tamanho de um conjunto dominante mínimo é notoriamente um problema difícil, que aparece na lista original de problemas NP-completos de Karp [39]. Determinar o tamanho de um conjunto totalmente dominante mínimo também é um problema NP-completo, como vemos em Pfaff, Laskar e Hedetniemi [42]. Para mais resultados e referências sobre dominação clássica e total, recomendamos as resenhas de Haynes, Hedetniemi e Slater [28] e Henning e Yeo [31], respectivamente.

Cada versão de dominância parece ser estudada de forma essencialmente independente de demais. Naturalmente, um resultado associado a uma versão de dominância parece não ter implicações diretas para as outras versões. No entanto, há relações bem básicas entre $\gamma_t(G)$ e $\gamma(G)$. Uma delas, por exemplo, é o seguinte teorema, encontrado em Bollobás e Cockayne (1979) [12].

Teorema 3.1. *Para todo grafo G sem vértices isolados, $\gamma(G) \leq \gamma_t(G) \leq 2\gamma(G)$.*

Demonstração. A desigualdade da esquerda segue do fato de que qualquer conjunto totalmente dominante também é dominante. A desigualdade da direita segue do fato de que qualquer conjunto dominante \tilde{V} pode ser estendido a um conjunto totalmente dominante pela adição de até um vizinho de cada vértice de \tilde{V} . \square

3.2 Dominância total

No caso de um grafo G com n vértices, podemos fornecer uma cota inferior natural em função do seu grau máximo $\Delta(G)$, dada por

$$\gamma_t(G) \geq \left\lceil \frac{n}{\Delta(G)} \right\rceil, \quad (3.1)$$

já que cada vértice domina totalmente no máximo $\Delta(G)$ vértices. Essa cota é justa, como ilustram os grafos bipartidos completos, que discutiremos a seguir.

Em geral, o tamanho de um conjunto totalmente dominante mínimo pode variar consideravelmente entre os grafos d -regulares com n vértices. Por exemplo, se G é uma coleção de grafos bipartidos completos disjuntos $K_{d,d}$, teremos $\gamma_t(G) = n/d$, pois cada componente é totalmente dominada se escolhermos um vértice de cada lado da bipartição. Isto mostra que a cota inferior trivial para $\gamma_t(G)$ mencionada acima é justa para todo d . Por outro lado, se G é uma coleção de grafos completos disjuntos K_{d+1} , teremos $\gamma_t(G) = 2n/(d+1)$, que é substancialmente maior. A Tabela 3.1 apresenta cotas superiores e inferiores para $\gamma_t(G)$ em função do grau d dos vértices de um grafo regular G . Veremos que os limites superiores da tabela são justos para $d \in \{3, 4\}$.

Tipicamente, as melhores cotas superiores encontradas para grafos d -regulares são cotas para um conjunto de grafos um maior: grafos de grau mínimo $\delta(G) = d$. No livro *Total Domination in Graphs* de Henning e Yeo [31], podemos encontrar as mais variadas propriedades do número de dominação total. Dentre elas, podemos destacar, por exemplo, o seguinte teorema.

Teorema 3.2 ([31]). *Se G é um grafo de n vértices e com grau mínimo $\delta(G) \geq 1$, então*

$$\gamma_t(G) \leq \left(\frac{1 + \ln(\delta)}{\delta} \right) n.$$

Naturalmente, o Teorema 3.2 pode ser usado para fornecer uma cota superior para os grafos d -regulares. Há melhoramentos recentes das cotas desse teorema para valores pequenos de $\delta(G)$. Por exemplo, Archdeacon et al. [4] provaram que, se $\delta(G) \geq 3$, então $\gamma_t(G) \leq n/2$ e, o mais importante, essa cota é justa. Um exemplo de grafo G no qual se tem a igualdade $\gamma_t(G) = n/2$ é dado pelo *Grafo de Petersen Generalizado*, que é um grafo *cúbico* (3-regular) com 16 vértices.

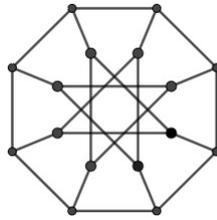


Figura 3.1: Grafo de Petersen Generalizado de 16 vértices.

É possível construir explicitamente a família de todos os grafos satisfazendo a igualdade, porém não vamos fazê-lo aqui para não desviar nosso foco, que está em melhorar essas cotas. Para leitores interessados, recomendamos o livro Henning e Yeo [31]. Para nós, a utilidade disso consiste, particularmente, em obter a igualdade $\gamma_t(G) = n/2$ para valores de n arbitrariamente grandes. Basta, por exemplo, tomar várias cópias disjuntas do Grafo de Petersen Generalizado acima. Abaixo, temos uma tabela com as melhores cotas conhecidas de γ_t para $\delta(G) = 3, 4$ e 5.

δ	Cota inferior (trivial)	Cota superior já publicada	Publicação ou fonte
3	$1/3 \simeq 0.3333$	$1/2 = 0.5000$	[4] (2004)
4	$1/4 = 0.2500$	$3/7 \simeq 0.4285$	[50] (2007)
5	$1/5 = 0.2000$	$17/44 \simeq 0.3863$	[20] (2015)
6	$1/6 \simeq 0.1666$	0,4653	Teorema 3.2

Tabela 3.1: Limitações conhecidas para $\gamma_t(G)/|V(G)|$ em função de δ , todas as cotas inferiores e as cotas superiores para $\delta(G) \in \{3, 4\}$ são justas

Para $\delta \geq 6$, podemos obter cotas usando o Teorema 3.2. Mas note que a cota para $\delta(G) = 6$, que, intuitivamente, deveria ser pelo menos tão boa quanto para $\delta(G) = 5$, está pior na tabela, pois estamos utilizando simplesmente a cota geral do Teorema 3.2.

Já foi provado que a cota da tabela para $\delta(G) = 4$ é justa assim como para $\delta(G) = 3$. O teorema que garante isso pode ser encontrado no livro [31] e tem o seguinte enunciado.

As seguintes definições esclarecem a igualdade no Teorema 3.4

Definição 3.3 (Complemento bipartido). *Dados dois grafos G e \bar{G} com $V(G) = V(\bar{G}) = \{v_1, \dots, v_{2m}\}$, dizemos que \bar{G} é um complemento bipartido de G se existe um grafo bipartido completo $H = K_{m,m}$ contendo G tal que $E(\bar{G}) = E(H) \setminus E(G)$.*

Na Figura 3.2, temos o grafo de Heawood.

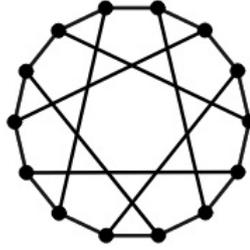


Figura 3.2: Grafo de Heawood

Teorema 3.4. *Se G é um grafo conexo de ordem n com $\delta(G) \geq 4$, então $\gamma_t(G) \leq 3n/7$, com igualdade se, e somente se, G é um complemento bipartido do grafo de Heawood.*

Ainda não se sabe se há algum grafo G com $\delta(G) \geq 5$ tal que $\gamma_t(G) = 17n/44$. Inclusive, há uma conjectura, devida a Thomassé e Yeo [50], de que, se $\delta(G) \geq 5$, então $\gamma_t(G) \leq 4n/11 \sim 0,3636n$. Dentre as cotas probabilísticas que deduziremos mais à frente, veremos que a cota para grafos 5-regulares é ainda um pouco melhor que a dessa conjectura, ou seja, o valor conjecturado é pelo menos uma cota superior para quase todos os grafos 5-regulares.

Com relação ao efeito que a cintura grande tem sobre o número de dominação total de um grafo d -regular, Henning e Yeo [32] abordaram um caso particular dessa questão, onde eles mostraram que, se G é um grafo de n vértices com $\delta(G) \geq 2$ e cintura $g \geq 3$, então

$$\gamma_t(G) \leq \left(\frac{1}{2} + \frac{1}{g}\right)n. \quad (3.2)$$

3.3 k -dominância

A k -dominância é uma generalização da dominância clássica que considera o número de vezes que um vértice do grafo deve ser dominado pelo conjunto de

vértices dominantes. Dado $G = (V, E)$ um grafo, dizemos que $D \subseteq V$ é um conjunto k -dominante de G se, para todo $v \in V \setminus D$, v é adjacente a pelo menos k elementos de D . Também seria natural considerar uma noção de k -dominância total em que cada vértice do grafo deva ser dominado por pelo menos k outros vértices. Porém optamos por abordar uma versão mais próxima da dominância clássica, isto é, os vértices que dominam não precisam ser dominados.

Definimos

$$\gamma_k(G) := \inf\{|D| : D \text{ é um conjunto } k \text{ dominante de } G\}$$

como o *número de k -dominação de G* .

Uma limitação inferior trivial para $\gamma_k(G)$ em função do grau máximo $\Delta = \Delta(G)$, pode ser obtida por considerar que cada vértice de G domina no máximo $\Delta(G)$ outros vértices. De fato, se $D \subseteq V(G)$ é um conjunto k -dominante de G e $n = |V(G)|$, então

$$(n - |D|)k \leq |D|\Delta \Rightarrow \frac{kn}{\Delta + k} \leq |D|.$$

Isso implica que

$$\gamma_k(G) \geq \frac{k}{\Delta + k}n. \quad (3.3)$$

É possível obter igualdade na expressão (3.3) em um grafo d -regular G para cada $d = \Delta(G)$. Com efeito, seja $G = (V, E)$ um grafo bipartido com bipartição $V(G) = C_1 \cup C_2$ onde $|C_1| = 2k$ e $|C_2| = 2d$ e onde cada vértice de C_1 possui grau d e cada vértice de C_2 possui grau k . Isso é possível porque o número de arestas com uma extremidade em C_1 é dk , que é o mesmo número de arestas com uma extremidade em C_2 . Para que G seja um grafo d -regular resta que cada vértice de C_2 possua mais $d - k$ arestas incidentes. Mas isso pode ser obtido pela adição das arestas de um grafo $(d - k)$ -regular $G' = (C_2, E')$. Se esse grafo $(d - k)$ -regular existir, C_1 será um conjunto com k vértices e k -dominante de um grafo d -regular

G que satisfaz $|V(G)| = d + k$. Isso fornece a igualdade desejada na expressão (3.3).

Resta provar que o grafo G' usado existe, o que segue da seguinte proposição.

Proposição 3.5. *Sejam $0 < d < n$ com dn par. Então existe um grafo d -regular de n vértices.*

Demonstração. Se d é par, defina $G = (V, E)$ onde $V = \mathbb{Z}_n = \{0, 1, 2, \dots, n-1\}$ é o corpo dos inteiros módulo n , e $\{n_1, n_2\} \in E(G)$ se, e somente se,

$$\min\{n_1 - n_2, n_2 - n_1\} \equiv 1, 2, 3, \dots, d/2 \pmod{(n)}.$$

Se d é ímpar, então n é par. Neste caso, construa o $G = (V, E)$ para o número par $d - 1$ e adicione a aresta $\{n_1, n_2\}$ sempre que

$$\min\{n_1 - n_2, n_2 - n_1\} \equiv n/2 \pmod{(n)}.$$

Esse passo fornece um emparelhamento adicional que completa a d -regularidade desejada. \square

Com respeito a cotas superiores para γ_k , um resultado de Hansberg e Volkman (2009) [27] estabelece cotas em função do grau mínimo $\delta(G)$ de um grafo G .

Teorema 3.6 (Corolário 1 de [27]). *Seja G é um grafo com n vértices e grau mínimo δ e seja $k \in \mathbb{N}$. Se $\frac{\delta + 1}{\ln(\delta + 1)} \geq 2k$, então*

$$\gamma_k(G) \leq \frac{k \ln(\delta + 1) + 1}{\delta + 1} n. \quad (3.4)$$

Quando $k = 2$, a desigualdade (3.4) é válida para todo $\delta \geq 9$. Mas sua conclusão só começa a ser útil a partir de $\delta \geq 13$, onde temos

$$\gamma_2(G) \leq \frac{2 \ln(13 + 1) + 1}{13 + 1} n \cong 0,4829n,$$

pois, para $\delta \leq 12$ e $k = 2$, há um resultado mais antigo (1990), devido a Caro e Roditty [15], que estabelece que, se $\delta(G) \geq 3$, então $\gamma_2(G) \leq n/2$. Essas cotas são justas para δ igual a 3 e 4, e os grafos que as justificam podem ser, por exemplo, os grafos bipartidos completos $K_{3,3}$ e $K_{4,4}$, respectivamente. Finalmente, há um resultado de 2016, devido a Bujtás e Jaskó [14], que melhora as cotas superiores anteriores para $6 \leq \delta \leq 21$. O resultado estabelece, por exemplo, que

$$\delta(G) = 6 \Rightarrow \gamma_2(G) \leq 0.498n,$$

$$\delta(G) = 7 \Rightarrow \gamma_2(G) \leq 0.467n,$$

$$\delta(G) = 8 \Rightarrow \gamma_2(G) \leq 0.441n \quad \text{e}$$

$$\delta(G) = 9 \Rightarrow \gamma_2(G) \leq 0.418n.$$

Atualmente, a nossa pesquisa bibliográfica sugere que as melhores cotas superiores para $\gamma_2(G)$ em função de $\delta(G)$ são dadas por Caro e Roditty para $3 \leq \delta \leq 5$ e por Bujtás e Jaskó para $6 \leq \delta \leq 21$.

Por outro lado, é natural que os grafos que têm um maior número de 2-dominância sejam aqueles que têm menos arestas. Assim, no conjunto dos grafos G com $\delta(G) = d$, é natural esperar que os grafos d -regulares sejam aqueles que possuem os maiores valores para γ_2 . Aparentemente, as melhores cotas para grafos d -regulares existentes na literatura são aquelas que são para o grau mínimo $\delta(G) = d$. De fato, a relação entre grau mínimo e várias noções de dominância já foi bem explorada.

3.4 Resultados principais

Nossos resultados fornecem cotas superiores assintóticas para os números de 2-dominância e de dominância total em grafos regulares. Basicamente, apresenta-

mos dois tipos de resultado: um deles, de caráter probabilístico, fornece essas cotas para a grande maioria dos grafos d -regulares, em função de d ; o outro, de caráter determinístico, fornece essas cotas para todos os grafos d -regulares com cintura suficientemente grande, também em função de d . Formalmente, os resultados relacionados a dominância total são dados pelos Teoremas 3.7 e 3.8, que correspondem às versões probabilística e determinística, respectivamente. Alguns dos enunciados dependem de soluções de sistemas de equações diferenciais que serão enunciados nos próximos capítulos. No teorema a seguir, pense em $q(x)$ como uma função que descreve a proporção do tamanho de um conjunto dominante parcial contruído por um certo algoritmo até um certo instante x e pense em x^* como o instante final do algoritmo.

Teorema 3.7. *Dados $\epsilon > 0$ e $d \geq 3$, um grafo aleatório d -regular G , com n vértices, assintoticamente quase certamente possui um conjunto totalmente dominante $D_T \subseteq V(G)$ tal que*

$$|D_T| \leq n(q(x^*) + \epsilon),$$

onde $x^* = \inf\{x > 0 : z_0(x) = 0\}$ e onde $z_0(x)$ e $q(x)$ são as soluções do problema de valor inicial (5.3).

Teorema 3.8. *Dados $\epsilon > 0$ e $d \geq 3$, existe um inteiro positivo g tal que todo grafo d -regular G com n vértices e cintura maior do que g possui um conjunto totalmente dominante D_T tal que*

$$|D_T| \leq (q(x^*) + \epsilon)n,$$

onde $x^* = \inf\{x > 0 : z_0(x) = 0\}$ e onde $z_0(x)$ e $q(x)$ são as soluções do problema de valor inicial (5.3).

Temos uma tabela comparando nossa cota probabilística para conjuntos totalmente dominantes $q(x^*)$ com as cotas determinísticas encontradas na literatura.

tura. O número $q(x^*)$ foi aproximado por uma implementação computacional via o *método de Euler* para resolução de sistemas de equações diferenciais, com tamanho do passo igual a 0,00005. As informações sobre as publicações foram obtidas do livro de Henning e Yeo [31] de 2013.

Grau mínimo $\delta(G)$	Cota inferior (trivial)	Cota superior já publicada	Publicação ou fonte	d	Aproximação para $q(x^*)$
3	$1/3 \simeq 0,3333$	$1/2 = 0,5000$	[4] (2004)	3	0,4883
4	$1/4 = 0,2500$	$3/7 \simeq 0,4285$	[50] (2007)	4	0,4136
5	$1/5 = 0,2000$	$17/44 \simeq 0,3863$	[20] (2015)	5	0,3656
6	$1/6 \simeq 0,1666$	0,4653	Teorema 3.2	6	0,3256

Tabela 3.2: Tabela comparando nossas cotas com as melhores cotas conhecidas para grafos em geral.

O Teorema 3.7 será provado no Capítulo 5 e uma justificativa para a validade do Teorema 3.8 será fornecida na Seção 5.6.

No caso da 2-dominância, apresentamos dois resultados independentes. O primeiro admite uma solução analítica para seu problema de valor inicial, o que leva a cotas explícitas. Isso permite que seus enunciados não dependam de equações diferenciais, como pode ser visto abaixo. O segundo resultado, que fornece cotas melhores, não admite uma solução desse tipo por ser obtido de um sistema de equações diferenciais mais complicado. Isso exige uma abordagem alternativa, como a aproximação computacional que usamos aqui. Cada um deles admite uma versão probabilística e uma determinística, similarmente à dominância total. As duas versões do primeiro resultado são os Teoremas 3.9 e 3.10 abaixo, cujas provas estão na Seção 6.

Teorema 3.9. *Para quaisquer $\epsilon > 0$ e $d \geq 4$, um grafo aleatório d -regular $G = (V, E)$ com n vértices assintoticamente quase certamente possui um conjunto 2-dominante $D \subseteq V(G)$, tal que*

$$|D| \leq n(x_2 + \epsilon),$$

onde

$$x_2 = \frac{d - 1 + u_d - d [(d - 2)(d - 1)(1 - \sqrt{u_d})]^{-2/(d-1)}}{2(d - 1)}$$

com $u_d := (d - 1)^{-2/(d-2)}$.

d	melhor cota conhecida	x_2
4	0,5	0,4519
5	0,5	0,4089
6	0,498	0,3747
7	0,467	0,3467
8	0,441	0,3233

Tabela 3.3: Comparação entre as melhores cotas conhecidas e as aproximações para o valor de x_2 em função de d no Teorema 3.9.

Como feito para a dominância total, a prova desse teorema terá como consequência um resultado determinístico para grafos regulares com cintura grande. Ele segue após duas aplicações sucessivas (uma para cada fase) do Teorema 5.8. O enunciado está abaixo e a justificativa é análoga à do Teorema 3.8.

Teorema 3.10. *Dado $\epsilon > 0$, existe um inteiro positivo g tal que todo grafo G d -regular com n vértices e cintura maior do que g possui um conjunto 2-dominante D tal que*

$$|D| \leq n(x_2 + \epsilon),$$

onde

$$x_2 = \frac{d-1 + u_d - d[(d-2)(d-1)(1 - \sqrt{u_d})]^{-2/(d-3)}}{2(d-1)}$$

com $u_d := (d-1)^{-2/(d-2)}$.

O resultado que trata das cotas superiores para o número de 2-dominância em grafos d -regulares com cotas melhores do que as do Teorema 3.9 é mais elaborado e seu enunciado depende de terminologias que serão definidas no Capítulo 8. Por exemplo, o Algoritmo 8(c), as matrizes M_k , os conjuntos Ω_{M_k} e as condições $(i.k)$ e $(1.k)$ estão definidas nas Seções 8.4 e 8.5.

Teorema 3.11. *Dados $d \geq 3$ e $\epsilon > 0$, seja $p_0 > 0$ tal que*

$$(1.1) \quad |\det M_1(z(0))| > \gamma,$$

$$(2.1) \quad |\alpha_i^1(0)| > \gamma \text{ para todo } 1 \leq i \leq d-1,$$

$$(3.1) \quad (0, z(0), q(0)) \in \Omega_1,$$

para algum $\gamma > 0$. Suponha que, para $k = 2, \dots, r$,

$$(a.k) \quad \det M_k(z(x_{k-1})) \neq 0, \text{ isto é, } (1.k) \text{ é satisfeita;}$$

$$(b.k) \quad (x_{k-1}, z(x_{k-1}), q(x_{k-1})) \in \Omega_{M_{k-1}}, \text{ isto é, } (i.k-1) \text{ não é satisfeita;}$$

Seja $y_r = x_1 + x_2 + \dots + x_r$. Então a cardinalidade do conjunto 2-dominante produzido pelo Algoritmo 8(c) aplicado a um grafo aleatório d -regular é a.q.c. no

máximo

$$y_r n + (1 - q(x_r))n + \epsilon n.$$

Teorema 3.12. *Dados $d \geq 3$ e $\epsilon > 0$, seja $p_0 > 0$ tal que*

$$(1.1) \quad |\det M_1(z(0))| > \gamma,$$

$$(2.1) \quad |\alpha_i^1(0)| > \gamma \text{ para todo } 1 \leq i \leq d-1,$$

$$(3.1) \quad (0, z(0), q(0)) \in \Omega_1,$$

para algum $\gamma > 0$. Suponha que, para $k = 2, \dots, r$,

$$(a.k) \quad \det M_k(z(x_{k-1})) \neq 0, \text{ isto é, } (1.k) \text{ é satisfeita;}$$

$$(b.k) \quad (x_{k-1}, z(x_{k-1}), q(x_{k-1})) \in \Omega_{M_{k-1}}, \text{ isto é, } (i.k-1) \text{ não é satisfeita;}$$

Seja $y_r = x_1 + x_2 + \dots + x_r$. Então existe um inteiro positivo g tal que cardinalidade do conjunto 2-dominante produzido pelo Algoritmo 8(c) aplicado a um grafo G d -regular com n vértices e cintura maior do que g é no máximo

$$y_r n + (1 - q(x_r))n + \epsilon n.$$

A coluna da direita da Tabela 3.4 apresenta algumas aproximações obtidas computacionalmente para as cotas dos Teoremas 3.11 e 3.12.

d	melhor cota conhecida	x_2	$y_r + 1 - q(x_r)$ (aproximação computacional)
3	0,5	—	0,4771
4	0,5	0,4519	0,4261
5	0,5	0,4089	0,3859
6	0,498	0,3747	0,3536

Tabela 3.4: Comparação entre as melhores cotas conhecidas para grafos em geral e as aproximações computacionais para as cotas obtidas nesta tese para 2-dominância em função de d

O Algoritmo 8(c) foi o que obteve melhor desempenho dentre vários algoritmos testados. O Teorema 3.11 fornece uma cota melhor do que a cota determinística para $d = 3$, o que não acontecia com o Teorema 3.9.

3.4.1 Uma relação entre cintura e aleatoriedade em grafos d -regulares

Podemos fornecer um resultado complementar aos resultados obtidos nesta tese, o Teorema 3.14, que relaciona a cintura e aleatoriedade em grafos d -regulares. Este resultado não depende de algoritmos locais. Por ser um pouco mais curto e independente dos demais, será provado integralmente nesta seção.

Relembre que, na introdução, apresentamos nossos objetivos, que consistiam em encontrar as melhores cotas superiores $c_1(d)$ e $c_2(d)$ que satisfizessem os Resultados 1 e 2, respectivamente. Nós também mencionamos que uma cota para o Resultado 2 dá uma cota para o Resultado 1. Trataremos deste fato agora. Por isso, é importante definir rigorosamente as melhores cotas superiores possíveis para os Resultados 1 e 2.

Dados $d \geq 2$ e $g_0 \geq 3$, seja

$$\gamma_t^g(d, g_0) := \sup\{\gamma_t(G)/|V(G)| : G \text{ é } d\text{-regular com cintura } g \geq g_0\}, \quad (3.5)$$

isto é, $\gamma_t^g(d, g_0)$ é a menor cota superior possível para grafos d -regulares com cintura pelo menos g_0 . Isto produz uma sequência monótona não-crescente quando g_0 cresce, e podemos considerar o parâmetro

$$\gamma_t^g(d, \infty) = \lim_{g_0 \rightarrow \infty} \gamma_t^g(d, g_0). \quad (3.6)$$

Esse é o ínfimo dos valores possíveis para $c_2(d)$ para o Resultado 2.

O resultado de Henning dado pela equação (3.2) combinado à cota inferior dada pela equação (3.3) mostra que $\gamma_t^g(2, \infty) = 1/2$.

Cabe mencionar que alguns de nossos principais resultados que envolvem cintura, que foram descritos na Seção 3.4, podem ser reescritos em função da notação dada por 3.6. Por exemplo, o Teorema 3.8, assumiria a seguinte forma.

Teorema 3.13. *Para $6 \geq d \geq 3$, temos que $\gamma_t^g(d, \infty)$ é no máximo os valores das aproximações para $q(x^*)$ da Tabela 3.2.*

Agora, podemos considerar o valor típico do número de dominação total em um grafo d -regular grande. Relembrando, $\mathbb{G}_{n,d}$ é o conjunto de grafos d -regulares (rotulados) com n vértices e, para um inteiro $d \geq 2$ e uma constante $\varepsilon > 0$, considere

$$\gamma_t^R(d, \varepsilon) = \lim_{n \rightarrow \infty} \inf_{\substack{\mathcal{A} \subseteq \mathbb{G}_{n,d}, n \in \mathbb{N}, \\ |\mathcal{A}| \geq (1-\varepsilon)|\mathbb{G}_{n,d}|}} \sup \left\{ \frac{\gamma_t(G)}{n} : G \in \mathcal{A} \right\}. \quad (3.7)$$

Note que, para d fixo, $\gamma_t^R(d, \varepsilon)$ é limitado e cresce quando ε decresce, assim o seguinte limite está bem definido:

$$\gamma_t^R(d) = \lim_{\varepsilon \rightarrow 0^+} \gamma_t^R(d, \varepsilon). \quad (3.8)$$

Em termos de probabilidade, encontrar uma cota superior $c_1(d)$ para $\gamma_t^R(d)$ significa que um grafo aleatório d -regular *assintoticamente quase certamente* possui um conjunto totalmente dominante mínimo com tamanho de no máximo $c_1(d)$. Portanto, $\gamma_t^R(d)$ é o menor $c_1(d)$ possível para o Resultado 1.

De forma análoga, poderemos reescrever os resultados que envolvam aleatoriedade da Seção 3.4 usando a notação $\gamma_t^R(d)$.

Uma construção bem conhecida permite-nos provar o seguinte.

Teorema 3.14. *É verdade que*

$$\gamma_t^R(d) \leq \gamma_t^g(d, \infty). \quad (3.9)$$

A conexão entre parâmetros de grafos com cintura grande e grafos aleatórios regulares apresentada por (3.9) no contexto de dominação total também é válida para vários parâmetros diferentes, e é uma questão em aberto significativa quando (3.9) vale com igualdade. Como mencionamos antes, Wormald e Hoppen [36] provaram que um limite superior para $\gamma_t^R(d)$ implica em um limite superior para $\gamma_t^g(d, \infty)$ se ele for obtido através da análise de um algoritmo local, como o que tratamos na Seção 1.3. (Novamente, a implicação anterior seria válida para uma série de parâmetros além da dominação total). Isso tem a interessante consequência de que a desigualdade (3.9) se tornaria uma igualdade caso os algoritmos locais tivessem a capacidade de aproximar $\gamma_t^R(d)$.

Em particular, a desigualdade (3.2) implica no fato simples de que a cota inferior trivial para a dominância total é assintoticamente justa para grafos 2-regulares quando $g \rightarrow \infty$ e, por causa do Teorema 3.14, implica que $\gamma_t^R(2)$ é igual a $n/2$, o que não parece tão trivial. É uma questão relevante se uma propriedade assintótica da mesma natureza também vale para grafos d -regulares com $d \geq 3$.

A prova do Teorema 3.14 pode ser conferida a seguir e versões análogas deste teorema para outros tipos de dominância podem ser facilmente obtidas por pequenas alterações na prova apresentada aqui.

Relembrando da Definição 2.6, $X_{n,d}^{g_0}(G)$ é o número de ciclos de G com comprimento menor do que g_0 e $A_{n,d}(g_0) := \{G \in \mathbb{G}_{n,d} : X_{n,d}^{g_0}(G) < \ln(n)\}$, isto é, $A_{n,d}(g_0)$ é o conjunto dos grafos de $\mathbb{G}_{n,d}$ com menos de $\ln(n)$ ciclos de comprimento menor do que g_0 .

Lema 3.15. *Para todo $G \in A_{n,d}(g_0)$, temos*

$$\frac{\gamma_t(G)}{n} \leq \gamma_t^g(d, g_0) + \frac{2 \ln(n)}{n}.$$

Demonstração. Seja $G \in A_{n,d}(g_0)$. Pela definição de $A_{n,d}(g_0)$, G tem menos de $\ln(n)$ ciclos de comprimento menor do que g_0 . Após remover uma aresta de cada ciclo curto formamos um grafo G' com cintura pelo menos g_0 e grau máximo d .

Afirmamos que existe um inteiro $s > 0$ tal que, usando s cópias de G' e acrescentando algumas arestas, é possível construir um grafo H , d -regular, com cintura pelo menos g_0 . Com efeito, o Corolário 2.8 garante que existe m par tal que, para todo $0 \leq i \leq d - 1$, existe um grafo $(d - i)$ -regular de m vértices e cintura pelo menos g_0 . Considere um grafo H_0 formado pelos grafos de uma família $\mathcal{G} = \{G_1, G_2, \dots, G_m\}$ de m cópias de G' . Se um vértice $v \in G'$ tem grau i , então há uma cópia desse vértice em cada elemento de \mathcal{G} , que, portanto, se repete m vezes em H_0 . Sobre esses m vértices, construa um grafo G_v $(d - i)$ -regular qualquer, com cintura pelo menos g_0 , e adicione suas arestas em qualquer ordem à H_0 sobre as cópias de v formando H_1 . Agora, repetimos esse procedimento para cada vértice de grau menor do que d , obtendo uma sequência H_2, H_3, \dots de grafos com grau mínimo menor ou igual a d . Para algum inteiro t teremos que H_t será d -regular. Basta definir $H = H_t$ para obter o grafo desejado.

Por definição de $\gamma_t^g(d, g_0)$, H possui um conjunto totalmente dominante D de tamanho no máximo $\gamma_t^g(d, g_0) \cdot ns$ vértices. Pelo Princípio da Casa dos Pombos, pelo menos uma das cópias de G' , digamos G_1 , possui uma quantidade menor ou igual a $\gamma_t^g(d, g_0) \cdot n$ de elementos de D .

No máximo $2 \ln(n)$ vértices de G_1 só são adjacentes a elementos de D fora de G_1 . Portanto, obtemos um conjunto totalmente dominante de G formado pelos $\gamma_t^g(d, g_0) \cdot n$ elementos de D que estavam em G_1 mais um vizinho de cada um dos até $2 \ln(n)$ vértices que não eram dominados por vértices de G_1 .

Como G_1 é isomorfo a G , a conclusão é que conseguimos construir um conjunto totalmente dominante de G com tamanho $\gamma_t^g(d, g_0) \cdot n + 2 \ln(n)$, isto é,

$$\frac{\gamma_t(G)}{n} \leq \gamma_t^g(d, g_0) + \frac{2 \ln(n)}{n}.$$

□

Demonstração. (Teorema 3.14) Mostraremos que, para todo $\delta > 0$, $\gamma_t^R(d) \leq \gamma_t^g(d, \infty) + \delta$.

Fixe $\delta > 0$. Seja g_0 tal que $\gamma_t^g(d, g_0) < \gamma_t^g(d, \infty) + \delta/3$ e $\epsilon > 0$ tal que $\gamma_t^R(d) < \gamma_t^R(d, \epsilon) + \delta/3$. O Corolário 2.7 garante que existe n_0 tal que, para todo $n \geq n_0$,

$$|A_{n,d}(g_0)| > (1 - \epsilon)|\mathbb{G}_{n,d}|.$$

Isso implica que, pela definição de $\gamma_t^R(d, \epsilon)$,

$$\gamma_t^R(d, \epsilon) \leq \sup \left\{ \frac{\gamma_t(G)}{n}; G \in A_{n,d}(g_0) \right\},$$

para todo $n \geq n_0$.

Por outro, o Lema 3.15 afirma que, para todo $G \in A_{n,d}(g_0)$,

$$\frac{\gamma_t(G)}{n} \leq \gamma_t^g(d, g_0) + \frac{2 \ln(n)}{n},$$

Tome $n_1 \geq n_0$ tal que, $2 \ln(n)/n < \delta/3$, para todo $n \geq n_1$. Assim, se $n \geq n_1$, temos

$$\frac{\gamma_t(G)}{n} \leq \gamma_t^g(d, g_0) + \frac{\delta}{3} \leq \gamma_t^g(d, \infty) + \frac{2\delta}{3}.$$

Finalmente, temos

$$\gamma_t^R(d) < \gamma_t^R(d, \varepsilon) + \delta/3 \leq \sup \left\{ \frac{\gamma_t(G)}{n}; G \in A_{n,d}(g_0) \right\} + \frac{\delta}{3} \leq \gamma_t^g(d, \infty) + \frac{2\delta}{3} + \frac{\delta}{3}.$$

concluindo a prova. □

4 O METODO DAS EQUAÇÕES DIFERENCIAIS

Vamos estudar processos aleatórios em grafos, que, no nosso caso, serão grafos que são modificados ao longo do processo. A cada passo, controlaremos o valor de certas variáveis, por exemplo, o número de vértices de cada grau. Considerando apenas valores esperados, isso nos leva a um sistema de recorrências para esses valores.

Para entender o processo aleatório, nosso objetivo será resolver essas recorrências e demonstrar que as suas soluções são próximas do que realmente acontece no processo, com alta probabilidade. A fim de fazer essa aproximação, vamos usar um teorema técnico que, resumidamente, afirma que as soluções assintóticas das recorrências são bem aproximadas por soluções de um sistema de equações diferenciais. Ele funciona como um resultado de concentração que aproxima o desempenho esperado do desempenho típico do processo.

O objetivo do capítulo é enunciar esse teorema e ilustrar como ele pode ser aplicado. A versão que apresentaremos está baseada no Teorema 5.1 em [55] e, aqui, seu enunciado segue como Teorema 4.2. Optamos por enunciá-lo sem prova pois a teoria necessária à sua compreensão é extensa, o que desviaria demais o nosso foco. O enunciado deste teorema aborda espaços de probabilidade e construções sob um ponto de vista mais geral.

4.1 Condição de Lipschitz

Definição 4.1. Dizemos que a função $f : D \rightarrow \mathbb{R}$ satisfaz a condição de Lipschitz em $D \subseteq \mathbb{R}^n$ se existe uma constante $L > 0$ com a propriedade de que

$$|f(u_1, u_2, \dots, u_n) - f(v_1, v_2, \dots, v_n)| \leq L \max_{1 \leq i \leq n} |u_i - v_i|$$

para todo (u_1, \dots, u_n) e (v_1, \dots, v_n) em D .

Neste caso dizemos que L é uma *constante Lipschitz* de f . Note que $\max_{1 \leq i \leq n} |u_i - v_i|$ é a distância entre (u_1, u_2, \dots, u_n) e (v_1, v_2, \dots, v_n) na métrica ℓ^∞ .

Vamos definir também a *norma* $\|v\|$ de $v = (v_1, \dots, v_n) \in \mathbb{R}^n$ por

$$\|v\| = \sqrt{v_1^2 + \dots + v_n^2}.$$

Note que a condição de Lipschitz é equivalente a

$$|f(u_1, u_2, \dots, u_n) - f(v_1, v_2, \dots, v_n)| \leq L \|u - v\|$$

para alguma constante L , pois

$$\max_{1 \leq i \leq n} |u_i - v_i| \leq \|u - v\| \leq \sqrt{n} \max_{1 \leq i \leq n} |u_i - v_i|.$$

4.2 O teorema da aproximação

Antes de enunciar o Teorema 4.2, vamos definir algumas estruturas um pouco mais gerais do que aquelas tratadas anteriormente, para fins de formalização.

Um *processo aleatório* é um espaço de probabilidade em tempo discreto que pode ser denotado por (Q_0, Q_1, \dots) , onde cada Q_i toma valores num conjunto S . Os elementos desse espaço são as seqüências (q_0, q_1, \dots) , onde cada $q_i \in S$.

Usamos h_t para denotar (q_0, q_1, \dots, q_t) , que denominamos a *história do processo* até o tempo t .

Consideraremos uma sequência de processos aleatórios indexados por n , com $n \in \mathbb{N}$. Assim $q_t = q_t^{(n)}$ e $S = S^{(n)}$, mas por simplicidade vamos omitir a dependência em n da notação. Note que o nosso “processo de grafos aleatórios” é um caso particular desse “processo aleatório”, se definirmos $S = S^{(n)} = \{G : G \text{ é algum grafo de } n \text{ vértices}\}$ e $q_t = G_t$.

A notação $S^{(n)+}$ denota o conjunto de todos $h_t = (q_0, \dots, q_t)$ onde $t \in \mathbb{N}$ e cada $q_i \in S^{(n)}$. No nosso caso, h_t é uma lista de grafos de n vértices que descreve a evolução da construção de um grafo até o tempo t .

Para variáveis Y_1, \dots, Y_a definidas sobre as componentes do processo, e $D \subseteq \mathbb{R}^{a+1}$, definimos o *tempo de parada* $T_D(Y_1, \dots, Y_a)$ como o menor t tal que

$$(t/n, Y_1(t)/n, \dots, Y_a(t)/n) \notin D.$$

Escreveremos apenas T_D quando as variáveis Y_1, \dots, Y_a estiverem claras no contexto. Em nossos exemplos, vamos aplicar o teorema no caso particular em que $Y_i(t)$ é o número de vértices de grau i no grafo q_t . Consideraremos também uma variável $Q(t)$, que fará a atualização da contagem do número de elementos no conjunto totalmente dominante em cada passo.

Escrevemos $a_n = O(b_n)$ se existe $n_0 \in \mathbb{N}$ e uma constante $C > 0$ tal que $|a_n| \leq Cb_n$, para todo $n > n_0$.

O enunciado do teorema exige três hipóteses. A primeira (i) assegura que cada Y_i não varia demais em cada passo, a segunda (ii) nos diz a taxa esperada dessa variação e estabelece que essa taxa basicamente depende apenas do conhecimento dos valores das variáveis Y_i no passo anterior, e a terceira (iii) assegura

que essa taxa não varia demais com o tempo. Com isso já podemos enunciar o teorema.

Teorema 4.2. *Para cada $1 \leq \ell \leq a$ onde a é fixo, sejam $y_\ell : S^{(n)+} \rightarrow \mathbb{R}$ e $f_\ell : \mathbb{R}^{a+1} \rightarrow \mathbb{R}$, tais que para alguma constante C_0 e todo ℓ , $|y_\ell(h_t)| < C_0 n$ para todo $h_t \in S^{(n)+}$ e para todo n . Seja $Y_\ell(t)$ a variável aleatória dada por $y_\ell(h_t)$. Suponha que as três condições a seguir sejam verdadeiras, onde, em (ii) e (iii), D é algum conjunto aberto, conexo e limitado contendo o fecho de*

$$\{(0, z_1, \dots, z_a) : \mathbb{P}[Y_\ell(0) = z_\ell n, 1 \leq \ell \leq a] \neq 0 \text{ para todo } n\}.$$

(i) *(Hipótese de limitação.) Para funções $\beta = \beta(n) \geq 1$ e $\gamma = \gamma(n)$, a probabilidade de que*

$$\max_{1 \leq \ell \leq a} |Y_\ell(t+1) - Y_\ell(t)| \leq \beta,$$

condicionada a qualquer história h_t , é pelo menos $1 - \gamma$ para $t < T_D$.

(ii) *(Hipótese de tendência.) Para alguma função $\lambda_1 = \lambda_1(n) = o(1)$, para todo $\ell \leq a$,*

$$|\mathbb{E}[Y_\ell(t+1) - Y_\ell(t) | h_t] - f_\ell(t/n, Y_1(t)/n, \dots, Y_a(t)/n)| \leq \lambda_1$$

para $t \leq T_D$.

(iii) *(Hipótese de Lipschitz.) Cada função f_ℓ é contínua e satisfaz a condição de Lipschitz em*

$$D \cap \{(t, z_1, \dots, z_a) : t \geq 0\}.$$

Então o seguinte é verdade.

(a) Para $(0, \hat{z}_1, \dots, \hat{z}_a) \in D$, o sistema de equações diferenciais

$$\frac{dz_\ell}{dx} = f_\ell(x, z_1, \dots, z_a) \quad \ell = 1, \dots, a$$

tem uma única solução em D para $z_\ell : \mathbb{R} \rightarrow \mathbb{R}$ passando por

$$z_\ell(0) = \hat{z}_\ell,$$

$1 \leq \ell \leq a$, que se estende a pontos arbitrariamente próximos da fronteira de D ;

(b) Seja $\lambda > \lambda_1 + C_0 n \gamma$ com $\lambda = o(1)$. Para uma constante C suficientemente grande, com probabilidade $1 - O\left(n\gamma + \frac{\beta}{\lambda} \exp\left(-\frac{n\lambda^3}{\beta^3}\right)\right)$,

$$Y_\ell(t) = n z_\ell(t/n) + O(\lambda n)$$

para $0 \leq t \leq \sigma n$ e para cada ℓ , onde $z_\ell(x)$ é a solução em (a) com $\hat{z}_\ell = Y_\ell(0)/n$, e $\sigma = \sigma(n)$ é o supremo dos x para os quais a solução pode ser estendida antes de alcançar distância ℓ^∞ igual a $C\lambda$ da fronteira de D .

Nos últimos anos, o Método das Equações Diferenciais tem sido intensamente aplicado a problemas de otimização de parâmetros em grafos regulares aleatórios. Restringindo a discussão apenas a parâmetros de dominância, temos, por exemplo o trabalho de Duckworth e Wormald [22], que obteve melhorias para o número de *dominação independente*. Duckworth e Mans [21] aplicaram o método à *r*-dominação, que é uma generalização de dominância que considera dominados os vértices a uma distância de até *r* passos de algum vértice do conjunto *r*-dominante.

Também tivemos aplicações recentes deste método feitas por Amini [2] a processos de percolação e difusão em grafos aleatórios com sequências de graus dadas e, posteriormente, por Amini, Cont e Minca [3] sobre a resiliência ao contágio

de redes financeiras. Ainda, na mesma linha dos grafos aleatórios com sequências de graus dadas, Brightwell, Janson e Łuczak [13] estudaram o tamanho de um conjunto independente produzido por um algoritmo simples.

Outra aplicação recente foi, como já mencionamos, a de Fiz Pontiveros, Griffiths e Morris [43] que obteve uma melhora assintótica significativa para o limite inferior do número de Ramsey $R(3, t)$ estabelecido anteriormente por Kim [40]. Enquanto Kim obteve que

$$R(3, t) \geq \left(\frac{1}{162} - o(1) \right) \frac{t^2}{\log(t)},$$

Fiz Pontiveros, Griffiths e Morris obtiveram que

$$R(3, t) \geq \left(\frac{1}{4} - o(1) \right) \frac{t^2}{\log(t)}.$$

Essa mesma melhora foi obtida independentemente por Bohman e Keevash [9].

Este teorema é a principal ferramenta que usaremos para obter nossos resultados. Por isso, na próxima seção, daremos um exemplo típico de aplicação com finalidade ilustrativa. Este exemplo será retomado no Capítulo 6.

4.3 Exemplo de aplicação do Teorema 4.2

Dado um grafo $G = (V, E)$, um conjunto $I \subseteq V(G)$ é dito *independente* se, para cada par $u, v \in I$, for verdade que $\{u, v\} \notin E(G)$, isto é, não existem arestas ligando vértices de I . Encontrar um conjunto independente de tamanho máximo é um problema clássico da teoria dos grafos que já foi amplamente abordado na literatura. Em particular, trata-se de um problema *NP*-completo. A seguir, vamos aplicar o teorema a fim de obter uma cota inferior, assintoticamente quase certa, para um conjunto independente máximo de um grafo aleatório d -regular. O que

faremos a seguir foi extraído de Wormald [55] e, conforme o autor, não é a melhor cota já descoberta. A finalidade é apenas ilustrar uma aplicação mais simples do que aquela que faremos no Capítulo 5. Vamos exibir alguns detalhes adicionais além dos já apresentados em [55].

Queremos construir um grafo aleatório d -regular de n vértices e um conjunto independente nesse grafo. Uma maneira de gerar esse grafo é por meio de um processo que gera uma configuração, escolhida como o já descrito na Seção 2.3. Partimos de n células, que representam os vértices do multigrafo, com d pontos em cada e, repetidamente, combinamos um ponto ainda não emparelhado, escolhido segundo algum critério, a um segundo ponto escolhido uniformemente dentre os pontos disponíveis, até que tenhamos um emparelhamento completo. No final, teremos um multigrafo aleatório. Lembre que, caso o emparelhamento seja grafo, então é escolhido de maneira uniforme. O interessante é que construir um emparelhamento aleatório é algo bem flexível, já que a ordem em que os pontos são tomados para formar seus pares não afeta a uniformidade probabilística do emparelhamento obtido no final.

Vamos estabelecer um algoritmo que construa um conjunto independente simultaneamente à construção do grafo. O conjunto contruído não precisa satisfazer alguma propriedade específica se o multigrafo construído não for grafo, mas, se o for, deve ser um conjunto independente.

Começamos em $t = 0$ sem pontos emparelhados, assim todos os vértices começam com grau zero. Em cada passo, que vai de $t - 1$ a t , fazemos o seguinte. Escolhemos um vértice dentre os vértices de grau zero uniformemente, e então pareamos seus pontos aleatoriamente e uniformemente. Deste modo, no final do passo, esse vértice escolhido terá grau d . Fim do passo!

É importante notar que o conjunto dos t vértices escolhidos até o tempo t é independente, pois cada vez que um passo é iniciado, selecionamos um vértice de grau zero que não está ligado aos vértices que foram escolhidos nos passos anteriores, já que todos os que estão ligados aos escolhidos antes têm grau maior do que zero. Esse passo é repetido até que não se tenha mais nenhum vértice de grau zero. O conjunto independente é o conjunto dos vértices escolhidos, sendo um em cada passo. O tamanho do nosso conjunto independente será exatamente igual ao t do último passo executado.

Formalmente, vamos denotar por P_t o emparelhamento parcial atualizado no tempo t . Vamos definir $Y(t)$ como o número de vértices de grau zero em P_t . Começamos com P_0 sendo um conjunto de nd pontos dispostos em n vértices (células), sem nenhum emparelhamento formado e, portanto, com $Y(0) = n$. Durante o processo, queremos determinar o primeiro t tal que $Y(t) = 0$.

O número de pontos em vértices de grau zero no tempo t é $dY(t)$, e existem $nd - 2dt$ pontos não emparelhado no total (visto que d pares são formado em cada unidade de tempo). A probabilidade de que um ponto qualquer esteja em um vértice de grau zero no tempo t é

$$\frac{dY(t)}{nd - 2dt} = \frac{Y(t)}{n - 2t}.$$

Essa probabilidade muda um pouco durante o passo, quando o novo vértice de grau zero é escolhido e o primeiro ponto dele é destacado (e a aresta deste ponto está prestes a ser exposta). Quando isso acontece, $Y(t)$ imediatamente decresce em 1 unidade, e o outro ponto da aresta é escolhido aleatoriamente dentre os $nd - 2dt - 1$ pontos restantes. Contudo, para cada uma das d arestas expostas num passo, a probabilidade de que pareie um ponto do vértice escolhido a um

ponto de um vértice de grau zero é de

$$\frac{Y(t)}{n-2t} + O\left(\frac{1}{n-2t-2}\right),$$

pois $2t+2 < n$, e assim

$$\mathbb{E}[Y(t+1) - Y(t)|P_t] = -1 - \frac{dY(t)}{n-2t} + O\left(\frac{1}{n-2t-2}\right).$$

Escrevendo $t = xn$ e $Y(t) = nz(t/n) = nz(x)$, a recorrência se torna

$$\begin{aligned} \frac{z(x+1/n) - z(x)}{1/n} &= \mathbb{E}[nz((xn+1)/n) - nz(x)|P_t] \\ &= -1 - \frac{dnz(x)}{n-2xn} + O\left(\frac{1}{n-2xn-2}\right). \end{aligned}$$

Como estamos supondo que n seja grande, isso sugere a equação diferencial

$$z'(x) = -1 - \frac{dz(x)}{1-2x}$$

com a condição inicial $z(0) = 1$. Esta é uma equação de primeira ordem cuja solução é

$$z(x) = \frac{(d-1)(1-2x)^{d/2} - (1-2x)}{d-2}.$$

A menor solução positiva de $z(x) = 0$ é

$$x^* = \frac{1}{2} - \frac{1}{2} \left(\frac{1}{d-1}\right)^{2/(d-2)}.$$

Podemos usar o teorema anterior para provar que x^*n é uma aproximação do tamanho do nosso conjunto independente se n for suficientemente grande. Com efeito, podemos verificar as hipóteses do teorema. Dado algum $\theta > 0$ pequeno, defina $D_\theta = \{(x, z) : -\theta < x < x^*, 0 < z < 1 + \theta\}$. A hipótese (i) se verifica trivialmente com $\beta(n) := d+1$ e $\gamma(n) := 0$, por exemplo. Para verificar (ii), note que $x^* < 1/2$. Daí, se $0 \leq x \leq x^*$,

$$O\left(\frac{1}{n-2t-2}\right) = O\left(\frac{1}{n}\right)$$

$$\Rightarrow \mathbb{E}[Y(t+1) - Y(t)|P_t] = -1 - \frac{dY(t)}{n-2t} + O\left(\frac{1}{n}\right).$$

Como

$$f(x, z) = -1 - \frac{dz}{1-2x},$$

temos que

$$\begin{aligned} & |\mathbb{E}[Y(t+1) - Y(t)|P_t] - f(t/n, Y(t)/n)| \\ &= -1 - \frac{dY(t)}{n-2t} + O\left(\frac{1}{n}\right) - \left(-1 - \frac{dY(t)/n}{1-2t/n}\right) \\ &= -1 - \frac{dY(t)}{n-2t} + O\left(\frac{1}{n}\right) - \left(-1 - \frac{dY(t)}{n-2t}\right) \\ &\leq \frac{A-1}{n} \end{aligned}$$

para algum $A > 0$. Basta tomar $\lambda_1(n) = (A-1)/n$ para obter (ii). Para verificar (iii), basta notar que $f(x, z)$ é uma função racional e que $x^* < 1/2 \Rightarrow 1-2x > 1-2x^* > 0$. Isso implica que $f(x, z)$ é Lipschitz em D_θ , para cada $\theta > 0$.

Com as hipóteses verificadas, vamos à conclusão. Note que $\lambda(n) = A/n^{1/4} > \lambda_1(n)$ satisfaz (b) e implica que a probabilidade descrita nesse item tenda a zero quando $n \rightarrow \infty$. A conclusão, com esse λ , é que, com probabilidade

$$1 - O\left(n^{1/4} \exp\left(-\frac{A^3}{(d+1)^3} n^{1/4}\right)\right),$$

vale que

$$Y(t) = nz(t/n) + O(\lambda n),$$

para $0 \leq t \leq \sigma n$, onde $\sigma = \sigma(n)$ é o supremo dos x para os quais a solução $z(x)$ pode ser estendida antes de alcançar a distância $C\lambda$ das laterais do retângulo aberto D_θ , para alguma constante $C > 0$. Em particular, dado $\epsilon > 0$, temos que $x^* - \sigma(n) < \epsilon$ desde que n seja suficientemente grande. Isso segue da continuidade e monotonicidade de $z(x)$ nas proximidades de x^* e também porque $C\lambda(n) \rightarrow 0$

quando $n \rightarrow \infty$. Como $|I| = \sigma(n)n$, então $|I| > (x^* - \epsilon)n$. Daí, provamos o seguinte.

Teorema 4.3. *Dado $\epsilon > 0$, um grafo aleatório d -regular G , com n vértices, assintoticamente quase certamente possui um conjunto independente $I \subseteq V(G)$, tal que*

$$|I| \geq n(x^* - \epsilon),$$

onde

$$x^* = \frac{1}{2} - \frac{1}{2} \left(\frac{1}{d-1} \right)^{2/(d-2)}.$$

Este é apenas um exemplo cuja finalidade é a de ilustrar uma aplicação do teorema. Como já mencionamos, há algoritmos melhores do que esse para conjuntos independentes que usam o Teorema 4.2. Por exemplo, em Wormald [55] temos um outro algoritmo que produz um conjunto independente maior do que este. Diferentemente do nosso algoritmo, que encerra logo que não tenha mais vértices de grau zero, o algoritmo encontrado em [55] é executado até completar o multigrafo.

O Teorema 4.2 permite a escolha do algoritmo. Porém, nas aplicações realizadas até o presente momento, as decisões tomadas pelo algoritmo em cada passo devem ser tomadas apenas em função das informações obtidas das proximidades do vértice escolhido no passo atual. Como mencionamos, algoritmos desse tipo são conhecidos como *algoritmos locais*.

5 DOMINÂNCIA TOTAL EM GRAFOS REGULARES

Neste capítulo, provaremos os Teoremas 3.7 e 3.8 usando o Método das Equações Diferenciais, que foi apresentado no Capítulo 4. Nossa abordagem faz uma análise de um algoritmo local capaz de construir conjuntos totalmente dominantes de maneira eficiente. As equações obtidas não foram resolvidas analiticamente, no entanto apresentamos aproximações computacionais que justificam a eficiência do seu desempenho.

5.1 Uma heurística para produzir um conjunto totalmente dominante pequeno

Nesta seção, definiremos formalmente o algoritmo usado e as variáveis que deverão ser analisadas pelo nosso método. Seguiremos passos similares aos do exemplo da Seção 4.3 do Capítulo 4. Como antes, usaremos o modelo de emparelhamentos descrito no Capítulo 2 e a independência que existe entre os pares para simular a construção de grafos aleatórios d -regulares.

Assim como na Seção 4.3 do Capítulo 4, para cada $0 \leq i \leq d - 1$, $Y_i(t)$ é o número de vértices de grau i em G_t .

O algoritmo depende de parâmetros fixos d , n e $\epsilon > 0$ e tem como entrada um grafo H d -regular com n vértices. Sua evolução é controlada por um parâmetro discreto de tempo t , onde $t = 0, 1, 2, \dots$. Definimos também um conjunto $D(t)$ que inicia com $D(0) = \emptyset$. Em cada instante t o seguinte procedimento é realizado. Escolhemos um vértice v_t de H_t de grau máximo d com probabilidade uniforme e

revelamos um de seus vizinhos u_t arbitrariamente. Se esse vizinho possuir grau d , adicionamos v_t e u_t em $D(t)$ e removemos v_t e u_t de H_t , produzindo H_{t+1} . Se o grau de u_t for menor do que d então adicionamos apenas u_t a $D(t)$ e removemos u_t de H_t , produzindo H_{t+1} . O processo termina no primeiro instante t no qual a quantidade de vértices de grau d estiver abaixo de um limiar $n\epsilon$. Para produzir o conjunto totalmente dominante final, adicionamos a $D(t)$ um vizinho de cada vértices de grau d restante. Chamamos de *etapa de seleção* a escolha do vértice inicial a cada passo, que neste caso é a escolha de v_t , e chamamos de *regra de exploração local* e *passo de recoloração* a escolha das adjacências investigadas e dos vértices revelados durante o passo, bem como das decisões que são tomadas a partir das informações obtidas. Em particular, a etapa de seleção corresponde precisamente à linha 5 do Algoritmo 4.1(a) e à linha 4 do Algoritmo 4.1(b) e a regra de exploração local corresponde às linhas 6 do Algoritmo 4.1(a) e 5 do Algoritmo 4.1(b) e o passo de recoloração corresponde aos fragmentos entre as linhas 7 e 14 do Algoritmo 4.1(a) e entre as linhas 6 e 13 do Algoritmo 4.1(b).

Algoritmo 4.1(a): com parâmetros $n \geq 4$, $d \geq 3$ e $\epsilon > 0$.

Entrada: Um grafo d -regular H com n vértices.

Saída: Um conjunto totalmente dominante D_T do grafo H .

```
1  $t = 0$ ;  
2  $H_0 = H$ ;  
3  $D(0) = \emptyset$ ;  
4 enquanto  $Y_d(t) > n\epsilon$  faça  
5     Escolha um vértice  $v_t$  de grau  $d$  com probabilidade uniforme;  
6     Revele um vizinho  $u_t$  de  $v_t$ ;  
7     se  $\deg_{G_t}(u_t) = d$  então  
8          $D(t+1) = D(t) \cup \{u_t, v_t\}$ ;  
9         Remova  $u_t$  e  $v_t$  de  $H_t$ , produzindo  $H_{t+1}$ ;  
10    senão  
11         $D(t+1) = D(t) \cup \{u_t\}$ ;  
12        Remova  $u_t$  de  $H_t$ , produzindo  $H_{t+1}$ ;  
13    fim  
14     $t = t + 1$ ;  
15 fim  
16  $D_T = D(t) \cup \{\text{um vizinho de cada vértice de grau } d \text{ de } H_t\}$ .
```

Na prática, esse algoritmo poderia ser estendido até que não restassem mais vértices de grau zero, isto é, até que $Y_d(t) = 0$. Porém, nesse caso, o algoritmo atingiria uma região que nosso método não é capaz de analisar. Por isso, definimos que nosso algoritmo termina um pouco antes, quando $Y_d(t) > n\epsilon$, para algum $\epsilon > 0$ pequeno.

Para analisar esse algoritmo, simulamos seu desempenho em um multigrafo aleatório regular. Pela propriedade de independência, isso pode ser feito gerando o grafo de acordo com as escolhas tomadas pelo algoritmo.

Assim, definiremos um algoritmo (Algoritmo 4.1(b) abaixo, que chamaremos de algoritmo de construção) que ilustra como o Algoritmo 4.1(a) (chamado de

algoritmo de deleção) pode ser aplicado durante o próprio processo de construção do multigrafo d -regular aleatório.

O Algoritmo 4.1(b) parte em $t = 0$ de uma estrutura que denotaremos por G_0 , que consiste num conjunto de nd pontos organizados em n células com d pontos em cada célula, sem nenhum ponto emparelhado. Como antes, as células se referem a vértices do multigrafo e os pares darão origem às arestas. Assim, dizer que um vértice é emparelhado é o mesmo que dizer que os pontos da célula que representa esse vértice, e que ainda não foram emparelhados, são emparelhados. Os multigrafos H_t e G_t representam essencialmente a mesma estrutura a menos da forma como olhamos para o grafo. Enquanto que o Algoritmo 4.1(a) inicia com um grafo d -regular bem definido, no Algoritmo 4.1(b), o grafo é construído à medida que aplicamos o algoritmo. Assim, quando um vértice ou uma aresta são removidos pelo algoritmo de deleção, eles são, na verdade, emparelhados ou construídos pelo algoritmo de construção. Assim, H_t denota o que ainda não foi removido pelo algoritmo, enquanto que G_t denota o que ainda não foi emparelhado na geração do grafo aleatório de forma que, do ponto de vista da geração do multigrafo aleatório d -regular, as adjacências ainda são desconhecidas. Como consequência, o papel de cada variável Y_i da versão de construção é trocado pelo papel da variável Y_{d-i} na versão de deleção.

O Algoritmo 4.1(b) pode ser adaptado facilmente ao nosso método de análise pois usa o *modelo dos emparelhamentos* discutido no Capítulo 2 para construir o grafo aleatório regular.

Algoritmo 4.1(b): com parâmetros $n \geq 4$, $d \geq 3$ e $\epsilon > 0$.

Entrada: A estrutura G_0 definida acima.

Saída: Um multigrafo aleatório d -regular G com n vértices e um conjunto totalmente dominante D_T de G .

1 $t = 0$;

2 $D(0) = \emptyset$;

3 **enquanto** $Y_0(t) > n\epsilon$ **faça**

4 Escolha um vértice v_t de grau zero com probabilidade uniforme;

5 Revele um vizinho u_t de v_t ;

6 **se** $\deg_{G_t}(u_t) = 0$ **então**

7 $D(t+1) = D(t) \cup \{u_t, v_t\}$;

8 Emparelhe u_t e v_t para obter G_{t+1} ;

9 **senão**

10 $D(t+1) = D(t) \cup \{u_t\}$;

11 Emparelhe u_t para obter G_{t+1} ;

12 **fim**

13 $t = t + 1$;

14 **fim**

15 $D_T = D(t) \cup \{\text{um vizinho de cada vértice de grau 0 de } G_t\}$;

16 Complete o emparelhamento de G_t para obter G .

Note que é possível que o primeiro par, que determina u_t , forme um laço, que, neste caso, liga um vértice de G_t de grau zero a si mesmo. Pela lei do algoritmo teremos $v_t = u_t$, que é acrescentado a D . O método se aplica ao modelo de emparelhamentos e, portanto, analisa o comportamento do algoritmo para multigrafos. Assim, laços em princípio poderiam afetar os resultados. Isso não ocorre por motivos relacionados aos cálculos das esperanças condicionais, conforme veremos. Grosso modo, ocorrências desse tipo são tão raras que não afetam as equações além dos erros permitidos pelo método.

Note que os passos eficientes do algoritmo são executados até o último t tal que $Y_0(t) > n\epsilon$. A partir desse ponto, nada mais é acrescentado a D e os pontos que não foram emparelhados na geração do emparelhamento podem ser emparelhados de forma arbitrária, pois são irrelevantes para D . Se o multigrafo final for um grafo, então $D(t)$ domina todos os vértices de grau maior ou igual a 1 no instante t . Cada um dos poucos vértices que ainda não foram dominados são dominados por um vizinho acrescentado ao conjunto D final, conforme feito no último passo, para formar um conjunto totalmente dominante D_T . Podemos definir a variável $Q(t)$ como o número de vértices em $D(t)$. Ela contabilizará o número de elementos do conjunto totalmente dominante que está sendo produzido em cada passo do processo, e é sempre não-decrescente com t .

Queremos aplicar o Teorema 4.2. Por isso, devemos verificar que suas hipóteses são satisfeitas. Também queremos identificar o instante mais avançado no qual o teorema dá suporte ao nosso algoritmo. Isso é o que faremos a seguir.

5.2 A mudança esperada em cada passo

Com o Algoritmo 4.1(b) bem definido, podemos fazer algumas deduções sobre as mudanças esperadas nas variáveis aleatórias em um passo do processo, isso será necessário para a análise do Método das Equações Diferenciais.

Defina $S(t)$ como o número total de pontos de G_t não emparelhados, assim

$$S(t) = \sum_{m=0}^{d-1} (d-m)Y_m(t).$$

Quando um ponto é selecionado com probabilidade uniforme para formar par com um ponto de v_t , ele pertence a um vértice u_t de G_t de algum grau, que é menor ou igual a $d-1$, com uma certa probabilidade. Definimos a variável $X(t)$ como

o grau do vértice u_t definido na linha 4 do Algoritmo 4.1(b). Desse modo, $X(t)$ determina quantos (e, com a escolha de u_t , quais) pontos ainda serão emparelhados nesse mesmo passo. Como $Y_0(t) < S(t)$, vale que

$$\mathbb{P}[X(t) = i | G_t] = \frac{(d-i)Y_i(t) - 1_{\{i=0\}}}{S(t)} = \frac{(d-i)Y_i(t)}{S(t)} + O\left(\frac{1}{Y_0(t)}\right).$$

Vamos considerar que t está num intervalo fixo $0 \leq t \leq t'$ onde $t' = \sup\{s : Y_0(t) \geq n\epsilon, \forall t \in [0, s]\}$ para algum $\epsilon > 0$. Neste caso, vale que

$$\frac{1}{Y_0(t)} \leq \frac{1}{n\epsilon} = O\left(\frac{1}{\sqrt{n}}\right),$$

e a última expressão pode ser reescrita como

$$\frac{(d-i)Y_i(t)}{S(t)} + O\left(\frac{1}{\sqrt{n}}\right). \quad (5.1)$$

Note que, em cada passo, apenas um quantidade limitada de pontos ($\leq 4d-2$) é emparelhada, o número exato depende de $X(t)$. Se $X(t) \geq 1$, então outros $d - X(t) - 1$ pares serão formados ainda nesse passo, se $X(t) = 0$, então outros $2d - 2$ pares serão formados. Vamos definir os seguintes eventos

$$A_t^k(j) := \{\text{o } k\text{-ésimo par do passo } t \text{ atinge um vértice de grau } j\}.$$

Pode-se provar, de modo similar à expressão anterior, que

$$\mathbb{P}[A_t^k(j) | G_t] = \frac{(d-j)Y_j(t)}{S(t)} + O\left(\frac{1}{\sqrt{n}}\right). \quad (5.2)$$

Sabemos que o valor de $Y_j(t+1) - Y_j(t)$ é igual a diferença entre o número de vértices que passam a ter grau j e aqueles que deixam de ter grau j no passo $t+1$. Assim, a escolha do segundo ponto em cada par pode contribuir em uma unidade (a mais ou a menos) para essa diferença. Note também que a diferença total é igual a soma dessas contribuições. Vamos definir que $A_t^k(-1) = \emptyset$, para todo k , e vamos fixar $Y_{-1}(t) \equiv 0$.

A função Delta de Dirac, denotada por $\delta_{i,j}$, será igual a 1 se $i = j$ e igual a 0 caso contrário.

Daí, condicionado a $X(t) = 0$, temos que, para todo $0 \leq j \leq d - 1$,

$$\begin{aligned} \mathbb{E}[Y_j(t+1) - Y_j(t) | X(t) = 0, G_t] &= -2\delta_{0,j} + \mathbb{E} \left[\sum_{k=2}^{2d-1} (1_{A_t^k(j-1)} - 1_{A_t^k(j)}) | G_t \right] \\ &= -2\delta_{0,j} + \sum_{k=2}^{2d-1} (\mathbb{E}[1_{A_t^k(j-1)} | G_t] - \mathbb{E}[1_{A_t^k(j)} | G_t]) \\ &= -2\delta_{0,j} + \sum_{k=2}^{2d-1} (\mathbb{P}[A_t^k(j-1) | G_t] - \mathbb{P}[A_t^k(j) | G_t]), \end{aligned}$$

onde o termo $-2\delta_{0,j}$ é devido ao fato de que os vértices v_t e u_t têm grau zero, o que não é contabilizado no somatório. Usando a igualdade (5.2), a última expressão pode ser reescrita como

$$-2\delta_{0,j} + (2d - 2) \left(\frac{(d-j+1)Y_{j-1}(t)}{S(t)} - \frac{(d-j)Y_j(t)}{S(t)} \right) + O\left(\frac{1}{\sqrt{n}}\right).$$

Condicionado a $X(t) = i$, onde $1 \leq i \leq d - 1$, um argumento análogo ao anterior mostra que, para todo $0 \leq j \leq d - 1$, temos

$$\begin{aligned} \mathbb{E}[Y_j(t+1) - Y_j(t) | X(t) = i, G_t] \\ = -\delta_{i,j} - \delta_{0,j} + \delta_{1,j} + (d-i-1) \left(\frac{(d-j+1)Y_{j-1}(t)}{S(t)} - \frac{(d-j)Y_j(t)}{S(t)} \right) + O\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

Com isso, temos uma representação assintótica para $\mathbb{E}[Y_j(t+1) - Y_j(t) | X(t) = i, G_t]$ para cada i tal que $0 \leq i \leq d - 1$.

Como a distribuição de $X(t)$, condicionada a G_t , é dada pela expressão (5.1), podemos desenvolver a esperança da diferença $Y_j(t+1) - Y_j(t)$, condicionada somente a G_t , para todo $0 \leq j \leq d - 1$, como segue:

$$\begin{aligned}
\mathbb{E}[Y_j(t+1) - Y_j(t)|G_t] &= \mathbb{E}[\mathbb{E}[Y_j(t+1) - Y_j(t)|X(t) = i, G_t]|G_t] \\
&= \sum_{i=0}^{d-1} (\mathbb{E}[Y_j(t+1) - Y_j(t)|X(t) = i, G_t] \cdot \mathbb{P}[X(t) = i|G_t]) \\
&= \sum_{i=1}^{d-1} \frac{(d-i)Y_i(t)}{S(t)} \left[-\delta_{i,j} + \delta_{1,j} + (d-i-1) \left(\frac{(d-j+1)Y_{j-1}(t)}{S(t)} - \frac{(d-j)Y_j(t)}{S(t)} \right) \right] \\
&\quad -\delta_{0,j} + \frac{dY_0(t)}{S(t)} \left[-\delta_{0,j} + (2d-2) \left(\frac{(d-j+1)Y_{j-1}(t)}{S(t)} - \frac{(d-j)Y_j(t)}{S(t)} \right) \right] \\
&\quad + O\left(\frac{1}{\sqrt{n}}\right).
\end{aligned}$$

Estamos interessados no comportamento das variáveis $Y_j(t)$, com t variando entre 0 e t' , para valores grandes de n . Faremos agora uma etapa menos formal e mais ilustrativa. Consideraremos versões “normalizadas” de $Y_j(t)$, que, ao invés de fornecer a quantidade de vértices de grau j , fornecem a proporção de vértices, dentre um total de n , que têm grau j . A escala de tempo também deve ser mudada, já que, para valores maiores de n , mais passos serão dados até o final do processo. Como o número total de passos até o final desse processo deve ser algo entre $n/2d$ e n (pois cada passo elimina entre 1 e $2d$ vértices de grau zero), podemos considerar a mudança da escala de passos unitários para passos de comprimento $1/n$ definindo, por exemplo, $x = t/n$. Assim, x varia entre 0 até algo entre $1/2d$ e 1. Finalmente, vamos supor que existam funções reais diferenciáveis $z_j(x)$, para modelar o comportamento de $Y_j(xn)/n$, onde $xn = t$. Queremos que, para valores de n suficientemente grandes, as variáveis $Y_j(t)$ sejam aproximadas pelas funções $nz_j(t/n)$.

Vamos, agora, substituir todos $Y_j(t)$ na equação acima por $nz_j(x)$ e fazer $n \rightarrow \infty$. Também fixaremos $z_{-1}(x) \equiv 0$. Note que a esperança condicional desaparece, já que não temos mais variáveis aleatórias e sim funções reais. No

lado esquerdo teremos

$$\lim_{n \rightarrow \infty} \frac{z_j(x + 1/n) - z_j(x)}{1/n} = z'_j(x).$$

No lado direito, temos

$$\sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)} \left[-\delta_{i,j} + \delta_{1,j} + (d-i-1) \left(\frac{(d-j+1)z_{j-1}(x)}{s(x)} - \frac{(d-j)z_j(x)}{s(x)} \right) \right] \\ - \delta_{0,j} + \frac{dz_0(x)}{s(x)} \left[-\delta_{0,j} + (2d-2) \left(\frac{(d-j+1)z_{j-1}(x)}{s(x)} - \frac{(d-j)z_j(x)}{s(x)} \right) \right],$$

onde

$$s(x) = \sum_{m=0}^{d-1} (d-m)z_m(x).$$

Resta fazer a mesma substituição para a função $Q(t)$. Nesse caso, temos

$$\begin{aligned} \mathbb{E}[Q(t+1) - Q(t)|G_t] &= \mathbb{E}[1\{X_t > 0\} + 2 \cdot 1\{X_t = 0\}|G_t] \\ &= \mathbb{P}[X_t > 0|G_t] + 2\mathbb{P}[X_t = 0|G_t] \\ &= \frac{S(t) - dY_0(t)}{S(t)} + \frac{2dY_0(t)}{S(t)} + O\left(\frac{1}{\sqrt{n}}\right) \\ &= 1 + \frac{dY_0(t)}{S(t)} + O\left(\frac{1}{\sqrt{n}}\right). \end{aligned}$$

Agora, após fazermos a substituição $q(x) = Q(t)/n$ e $n \rightarrow \infty$, obtemos

$$q'(x) = 1 + \frac{dz_0(x)}{s(x)}.$$

Finalmente, para cada $0 \leq j \leq d-1$, definimos cada expressão $f_j(x, z_0, \dots, z_{d-1})$

por

$$\sum_{i=1}^{d-1} \frac{(d-i)z_i}{s} \left[-\delta_{i,j} + \delta_{1,j} + (d-i-1) \left(\frac{(d-j+1)z_{j-1}}{s} - \frac{(d-j)z_j}{s} \right) \right] \\ - \delta_{0,j} + \frac{dz_0}{s(x)} \left[-\delta_{0,j} + (2d-2) \left(\frac{(d-j+1)z_{j-1}}{s} - \frac{(d-j)z_j}{s} \right) \right],$$

Adicionalmente, definimos também

$$f_d(x, z_0, \dots, z_{d-1}) = 1 + \frac{dz_0}{s}.$$

Isso sugere o seguinte problema de valor inicial:

$$\begin{cases} z'_j(x) = f_j(x, z_0, z_1, \dots, z_{d-1}) & \text{para cada } 0 \leq j \leq d-1 \\ q'(x) = f_d(x, z_0, z_1, \dots, z_{d-1}) \\ z_0(0) = 1 \\ z_j(0) = 0 & \text{para cada } 1 \leq j \leq d-1 \\ q(0) = 0 \end{cases} \quad (5.3)$$

O Método das Equações Diferenciais garantirá que, para n suficientemente grande, as soluções $z_j(x)$ e $q(x)$ são, com alta probabilidade, boas aproximações para $Y_j(xn)/n$ e para $Q(xn)/n$, respectivamente, para todos os valores de x em $\{1/n, \dots, t'/n\}$. Além disso, veremos que o valor de t'/n está concentrado na vizinhança de um valor que conseguimos aproximar.

5.3 Algumas propriedades das soluções do sistema de equações diferenciais

Parece não haver uma forma simples de encontrar uma solução analítica explícita pra o PVI (5.3), o que nos leva a estabelecer propriedades qualitativas das soluções. Especificamente, provaremos que todas as soluções $z_j(x)$ e $q(x)$ tomam valores no intervalo $[0, 1]$ desde que x varie de zero até algum tempo suficientemente avançado, isto é, algum tempo x onde $z_0(x)$ está próximo de zero. Isso permitirá que o Teorema 5.1 de Wormald [55] seja aplicado em todo intervalo de interesse, garantindo que essas soluções descrevam o comportamento assintótico das variáveis

Y_j/n e $Q(t)/n$ até o final do processo. Assim, o número de vértices afetados pelo último passo do algoritmo será reduzido, de forma que o número de vértices afetado pelo último passo do algoritmo, que é ineficiente, seja desprezível..

Proposição 5.1. *Existe $\delta > 0$ tal que, para todo $x \in (0, \delta]$, e todo $0 \leq j \leq d-1$, vale que $z_j(x) > 0$.*

Demonstração. Como $z_0(0) = 1$ e $z_0(x)$ é diferenciável em $x = 0$ e, portanto, contínua nesse ponto, temos que existe $\delta_0 > 0$ tal que $z_0(x) > 0$ para todo $x \in [0, \delta_0]$.

Agora vamos considerar apenas as equações diferenciais dos z'_j tais que $1 \leq j \leq d-1$. Neste caso, podemos reescrever essas equações diferenciais assim:

$$\begin{aligned} z'_j(x) = & z_{j-1}(x)(d-j+1) \left((2d-2) \frac{dz_0(x)}{s(x)^2} + \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)^2} (d-i-1) \right) \\ & - z_j(x)(d-j) \left((2d-2) \frac{dz_0(x)}{s(x)^2} + \frac{1}{s(x)} + \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)^2} (d-i-1) \right) \\ & + \delta_{1,j} \cdot \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)} \end{aligned}$$

Para simplificar a escrita, vamos denotar

$$\begin{aligned} u_{1,j}(x) := & (d-j+1) \left((2d-2) \frac{dz_0(x)}{s(x)^2} + \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)^2} (d-i-1) \right) \quad \text{e} \\ u_{2,j}(x) := & (d-j) \left((2d-2) \frac{dz_0(x)}{s(x)^2} + \frac{1}{s(x)} + \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)^2} (d-i-1) \right). \end{aligned}$$

Assim o sistema pode ser reescrito como

$$z'_j(x) = z_{j-1}(x)u_{1,j}(x) - z_j(x)u_{2,j}(x) + \delta_{1,j} \cdot \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)}. \quad (5.4)$$

Isso permite demonstrar, por indução, que z_j é infinitamente diferenciável nos pontos em que $s(x) > 0$.

Para simplificar a notação da k -ésima derivada, escrevemos

$$f^{(k)}(x_0) := \left. \frac{d^k[f(x)]}{dx^k} \right|_{x=x_0}.$$

O que vamos fazer é provar que a derivada não-nula de ordem mais baixa de cada z_j em $x = 0$ é maior do que zero.

Lema 5.2. *Para cada $1 \leq j \leq d-1$, temos que $z_j^{(k)}(0) = 0$ para todo $1 \leq k \leq j-1$ e que $z_j^{(j)}(0) > 0$.*

Demonstração. Usaremos indução em $j \geq 1$. A base é dada pelo fato de que

$$z'_1(0) = z_0(0)u_{1,1}(0) - z_1(0)u_{2,1}(0) + \delta_{1,j} \cdot \sum_{i=1}^{d-1} \frac{(d-i)z_i(0)}{s(0)} = u_{1,1}(0) = 2(d-1) > 1.$$

já que, para todo $1 \leq j \leq d-1$,

$$u_{1,j}(0) = \frac{3(d-j+1)(d-1)}{d} > 0, \quad z_0(0) = 1 \quad \text{e} \quad z_1(0) = 0.$$

No passo de indução, vamos usar a regra geral de Leibniz, para n -ésima derivada do produto de funções na reta, e a hipótese de indução, a saber: $z_{j-1}(0) = z'_{j-1}(0) = z''_{j-1}(0) = \dots = z_{j-1}^{(j-2)}(0) = 0$ e $z_{j-1}^{(j-1)}(0) > 0$. Assim, podemos derivar $k-1$ vezes ambos os lados de (5.4). Daí obtemos que, para todo $1 \leq k \leq j-1$,

$$\begin{aligned} z_j^{(k)}(0) &= \sum_{m=0}^{k-1} \binom{k-1}{m} \left(z_{j-1}^{(m)}(0)u_{1,j}^{(k-m)}(0) - z_j^{(m)}(0)u_{2,j}^{(k-m)}(0) \right) \\ &= - \sum_{m=0}^{k-1} \binom{k-1}{m} z_j^{(m)}(0)u_{2,j}^{(k-m)}(0). \end{aligned}$$

A igualdade acima, com $k = 1$, implica que $z'_j(0) = 0$, pois $z_j(0) = 0$. Agora, como $z_j(0) = 0$ e $z'_j(0) = 0$, essa igualdade com $k = 2$ implicará que $z''_j(0) = 0$. Esse mesmo argumento pode ser repetido até obtermos $z_j(0) = z'_j(0) = z''_j(0) = \dots =$

$z_j^{(j-1)}(0) = 0$. Resta provar que $z_j^{(j)}(0) > 0$, mas isto é consequência de que

$$\begin{aligned} z_j^{(j)}(0) &= \sum_{m=0}^{j-1} \binom{j-1}{m} \left(z_{j-1}^{(m)}(0) u_{1,j}^{(j-m)}(0) - z_j^{(m)}(0) u_{2,j}^{(j-m)}(0) \right) \\ &= z_{j-1}^{(j-1)}(0) u_{1,j}(0) > 0. \end{aligned}$$

Isso conclui a indução. □

Com isso em mãos, podemos escrever a j -ésima Expansão de Taylor [41] de $z_j(x)$ em torno de $x = 0$, para cada j tal que $1 \leq j \leq d - 1$. Daí obtemos

$$z_j(x) = \frac{z_j^{(j)}(0)}{j!} x^j + r(x),$$

onde $r(x)$ cumpre

$$\lim_{x \rightarrow 0^+} r(x)/x^j = 0.$$

Assim, pela definição de limite de funções, existe $\delta_j > 0$ tal que, para todo $x \in (0, \delta_j]$, o lado direito de

$$z_j(x) = \left(\frac{z_j^{(j)}(0)}{j!} + \frac{r(x)}{x^j} \right) x^j$$

é positivo. Isto é, para todo $x \in (0, \delta_j]$ vale que $z_j(x) > 0$. Para concluir a prova, basta tomar $\delta = \min\{\delta_j : 0 \leq j \leq d - 1\}$. □

A próxima proposição vai garantir uma cota superior para $z_j(x)$ e $s(x)$ desde que certas condições sejam satisfeitas.

Proposição 5.3. *Se $z_j(x) \geq 0$, $z_0(x) > \epsilon_0$ para todo $x \in [0, \theta]$ e todo $j \in \{0, 1, \dots, d - 1\}$, onde $\epsilon_0 > 0$ e $\theta > 0$, então $s(x) \leq d$ e cada $z_j(x) \leq 1$ nesse intervalo.*

Demonstração. Seja $Z(x) = z_0(x) + z_1(x) + \cdots + z_{d-1}(x)$. Note que $Z'(x) = z'_0(x) + z'_1(x) + \cdots + z'_{d-1}(x)$, o que pode ser desenvolvido somando todas equações diferenciais em (5.3). Isso vai fornecer que, para todo $x \in [0, \theta]$,

$$Z'(x) = -1 - \frac{dz_0(x)}{s(x)} - \left[\frac{(d - (d - 1))z_{d-1}(x)}{s(x)} \right] \\ \times \left[(2d - 2) \frac{dz_0(x)}{s(x)} + \sum_{i=1}^{d-1} \frac{(d - i)z_i(x)}{s(x)} (d - i - 1) \right] \leq -1,$$

onde a desigualdade da direita segue do fato de que $s(x) \geq dz_0(x) \geq d\epsilon_0$. Isso prova que $Z(x)$ é decrescente no intervalo $[0, \theta]$. Agora, como $dZ(0) = dz_0(0) = d = s(0)$, e $dZ(x) \geq s(x)$, segue que $s(x) \leq d$.

Na segunda parte, para limitar $z_j(x)$, basta notar que, para todo $j \in \{0, 1, \dots, d-1\}$, vale que $1 \geq Z(x) \geq z_j(x)$ e que $Z(0) = 1 \geq z_j(0)$. \square

Sabemos que existe $\theta > 0$ satisfazendo as hipóteses da proposição acima, por exemplo $\theta = \delta$. Nas próximas proposições, sempre que falarmos em θ ele será tal que satisfaça as hipóteses da Proposição 5.3, sem reafirmar isto, a menos que haja ambiguidade no contexto.

Proposição 5.4. *Suponha que, para todo $x \in [\delta, \theta]$, todo $j \in \{0, 1, \dots, d-1\}$, algum $i \in \{0, 1, \dots, d-2\}$ e algum $\epsilon_i > 0$ tal que $\epsilon_i \leq \min\{z_j(\delta)/2 : 0 \leq j \leq d-1\} > 0$, tenhamos $z_j(x) \geq 0$ e $z_i(x) > \epsilon_i$. Então existe $\epsilon_{i+1} > 0$ tal que $z_{i+1}(x) > \epsilon_{i+1}$ em $[\delta, \theta]$.*

Demonstração. Usando as notações da demonstração da Proposição 5.1, temos que

$$z'_{i+1}(x) = z_i(x)u_{1,i+1}(x) - z_{i+1}(x)u_{2,i+1}(x) + \delta_{1,j} \cdot \sum_{i=1}^{d-1} \frac{(d-i)z_i(x)}{s(x)} \\ > \epsilon_i u_{1,i+1}(x) - z_{i+1}(x)u_{2,i+1}(x).$$

Note que $\epsilon_i u_{1,i+1}(x) - z_{i+1}(x) u_{2,i+1}(x)$ é maior do que zero sempre que

$$z_{i+1}(x) < \frac{\epsilon_i u_{1,i+1}(x)}{u_{2,i+1}(x)}, \quad (5.5)$$

Agora, como $s(x) \leq d$, $z_j \geq 0$ e $z_i > \epsilon_i$ com $0 \leq i \leq d-2$, a definição de $u_{1,j}$ fornece que

$$u_{1,i+1} \geq 2 \cdot \frac{2 \cdot \epsilon_i}{d^2} = \frac{4\epsilon_i}{d^2}.$$

Por outro lado, como $s(x) \geq \epsilon_i$ e $z_j \leq 1$, a definição de $u_{2,j}$ fornece que

$$u_{2,i+1} \leq d \cdot \frac{(2d \cdot d + d^3)}{\epsilon_i^2} \leq \frac{3d^4}{\epsilon_i^2}$$

Como consequência, o lado direito de (5.5) é maior do que o valor positivo $(\epsilon_i/d)^6$.

Portanto, é verdade a implicação

$$z_{i+1}(x) < (\epsilon_i/d)^6 \Rightarrow z'_{i+1}(x) > 0.$$

Agora, como $z_{i+1}(\delta) > \epsilon_i > (\epsilon_i/d)^6$ e como z_{i+1} é contínua, não tem como decrescer até alcançar, por exemplo, $(\epsilon_i/d)^6/2$ pois, quando $z_{i+1}(x)$ está entre $(\epsilon_i/d)^6$ e $(\epsilon_i/d)^6/2$, sua derivada deve ser positiva. Daí a prova segue com $\epsilon_{i+1} = (\epsilon_i/d)^6/2$. \square

Seja $\epsilon_0 > 0$ um número real arbitrário menor do que $\min\{z_j(\delta)/2 : 0 \leq j \leq d-1\}$. Para um θ adequado, podemos aplicar a Proposição 5.4 repetidamente de modo a construir uma sequência $\{\epsilon_0, \epsilon_1, \dots, \epsilon_{d-1}\}$ que cumpre $z_j(x) > \epsilon_j$ para todo $j \in \{0, 1, \dots, d-1\}$.

Vamos então definir $\theta_0 := \inf\{x > \delta : z_j(x) = \epsilon_j, \text{ algum } j\}$. Esse número está bem definido porque cada $z_j(\delta) > \epsilon_j$ e $z'_0(x) \leq -1$ em qualquer intervalo $[\delta, \theta]$ no qual todas as $z_j(x)$ sejam maiores ou iguais a zero. Afirmamos que $z_0(\theta_0) = \epsilon_0$. De fato, a continuidade das z_j garante que haveria algum $j \neq 0$ tal que $z_j(\theta_0) = \epsilon_j$.

Com isso, a Proposição 5.4 garante que $z_{j-1}(\theta_0) = \epsilon_{j-1}$, pois θ_0 é o supremo dos θ que satisfazem a proposição com o ϵ_0 dado (mas θ_0 não satisfaz porque $z_j(\theta_0) = \epsilon_j$). Pelo mesmo argumento, obtemos que $z_{j-2}(\theta_0) = \epsilon_{j-2}$ e assim por diante. Após j passos concluiremos que $z_0(\theta_0) = \epsilon_0$.

Esse argumento garante que nenhuma função $z_j(x)$, partindo de $z_j(\delta)$ alcance zero antes de $z_0(x)$. Assim, a construção acima prova o seguinte resultado.

Proposição 5.5. *Para todo $\epsilon_0 > 0$ menor do que $\min\{z_j(\delta_j)/2 : 0 \leq j \leq d-1\}$, existe $\theta_0 > \delta$ tal que $z_0(\theta_0) = \epsilon_0$ e, para todo $x \in [\delta, \theta_0]$ e $j \in \{0, 1, \dots, d-1\}$, $z_j(x) > 0$.*

Finalmente, como z_0 decresce até alcançar esse ϵ_0 , as Proposições 5.1, 5.3 e 5.5 permitem enunciar o seguinte resultado.

Proposição 5.6. *Considere o problema de valor inicial (5.3). Seja $\epsilon > 0$ fixo e $\theta_\epsilon := \inf\{x > 0 : z_0(x) = \epsilon\}$. Então, para todo $x \in [0, \theta_\epsilon]$ e $1 \leq j \leq d-1$, vale que $0 \leq z_j(x) \leq 1$, $0 \leq q(x) \leq 1$ e $\epsilon \leq z_0(x) \leq 1$.*

Demonstração. Resta apenas verificar que $0 \leq q(x) \leq 1$, já que o resto é consequência das proposições anteriores.

Com efeito, seja $\ell(x) = -z_0(x) + 1$. Note que $q(0) = \ell(0) = 0$ e, se $x \in [0, \theta_\epsilon]$, então $1 \leq 1 + dz_0(x)/s(x) = q'(x) \leq -z_0'(x) = \ell'(x)$. Assim $0 \leq q(x) \leq \ell(x) \leq 1 - \epsilon$. □

Uma consequência dessa proposição é que z_0 vai decrescer até alcançar zero pois, para todo $\epsilon > 0$, z_0 tem derivada menor do que -1 no intervalo $[0, \theta_\epsilon]$. Assim, fica bem definido $x^* = \min\{x > 0 : z_0(x) = 0\}$.

5.4 Verificando a condição de Lipschitz

Dado $\epsilon > 0$ e o vetor $\mathbf{z} = (z_0, z_1, \dots, z_{d-1})$, seja

$$D_\epsilon = \left\{ (x, q, \mathbf{z}) \in \mathbb{R}^{d+2} : -\epsilon < x < \theta_\epsilon, \epsilon < z_0 < 1 + \epsilon, -\epsilon/d < q, z_j < 1 + \epsilon \right\},$$

onde $j = 1, 2, \dots, d-1$.

Proposição 5.7. *Dado $\epsilon > 0$, cada f_j satisfaz a condição de Lipschitz em D_ϵ .*

Demonstração. Recorde que, se $0 \leq j \leq d-1$, $f_j(x, z_0, \dots, z_{d-1})$ é dada pela seguinte expressão

$$\sum_{i=1}^{d-1} \frac{(d-i)z_i}{s} \left[-\delta_{i,j} + \delta_{1,j} + (d-i-1) \left(\frac{(d-j+1)z_{j-1}}{s} - \frac{(d-j)z_j}{s} \right) \right] \\ - \delta_{0,j} + \frac{dz_0}{s(x)} \left[-\delta_{0,j} + (2d-2) \left(\frac{(d-j+1)z_{j-1}}{s} - \frac{(d-j)z_j}{s} \right) \right],$$

onde

$$s = \sum_{m=0}^{d-1} (d-m)z_m,$$

e, se $j = d$, é dada por

$$f_d(x, z_0, \dots, z_{d-1}) = 1 + \frac{dz_0}{s}.$$

Note que cada $f_j(z)$, onde $z = (x, z_0, \dots, z_{d-1})$, é uma função racional da forma $p(z)/s(z)^2$, onde $p(z)$ e $s(z)$ são funções polinomiais em $d+2$ variáveis. Note também que $s(z) \geq \epsilon$ em qualquer ponto de $\overline{D_\epsilon}$ (fecho de D_ϵ). Neste caso, é fácil concluir que cada f_j tem derivadas parciais de qualquer ordem em $\overline{D_\epsilon}$ e, portanto, cada f_j é uma função diferenciável com derivadas parciais contínuas em um conjunto compacto $\overline{D_\epsilon}$. Isso implica que cada derivada parcial deve ser limitada, isto é, para cada $0 \leq i \leq d$, existe $M_i > 0$ tal que $\left| \frac{\partial f_j}{\partial z_i}(z) \right| \leq M_i$, para todo $z \in \overline{D_\epsilon}$. Tomando $M = \sum_i M_i$ temos,

$$|f(u) - f(v)| \leq M \|u - v\|.$$

O último passo é consequência do Teorema do Valor Médio em \mathbb{R}^{d+2} . \square

5.5 Aplicando o Teorema 4.2

Vamos aplicar o Teorema 4.2 ao processo que definimos antes. Lembremos que cada variável Y_j denota o número de vértices de grau j no tempo t . Assim fixamos $a = d$ no enunciado do Teorema 4.2. Tomamos $C_0 = 1$, $D = D_{\epsilon/8}$, onde $\epsilon > 0$ é fixo. A hipótese de limitação segue com $\beta = 2d$, o que permite tomar $\gamma = 0$. Já provamos que $\lambda_1 = O(1/\sqrt{n})$. Assim, vamos tomar algum $A > 0$ suficientemente grande tal que $\lambda_1 \leq A/\sqrt{n}$. A hipótese de Lipschitz já foi verificada para as nossas funções f_j sobre esse conjunto D na seção anterior. A função $\lambda(n)$ do item (b), além de ser maior do que λ_1 , deve ser tal que a probabilidade do item (b) tenda a zero quando $n \rightarrow \infty$. Vamos escolher $\lambda(n) = A/\sqrt[4]{n}$ por motivos que ficarão claros mais a frente. Lembremos que $\sigma = \sigma(n)$ é o supremo dos x para os quais a solução pode ser estendida antes de alcançar a distância $C\lambda$ da fronteira de $D_{\epsilon/8}$, onde C é alguma constante suficientemente grande.

Como z_0 é contínua e $z_0(\theta_{\epsilon/8}) = \epsilon/8$, existe $\alpha < \theta_{\epsilon/8}$ tal que $z_0(x) \leq \epsilon/4$ para todo $x \in [\alpha, \theta_{\epsilon/8}]$. Por outro lado, existe n_0 tal que para todo $n > n_0$, $C\lambda(n) < \min\{\theta_{\epsilon/8} - \alpha, \epsilon/8d\}$. Assim, se $n > n_0$, a Proposição 5.6 garante que a primeira z_j a se aproximar da fronteira de $D_{\epsilon/8}$ a uma distância menor do que $C\lambda$, é z_0 . Além disso, para $n > n_0$, pela definição de α ,

$$\alpha \leq \sigma(n) \Rightarrow z_0(\sigma) \leq \frac{\epsilon}{4}$$

e como $\alpha < \theta_{\epsilon/8} - C\lambda$ e pela definição de σ ,

$$\alpha > \sigma(n) \Rightarrow z_0(\sigma) = \frac{\epsilon}{8} + C\lambda \leq \frac{\epsilon}{4}.$$

Consequentemente, para todo $n > n_0$, $z_0(\sigma) \leq \epsilon/4$. Primeiramente vamos supor que $T_D \geq n\sigma$. Assim, com a probabilidade descrita no item (b) do Teorema 4.2, $Q(\sigma n)$ é dada por

$$Q(\sigma n) = nq(\sigma) + O(\lambda n).$$

Recorde que $Q(t)$ faz a contagem de um conjunto $D(t)$ que, no final do processo deverá ser o conjunto totalmente dominante do grafo gerado durante o algoritmo. Note que $D(\sigma n)$ é totalmente dominante para todos os vértices de grau não nulo no tempo σn . Assim, para termos um conjunto totalmente dominante de todo grafo devemos adicionar vértices a esse conjunto a fim de dominar os vértices restantes (de grau zero). Para resolver isso, podemos, por exemplo, adicionar um vizinho de cada vértice de grau zero ao conjunto $D(\sigma n)$. O número de vértices de grau zero em σn , com a probabilidade descrita, é dado por

$$Y_0(\sigma n) = nz_0(\sigma) + O(\lambda n).$$

Assim, o tamanho do conjunto totalmente dominante construído, que denotamos por D_T , será

$$\begin{aligned} |D_T| &= Q(\sigma n) + Y_0(\sigma n) \\ &= nq(\sigma) + nz_0(\sigma) + O(\lambda n) \\ &\leq nq(x^*) + \frac{n\epsilon}{2} + O(\lambda n) \\ &= n \left(q(x^*) + \frac{\epsilon}{2} + O(\lambda) \right) \end{aligned}$$

onde a desigualdade segue pelo fato de que a função $q(x)$ é crescente. Podemos escolher n_1 tal que, se $n > n_1$, $O(\lambda(n)) \leq \epsilon/2$, e daí, para todo $n > \max\{n_0, n_1\}$, temos

$$|D_T| \leq n(q(x^*) + \epsilon).$$

O caso em que $T_D < \sigma n$ é mais simples. Neste caso $Y(T_D) \leq \epsilon/8$ e daí, como antes, adicionamos a D_T um vizinho de cada um desses poucos $Y_0(T_D)$ vértices de

grau zero e, da mesma forma, temos

$$|D_T| \leq nq(T_D) + \frac{2\epsilon n}{8} \leq n(q(x^*) + \epsilon).$$

Resta dizer com que probabilidade isso ocorre. Mas isso está na conclusão do item (b) e é função de n . Com as nossas escolhas de $\lambda(n) = A/n^{1/4}$, $\gamma = 0$ e $\beta = 2d$, a expressão da probabilidade torna-se

$$1 - O\left(n\gamma + \frac{\beta}{\lambda} \exp\left(-\frac{n\lambda^3}{\beta^3}\right)\right) = 1 - O\left(\frac{2d}{n^{1/4}} \exp\left(-\frac{An^{1/4}}{8d^3}\right)\right).$$

Esta última converge para 1 quando $n \rightarrow \infty$. Com isso, provamos o Teorema 3.7.

A fim de ilustrar as soluções do PVI (5.3), fornecemos abaixo um esboço computacional de suas curvas para o caso particular em que $d = 3$:

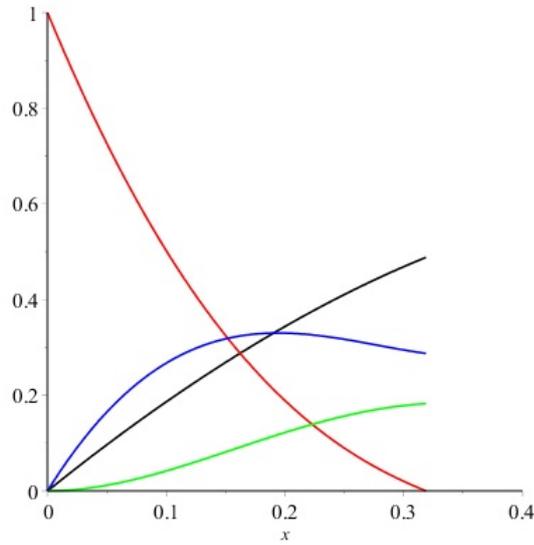


Figura 5.1: Gráfico das soluções do PVI no caso $d = 3$ com as curvas $z_0(x)$ em vermelho, $z_1(x)$ em azul, $z_2(x)$ em verde e $q(x)$ em preto.

5.6 Uma consequência da prova do Teorema 3.7

Conforme já mencionamos, há um trabalho recente, devido a Hoppen e Wormald [36], que leva a uma consequência interessante da prova do Teorema 3.7. De fato, o Teorema 5.5 desse artigo pode ser aplicado aos nossos dados permitindo uma consequência determinística cujo enunciado tem a estrutura do Resultado 2 visto no final da Seção 2.3. Ele usa as mesmas cotas do Teorema 3.7 e vale para todos os grafos d -regulares G com cintura $g(G)$ suficientemente grande. O Teorema 5.8 a seguir é uma consequência do Teorema 5.5 [36] que funciona como uma nova versão cujo objetivo é o de seguir a nossa terminologia.

Um *algoritmo local de deleção* é uma estrutura composta por uma *regra de seleção*, uma *regra de exploração local* e uma *função de recoloração*. Seja $C = \{c_1, \dots, c_k, c_{k+1}, \dots, c_\ell\}$ um conjunto de cores, onde as cores c_1, \dots, c_k são ditas cores *transientes* e c_{k+1}, \dots, c_ℓ são ditas *cores de saída*. Dado um grafo $G = (V, E)$ cujos vértices são coloridos com cores de C , o *tipo* de um vértice $v \in V(G)$ é um par (cor, grau) associado a esse vértice, onde a cor é a cor do próprio vértice e o grau é o número de vizinhos do vértice com cores transientes. *Identificar* uma adjacência é determinar o tipo de um vizinho do vértice original. *Revelar* um vizinho de um certo tipo é determinar qual dos vértices com tal tipo é o vizinho do vértice original. Atribuir uma cor de saída é uma maneira de implementar a remoção de um vértice por algoritmos que envolvam remoção de vértices.

A *regra de seleção* consiste em uma operação que produz um conjunto $S \subseteq V(G)$ de maneira aleatória com a propriedade de que vértices do mesmo tipo devem pertencer a S com a mesma probabilidade. Vértices com cores de saída não podem pertencer a S . A *granularidade* de um algoritmo é a probabilidade máxima de um vértice pertencer ao conjunto S da regra de seleção. Uma forma

natural de obter S é atribuir uma probabilidade para cada vértice pertencer a S de forma que vértices do mesmo tipo tenham a mesma probabilidade. Outra forma é fixar um tipo e escolher vértices desse tipo com probabilidade uniforme, neste caso S seria composto apenas por vértices de um mesmo tipo. Naturalmente, a probabilidade de que um vértice esteja em S é determinada pelo seu tipo. Nos algoritmos tratados nesta tese, S contém sempre um único vértice.

Uma *regra de exploração local* com raio r é uma sequência de operações de identificação e revelação com a propriedade de que só são identificados ou revelados vizinhos de vértices que já foram revelados e que estão a uma distância limitada por r de algum elemento de S .

A *função de recoloração* é uma operação executada após uma aplicação da regra de exploração local que permite alterar cores de vértices que já foram revelados pela regra de exploração local recém aplicada e de seus vizinhos identificados. Vértices que tiveram todos os seus vizinhos identificados recebem *cores de saída*, que são cores inalteráveis por futuras aplicações da função de recoloração. As decisões da função de recoloração só dependem da estrutura do grafo que foi identificada pela regra de exploração local, e portanto, que está a uma distância limitada por $2r$ do vértice a ser recolorido.

Um *passo* do algoritmo local de deleção é uma aplicação da regra de seleção seguida por uma aplicação da regra de exploração local a cada vértice selecionado e terminando com uma aplicação da função de recoloração aos vértices identificados ou revelados pela regra de exploração local. A execução do algoritmo local de deleção consiste na aplicação de uma sequência de passos desse tipo.

No Algoritmo 4.1(a) não atribuímos cores, o que equivale a assumir uma função de recoloração trivial, que possui somente uma cor transiente e uma cor de saída. Neste caso o algoritmo é denominado por *algoritmo nativo*.

No contexto de algoritmos locais de deleção apresentados aqui, o Algoritmo 4.1(a) pode ser interpretado da seguinte forma. O algoritmo tem uma única cor transiente e uma única cor de saída. Como só há uma cor transiente, o tipo de cada vértice é determinado por seu grau. O algoritmo tem como entrada um grafo d -regular com todos os vértices de cor transiente. A regra de seleção escolhe um vértice v_t com probabilidade uniforme entre os vértices de grau d , de forma que ela claramente satisfaz a propriedade desejada. A regra de exploração local inicialmente identifica o grau de um dos vizinhos de v_t , escolhido uniformemente dentre as d adjacências, e revela esse vizinho u_t . Se $\deg(u_t) = d$ então todas as adjacências de u_t e v_t são identificadas, senão, somente as adjacências de u_t são identificadas. Isso encerra a regra de exploração local, que tem raio $r = 1$. Finalmente, a função de recoloração entra em cena atribuindo cores de saída para os vértices revelados pela regra de exploração local. Nesse passo, as adjacências identificadas, mas não reveladas, no passo exploração, são reveladas para que tais vizinhos possam ser recoloridos. Nesse algoritmo, esse passo é inócuo, pois nenhum vértice receberá uma cor transiente distinta da original. Essa interpretação estabelece que o Algoritmo 5.1(a) é um algoritmo local de deleção.

As concentrações ρ desse enunciado são os valores das variáveis definidas no Teorema 4.2 no instante $n\sigma$ definido nas conclusões do teorema.

Teorema 5.8. *Considere $d \geq 1$, uma regra de seleção, uma regra de exploração local L e uma função de recoloração c . Suponha que o número de tipos seja igual a s e o número de cores de saída seja igual a s_0 . Sejam $Y_1(t), \dots, Y_{s+s_0}(t)$ variáveis que contam o número de vértices de cada tipo e vértices com cada cor de saída no*

instante t . Suponha que essas variáveis satisfazem as hipóteses do Teorema 4.2. Sejam $\rho_1, \dots, \rho_{s+s_0}$, as concentrações para os tamanhos das variáveis fornecidas pela conclusão do Teorema 4.2. Então, para todo $\delta > 0$, existe um algoritmo local de deleção \mathcal{A} com a mesma regra de exploração local L e função de recoloração c , e granularidade no máximo δ , tal que quando \mathcal{A} é aplicado em qualquer grafo d -regular de n vértices G com cintura suficientemente grande, existe um passo T no qual, para todo $1 \leq i \leq s + s_0$, o número esperado de vértices de cor de saída ou tipo i no grafo G_T , obtido após a aplicação do T -ésimo passo do algoritmo \mathcal{A} , está a distância menor ou igual a δn de $\rho_i n$.

No nosso contexto, simplesmente verificamos que o algoritmo que utilizamos para demonstrar o Teorema 3.7 está na classe de algoritmos descrita em Hoppen e Wormald [36] para obter o Teorema 3.8.

6 2-DOMINÂNCIA EM GRAFOS ALEATÓRIOS REGULARES: UM ALGORITMO SIMPLES

Queremos abordar o parâmetro γ_2 usando o Método das Equações Diferenciais a fim de obter cotas melhores do que as cotas determinísticas apresentada na Seção 3.3 que sejam válidas para a grande maioria dos grafos d -regulares para cada d , a menos de uma proporção insignificante.

6.1 Uma heurística para produzir um conjunto 2-dominante pequeno

Primeiramente, vamos estudar um algoritmo simples que permite encontrar cotas explícitas melhores do que as cotas determinísticas apresentadas na literatura. Notemos as semelhanças da versão de construção desse algoritmo com o exemplo apresentado no Capítulo 4: Em cada passo tomamos um vértice de grau zero e o acrescentamos a um conjunto D . Note que, durante todo o processo, o grau de cada vértice é igual ao número de vezes que ele foi dominado. Assim, quando os vértices de grau zero acabam, obtemos um conjunto de saída D que é dominante desse grafo. Como queremos um conjunto 2-dominante, vamos estender esse processo escolhendo, agora, vértices de grau 1 para pôr no conjunto dominante. Quando os vértices de grau 1 acabarem, não haverá mais vértices de grau 0 e 1, o que implica que todos os vértices são dominados pelo menos duas vezes. Portanto, D será 2-dominante. Vamos chamar de fase 0 o intervalo no qual tomamos vértices de grau zero e de fase 1 o intervalo no qual tomamos vértices de grau 1. Cada fase será estudada como um processo independente e vamos começar

com a fase 0. Formalmente, o enunciado formal desse algoritmo é originalmente dado pelo Algoritmo 5.1(a), que, como antes, é equivalente a um algoritmo de simulação que pode ser analisado pelo Método das Equações Diferenciais. Esse último será denotado por Algoritmo 5.1(b).

Algoritmo 5.1(a): com parâmetros $n \geq 4$, $d \geq 4$ e $\epsilon > 0$.	
	Entrada: Um grafo d -regular G com n vértices.
	Saída: Um conjunto 2-dominante D_2 do grafo G .
1	$t = 0$;
2	$H_0 = G$;
3	$D(0) = \emptyset$;
4	enquanto $Y_d(t) > n\epsilon$ faça
5	Escolha um vértice u_t de grau d com probabilidade uniforme;
6	Remova u_t de H_t , produzindo H_{t+1} ;
7	$D(t+1) = D(t) \cup \{u_t\}$;
8	$t = t + 1$;
9	fim
10	enquanto $Y_{d-1}(t) > n\epsilon$ faça
11	Escolha um vértice u_t de grau $d - 1$ com probabilidade uniforme;
12	Remova u_t de H_t , produzindo H_{t+1} ;
13	$D(t+1) = D(t) \cup \{u_t\}$;
14	$t = t + 1$;
15	fim
16	$D_2 = D(t) \cup \{\text{vértices de graus } d \text{ e } d - 1 \text{ de } H_t\}$.

Como feito no Capítulo 5, o Algoritmo 5.1(b) parte em $t = 0$ de uma estrutura que denotaremos por G_0 , que consiste num conjunto de nd pontos organizados em n células com d pontos em cada célula, sem nenhum ponto emparelhado. Como antes, as células serão referidas a vértices do multigrafo e os pares darão origem às arestas.

Algoritmo 5.1(b): com parâmetros $n \geq 4$, $d \geq 4$ e $\epsilon > 0$.

Entrada: A estrutura G_0 descrita anteriormente.

Saída: Um multigrafo aleatório d -regular G com n vértices e um conjunto 2-dominante D_2 de G .

```
1  $t = 0$ ;  
2  $D(0) = \emptyset$ ;  
3 enquanto  $Y_0(t) > n\epsilon$  faça  
4   Escolha um vértice  $u_t$  de grau 0 com probabilidade uniforme;  
5   Emparelhe  $u_t$  produzindo  $G_{t+1}$ ;  
6    $D(t+1) = D(t) \cup \{u_t\}$ ;  
7    $t = t + 1$ ;  
8 fim  
9 enquanto  $Y_1(t) > n\epsilon$  faça  
10  Escolha um vértice  $u_t$  de grau  $d - 1$  com probabilidade  
    uniforme;  
11  Emparelhe  $u_t$  produzindo  $G_{t+1}$ ;  
12   $D(t+1) = D(t) \cup \{u_t\}$ ;  
13   $t = t + 1$ ;  
14 fim  
15  $D_2 = D(t) \cup \{\text{vértices de graus 0 e 1 de } G_t\}$ .
```

6.2 Analisando o desempenho do Algoritmo 5.1(b)

Essa análise pode ser feita em duas fases onde a fase 0 corresponde ao intervalo que parte da linha 1 e vai até a linha 8 do Algoritmo 5.1(b) e a fase 1 parte da linha 9 e vai até a linha 14.

Fase 0

Um detalhe importante, é que, agora, devemos controlar o número de vértices de grau 1 em cada passo. Por isso, vamos definir por Y_0 e Y_1 as variáveis que

controlam os números de vértices de graus 0 e 1, respectivamente. Para a fase 0, a curva $z_0(x)$ pode ser construída da mesma forma que a curva $z(x)$ do exemplo do final do Capítulo 4. Assim, a partir do problema de valor inicial

$$z_0'(x) = -1 - \frac{dz_0(x)}{1-2x}, \quad z_0(0) = 1,$$

obtemos que

$$z_0(x) = \frac{(d-1)(1-2x)^{d/2} - (1-2x)}{d-2}.$$

A menor solução positiva de $z_0(x) = 0$ é

$$x^* = \frac{1}{2} - \frac{1}{2} \left(\frac{1}{d-1} \right)^{2/(d-2)},$$

conforme vimos antes.

Agora, note que, de forma análoga, obtemos um PVI para $z_1(x)$, ainda nessa primeira fase, dado por

$$z_1'(x) = \frac{dz_0(x) - (d-1)z_1(x)}{1-2x}, \quad z_1(0) = 0.$$

Substituindo pela expressão de $z_0(x)$ já determinada, esse PVI se torna

$$z_1'(x) = \frac{d(d-1)(1-2x)^{d/2} - d(1-2x)}{(d-2)(1-2x)} - \frac{(d-1)z_1(x)}{1-2x}, \quad z_1(0) = 0,$$

Essa é uma EDO Linear de primeira ordem, da forma

$$y' + f(x)y = g(x),$$

que pode ser resolvida com o Método do Fator Integrante, pela fórmula:

$$y(x) = e^{-\int f(x)dx} \int e^{\int f(x)dx} g(x)dx.$$

Realizando os cálculos, obtemos a função

$$z_1(x) = \frac{d((d-2)^2(1-2x)^{\frac{d-1}{2}} - (d-1)(d-3)(1-2x)^{\frac{d}{2}} - (1-2x))}{(d-2)(d-3)},$$

que, quando aplicada no ponto x^* , fornece

$$z_1(x^*) = \frac{d((d-2)u_d^{\frac{d-1}{2}} - u_d)}{d-3}$$

onde $u_d := (d-1)^{-2/(d-2)}$.

Dado algum $\theta > 0$ pequeno, defina $D_{1,\theta} = \{(x, z_0, z_1) : -\theta < x < x^*, 0 < z_0 < 1 + \theta, -\theta < z_1 < 1 + \theta\}$. Essa região é a mesma que a do Capítulo 4 mas com uma dimensão a mais, que é referente à função z_1 .

Precisamos provar que as curvas permanecem em $D_{1,\theta}$ durante o intervalo necessário para a análise. Sobre o comportamento dessas soluções, afirmamos que, no intervalo $[0, x^*]$, as curvas z_0 e z_1 têm suas imagens no intervalo $[0, 1]$ e o que justifica isso é o seguinte.

$z_0(0) = 1 \Rightarrow z_1'(0) > 0 \Rightarrow z_1(x) > 0$, para todo $x \in (0, \epsilon_1]$ para algum $\epsilon_1 > 0$. Portanto há um intervalo $(0, \epsilon_1]$ no qual ambas as soluções z_0 e z_1 têm imagem dentro de $(0, 1)$. Agora resta provar que elas permanecem entre 0 e 1 até que $x = x^*$. De fato, z_0 permanece maior que zero até sua primeira raiz positiva, que é x^* , conforme visto antes. Note que, se $x \in [0, x^*]$, então

$$z_1'(x) > 0 \Leftrightarrow dz_0(x) - (d-1)z_1(x) > 0 \Leftrightarrow z_1(x) < \frac{dz_0(x)}{d-1}$$

isto é, a função z_1 tem derivada positiva para valores de $z_1(x)$ muito menores do que $z_0(x)$ e, por isso, não pode alcançar zero antes de z_0 . Para ver que ambas permanecem menores do que 1, basta notar que cada uma é menor do que a soma de ambas, e que essa soma tem derivada $z_0'(x) + z_1'(x) \leq -1$ em $D_{1,\theta}$ com $z_0(0) + z_1(0) = 1$. Isso completa a análise de comportamento das curvas da fase 0.

Agora, podemos usar o Teorema 4.2 para obter um resultado de concentração das variáveis Y_0 e Y_1 nos momentos finais da fase zero. Com exceção de $D_{1,\theta}$, a verificação das hipóteses do Teorema 4.2 pode ser feita usando os mesmos

parâmetros que foram usados no exemplo do Capítulo 4, para ambas as variáveis. A conclusão é que, com probabilidade

$$1 - O\left(n^{1/4} \exp\left(-\frac{M^3}{(d+1)^3} n^{1/4}\right)\right),$$

para alguma constante $M > 0$, vale que

$$Y_i(t) = nz_i(t/n) + O(\lambda n), \quad i \in \{0, 1\},$$

para $0 \leq t \leq \sigma n$, onde $\sigma = \sigma(n)$ é o supremo dos x para os quais a solução $z(x)$ pode ser estendida antes de alcançar a distância $C\lambda$ das laterais do retângulo aberto $D_{1,\theta}$, para alguma constante $C > 0$. Note que $\sigma(n) \rightarrow x^*$ quando $n \rightarrow \infty$.

Fase 1

Uma das principais complicações de uma fase nova é que as condições iniciais não estão bem definidas, pois elas são, na verdade, variáveis aleatórias e o ponto de parada $\sigma(n)$ não é variável aleatória mas pode variar quando n cresce. Por outro lado, os limites acima permitem controlar a aproximação desses valores, em termos probabilísticos, pelos seus respectivos pontos de convergência. Além disso, se as equações que regem o desempenho da Fase 1 forem estáveis com relação às condições iniciais, os pontos finais da fase 1 poderão ser aproximados pelos pontos finais relativos às condições iniciais limite: $\hat{z}_0 = z_0(x^*)$, $\hat{z}_1 = z_1(x^*)$ e $\hat{x} = x^*$. Felizmente, a estabilidade é consequência da condição de Lipschitz do Teorema 4.2. Basta que o nosso sistema satisfaça essa condição.

Teorema 6.1 ([51]). *Suponha que \mathbf{y} satisfaça as equações*

$$\frac{dy_i}{dx} = g_i(x, \mathbf{y})$$

para $(x, \mathbf{y}(x))$ em um conjunto aberto e limitado \mathcal{D} , com condições iniciais $\mathbf{y}(0) = \hat{\mathbf{y}} = \hat{\mathbf{y}}(n)$. Seja \mathbf{z} outra solução, com condições iniciais $\mathbf{z}(0) = \hat{\mathbf{z}} = \hat{\mathbf{z}}(n)$. Suponha

que as funções g_i são Lipschitz em \mathcal{D} e que $|\hat{\mathbf{y}}(n) - \hat{\mathbf{z}}(n)| \rightarrow 0$ quando $n \rightarrow \infty$. Seja $x_1 = \inf\{x : (x, \mathbf{y}(x)) \notin \mathcal{D} \text{ ou } (x, \mathbf{z}(x)) \notin \mathcal{D}\}$. Então $|\mathbf{y}(x) - \mathbf{z}(x)| \rightarrow 0$ uniformemente para $x \in [0, x_1)$.

Nessa segunda fase tomamos apenas vértices de grau 1 na etapa de seleção, daí as equações diferenciais que aproximam as variáveis serão:

$$\begin{aligned} z'_0(x) &= -\frac{(d-1)dz_0(x)}{d-2d\hat{x}-2(d-1)(x-\hat{x})} \\ z'_1(x) &= -1 + \frac{(d-1)(dz_0(x) - (d-1)z_1(x))}{d-2d\hat{x}-2(d-1)(x-\hat{x})}, \end{aligned}$$

onde \hat{x} é o ponto de parada $\sigma(n)$ da fase anterior. Se as condições iniciais fossem iguais às condições finais (assintóticas) da fase anterior, teríamos $z_0 \equiv 0$ com um PVI na forma simplificada

$$\begin{cases} z'_1(x) = -1 - \frac{(d-1)z_1(x)}{d-2dx^* - 2(d-1)(x-x^*)}, \\ z_1(x^*) = \frac{d((d-2)u_d^{(d-1)/2} - u_d)}{d-3}. \end{cases} \quad (6.1)$$

onde

$$x^* = \frac{1}{2} - \frac{1}{2} \left(\frac{1}{d-1} \right)^{2/(d-2)}$$

e $u_d := (d-1)^{-2/(d-2)}$.

O ponto de parada dessa última fase ocorre quando não houver mais vértices de grau 1. Felizmente, a equação de $z'_1(x)$ também é linear e pode ser resolvida com o método do fator integrante, fornecendo

$$z_1(x) = -\frac{d-1+u_d-2(d-1)x}{(d-1)(d-3)} + (d-1+u_d-2(d-1)x)^{(d-1)/2} c_d$$

onde c_d fica determinada pelas condições iniciais:

$$c_d := \frac{(d-2)d^{(3-d)/2}(1-\sqrt{u_d})}{d-3}$$

Daí, a menor solução positiva de $z_1(x) = 0$ é

$$x_2 = \frac{d - 1 + u_d - d [(d - 2)(d - 1)(1 - \sqrt{u_d})]^{-2/(d-3)}}{2(d - 1)}.$$

Note que $x_2 < 1$ pois para todo $0 \leq x \leq x_2$ vale que $z'_0(x) + z'_1(x) \leq -1$ com $z_0(0) + z_1(0) = 1$.

6.3 Formalizando os resultados

Demonstração. (Teorema 3.9). A prova segue por duas aplicações sucessivas do Teorema 4.2, sendo uma para cada fase. A aplicação da fase 0 já foi executada na seção 4.3. Assim, resta a segunda aplicação, que, naturalmente, depende do estado final da primeira fase. A transição entre essas fases deve ser considerada.

Na nova fase, as condições iniciais são, na verdade, variáveis aleatórias resultantes da fase anterior, e o ponto de parada $\sigma(n)$ não é variável aleatória mas pode variar quando n cresce. Portanto, a fim de adaptar o PVI às configurações do teorema, vamos transladar o tempo para a origem por meio de uma operação de substituição: $s := x - \hat{x}$, onde \hat{x} é o valor do último x , no final da fase 0, dado por $\sigma(n)$. O objetivo disso é fazer com que a fase 1 comece no tempo $s = 0$ fixo, o que é exigido no enunciado do Teorema 4.2. O momento no qual a fase anterior termina será uma nova variável (constante) na segunda fase $\hat{X}(t) = \hat{x}(s)n$, que fará parte do sistema de equações com

$$\hat{x}'(s) = 0.$$

Assim o sistema pode ser reescrito como

$$\begin{cases} z'_0(s) = f_0(s, \hat{x}, z_0, z_1) \\ z'_1(s) = f_1(s, \hat{x}, z_0, z_1) \end{cases} \quad (6.2)$$

com condições iniciais $z_j(0) = Y_j(\sigma(n)n)/n$, $j \in \{1, 2\}$, e $\hat{x}(0) = \sigma(n)$, para algum n , onde

$$f_0(s, \hat{x}, z_0, z_1) := -\frac{(d-1)dz_0}{d-2d\hat{x}-2(d-1)s}$$

$$f_1(s, \hat{x}, z_0, z_1) := -1 + \frac{(d-1)(dz_0 - (d-1)z_1)}{d-2d\hat{x}-2(d-1)s}$$

$$f_2(s, \hat{x}, z_0, z_1) = 0$$

Relembremos ainda que $\sigma(n) \leq x^*$ e que $\sigma(n) \rightarrow x^*$ quando $n \rightarrow \infty$. Assim, exceto pela translação horizontal, as curvas e as condições iniciais com $n \rightarrow \infty$ são as mesmas do PVI (6.1).

Seja

$$D_{2,\theta} := \left\{ \begin{array}{l} -\theta < s < x_2 - x^*, \\ (s, \hat{x}, z_0, z_1) : -\theta < \hat{x} < x^* + \theta, \\ -\theta < z_0, z_1 < 1 + \theta, \end{array} \right\}$$

onde $\theta > 0$ é pequeno.

Devemos verificar as hipóteses do Teorema 4.2 para alguma região $D_{2,\theta}$. A região $D_{2,\theta}$ deve ser aberta, conexa e conter o fecho de

$$\{(0, z_0, z_1, \hat{x}) : \mathbb{P}[\hat{X}(0) = \hat{x}(0)n, Y_j(0) = z_j(0)n, j \in \{1, 2\}] \neq 0\}.$$

Queremos que ela também contenha as curvas z_0 e z_1 até que a curva z_1 intercepte o intervalo $[0, 1]$, mas também é preciso que as f_j sejam Lipschitz na região, por isso teríamos que excluir de $D_{2,\theta}$ uma vizinhança das soluções de $d-2d\hat{x}s-2(d-1)s = 0$. Com efeito, note que o fecho dos valores definidos para $z_0(0)$, $z_1(0)$ e $\hat{x}(0)$ estão

em $D_{2,\theta}$. Também é verdade que, em $D_{2,\theta}$,

$$\begin{aligned}
d - 2d\hat{x}s - 2(d-1)s &\geq d - 2d(x^* + \theta) - 2(d-1)(x_2 - x^*) \\
&\geq d - 2d\theta - 2(d-1)x_2 + 2x^* \\
&\geq d - 2d\theta - 2(d-1)\frac{1}{2} - 2x^* \\
&\geq 1 - 2x^* - 2d\theta \\
&> 0,
\end{aligned}$$

desde que θ seja suficientemente pequeno. Isso garante que a condição (iii) do Teorema 4.2 é satisfeita. Além disso, para aplicarmos o Método das Equações Diferenciais, é necessário verificar ainda as duas primeiras hipóteses do Teorema 4.2. A Hipótese de Limitação é satisfeita pois

$$|Y_i(t+1) - Y_\ell(t)| \leq d+1, \quad i = 1, 2.$$

para todo $t = 0, 1, 2, \dots, T_D$.

A Hipótese de tendência segue se verificarmos que

$$|\mathbb{E}[Y_i(t+1) - Y_i(t)|h_i] - f_i(t/n, Y_1(t)/n, Y_2(t)/n)| \leq o(1)$$

para $t \leq T_D$. Esse fato segue automaticamente quando se determinam as funções f_i , para $i = 1, 2, 3$.

A conclusão é que, com probabilidade

$$1 - O\left(n^{1/4} \exp\left(-\frac{N^3}{(d+1)^3} n^{1/4}\right)\right), \quad (6.3)$$

para alguma constante $N > 0$, vale que

$$Y_i(t) = nz_i(t/n) + O(\lambda n), \quad i \in \{0, 1\},$$

para $0 \leq t \leq \sigma_2 n$, onde $\sigma_2 = \sigma_2(n)$ é o supremo dos x para os quais a solução $z(x)$ pode ser estendida antes de alcançar a distância $C\lambda$ das laterais do retângulo

aberto $D_{2,\theta}$, para alguma constante $C > 0$, onde z_0 , z_1 e \hat{x} são as soluções do PVI (6.2).

A estabilidade das equações desse sistema permite que aproximemos as soluções gerais de (6.2) por soluções específicas para as condições iniciais de concentração: $z_0(0) = 0$, $z_1(0) = \frac{d((d-2)u_d^{\frac{d-1}{2}} - u_d)}{d-3}$ e $\hat{x}(0) = x^*$. Em particular, podemos obter que

$$Y_i(\sigma_2 n) = n z_i(x_2) + O(\lambda n), \quad i \in \{0, 1\}, \quad (6.4)$$

com a probabilidade dada pela expressão (6.3), desde que o final da primeira fase satisfaça

$$Y_i(\sigma_1 n) = n z_i(x^*) + O(\lambda n), \quad i \in \{0, 1\}. \quad (6.5)$$

Sejam E_1 e E_2 eventos dados pelas equações (6.5) e (6.4), respectivamente. Note que

$$\mathbb{P}[E_1 \cap E_2] = 1 - \mathbb{P}[\overline{E_1}] - \mathbb{P}[E_1] \mathbb{P}[\overline{E_2} | E_1] \geq 1 - \mathbb{P}[\overline{E_1}] - \mathbb{P}[\overline{E_2} | E_1],$$

onde,

$$\begin{aligned} \mathbb{P}[\overline{E_1}] &= O\left(n^{1/4} \exp\left(-\frac{M^3}{(d+1)^3} n^{1/4}\right)\right) e \\ \mathbb{P}[\overline{E_2} | E_1] &= O\left(n^{1/4} \exp\left(-\frac{N^3}{(d+1)^3} n^{1/4}\right)\right). \end{aligned}$$

Isso implica que $\mathbb{P}[E_1 \cap E_2] \rightarrow 1$ quando $n \rightarrow \infty$, fornecendo a concentração desejada. \square

Podemos interpretar cada fase do Algoritmo 5.1(a) como um algoritmo local de deleção da seguinte forma. Existe apenas uma cor transiente e uma cor de saída. Neste caso, o tipo de um vértice é dado por seu grau. Na fase 0, temos como entrada um grafo d -regular com todos os vértices da mesma cor transiente neutra. A regra de seleção escolhe com probabilidade uniforme um vértice u_t de grau d , de forma que ela claramente satisfaz os requisitos que regra de seleção deve ter. A regra de exploração local identifica as adjacências de u_t . Isso encerra a regra de

exploração local, que tem raio $r = 1$. Finalmente, a função de recoloração revela as adjacências identificadas pela regra de exploração local mas não altera suas cores, e atribui a cor de saída para u_t . O algoritmo da fase 1 possui como entrada um grafo d -regular com parte de seus vértices com a cor de saída e o restante dos vértices com a cor transiente. A regra de seleção escolhe com probabilidade uniforme um vértice u_t de grau $d - 1$ e a regra de exploração local e a função de recoloração são iguais às da fase 0. Isso estabelece o Teorema 3.10 como consequência de duas aplicações sucessivas do Teorema 5.8.

7 ALGORITMOS COM PRIORIDADE

O conceito de prioridade está relacionado a uma ordem de preferência entre as opções de escolha para os vértices pelo algoritmo. Podemos classificar em duas classes: *Local* e *Global*, que possuem abordagens substancialmente distintas.

Relembrando, dado um algoritmo local, chamamos de *etapa de seleção* a escolha do vértice inicial a cada passo e de *regra de exploração local* e *passo de recoloração* a escolha das adjacências investigadas e dos vértices revelados durante o passo, bem como das decisões que são tomadas a partir das informações obtidas. Por exemplo, a etapa de seleção corresponde precisamente à linha 5 do Algoritmo 4.1(a) e à linha 4 do Algoritmo 4.1(b) e a regra de exploração local corresponde às linhas 6 do Algoritmo 4.1(a) e 5 do Algoritmo 4.1(b) e o passo de recoloração corresponde aos fragmentos entre as linhas 7 e 14 do Algoritmo 4.1(a) e entre as linhas 6 e 13 do Algoritmo 4.1(b).

7.1 Prioridade local

A prioridade *local* é um critério de ordenamento das preferências aplicado durante a regra de exploração local e da função de recoloração que, em cada passo, redefine uma ordem específica para os vértices envolvidos no passo e é válida até o final do passo. Esse tipo de critério costuma ser simples e pode ser analisado com o MED, sendo possível obter otimizações com essa abordagem.

Com relação a análise do MED, a principal diferença ocorre nas equações diferenciais obtidas, onde as probabilidades associadas ficam concentradas em torno dos elementos que têm mais prioridade. Para ilustrar a diferença entre os tipos de

expressões que podem ser obtidas, apresentamos um cálculo de uma probabilidade para duas situações: na primeira não há prioridade local e na segunda há. O exemplo sem prioridade local vem do Algoritmo 4.1(b).

Relembrando, na linha 5 do Algoritmo 4.1(b), que está logo após a etapa de seleção, um vizinho u_t de v_t foi escolhido aleatoriamente, com probabilidade uniforme, dentre os vizinhos de v_t . Em vez disso, poderíamos exigir que u_t tivesse o menor grau possível dentre os vizinhos de v_t , o que caracterizaria uma prioridade local. As distribuições de probabilidade do grau de u_t nessas duas situações são diferentes e abaixo fornecemos um exemplo de como calculá-las.

Se um vértice v_t de grau 0 é selecionado no instante t :

- (a) a probabilidade de que um vizinho escolhido com probabilidade uniforme tenha grau j é igual a $Y_j(t)/S(t)$, enquanto que,
- (b) a probabilidade de que um vizinho escolhido entre seus vizinhos de grau mínimo tenha grau j é igual a

$$\left(\frac{\sum_{m=j}^d (d-m)Y_m(t)}{S(t)} \right)^d - \left(\frac{\sum_{m=j+1}^d (d-m)Y_m(t)}{S(t)} \right)^d + o(1).$$

Na diferença acima, a expressão da esquerda é a probabilidade de que todos os vizinhos de v_t tenham grau maior ou igual a j , a da direita é a probabilidade de que todos os vizinhos de v_t tenham grau maior ou igual a $j+1$. Portanto, a diferença é a probabilidade de que um vizinho de menor grau tenha grau j .

Nem todos os algoritmos de que tratamos admitem uma versão com priorização local que pode ser analisada por nossos métodos. Este é o caso do Algoritmo 4.1(b). Nele, atribuir prioridade poderia exigir modificações estruturais adicionais, o que pode não ser interessante. Por exemplo, para que o vértice u_t

fosse o de menor grau dentre os vizinhos de v_t , seria necessário conhecer o grau de todos os vizinhos de v_t . Para identificar os seus vizinhos, os pontos que ainda não foram emparelhados na célula referente a v_t devem ser emparelhados. Mas isso prejudica nossa opção de não incluir v_t no conjunto $D(t)$, pois seus vizinhos acabariam o passo com grau maior do que zero, e a correspondência entre os vértices não dominados e os vértices de grau zero seria perdida. Assim, outras modificações deveriam ser feitas.

7.2 Prioridade global

No Capítulo 8 apresentaremos uma nova contribuição, na qual, além de usarmos prioridade local, usaremos um tipo de prioridade que chamaremos de *global*. Optamos pela denominação “global” porque a ordem que estabelece a prioridade das classes é fixa durante todo o processo, funcionando, portanto, como um parâmetro do algoritmo. Essa prioridade se refere unicamente às etapas de seleção de cada instante t . Algoritmos com prioridades já foram analisados na literatura utilizando o Método das Equações Diferenciais, como vemos em Duckworth e Wormald [22] e Wormald [51].

A análise da prioridade global com o Método das Equações Diferenciais é um pouco mais complicada do que a prioridade local e exige considerações específicas para as quais dedicamos o restante desse capítulo.

7.2.1 Terminologias preliminares

Como vimos antes, é natural particionar o conjunto de vértices de um grafo G_t , produzido após t passos do algoritmo, em uma família finita de classes (vimos

exemplos onde os vértices eram particionados de acordo com o seu grau ou de acordo com o seu tipo). Um ingrediente importante na análise do algoritmo é contabilizar o número de vértices de cada classe. Assim, essa contagem será feita por variáveis, cujo valor depende de G_t e, por isso, variam conforme t cresce. Essa é um forma muito comum e bem geral de estrutura que o MED costuma analisar na literatura. A seguir precisaremos de algumas terminologias.

Suponhamos que existem r classes das quais s são selecionáveis pela etapa de seleção. Definimos o *ordenamento* $C(t) := (c_1(t), c_2(t), \dots, c_r(t))$ das classes c_i , com $1 \leq i \leq r$, tal que, se $1 \leq j < i \leq s \leq r$, então o conjunto de vértices $c_j(t)$ tem mais prioridade do que o conjunto $c_i(t)$ e se $i > s$ então os vértices de $c_i(t)$ não são selecionáveis. Para todo $1 \leq j \leq r$, definimos por $Y_j(t)$ a variável que contabiliza o número de vértices na classe j no instante t .

Em geral, quando um vértice de uma classe c_i é selecionado pela etapa de seleção do passo do instante t , uma regra de exploração local é executada a partir desse vértice até o final do passo. Damos o nome de *Operação* Op_i ao conjunto formado:

- pela etapa de seleção de um vértice da classe c_i e
- pela regra de exploração local que sucede essa etapa seleção.

O algoritmo será decomposto em um conjunto de *operações* Op_i com índices i no conjunto de classes C , onde, em cada instante t , exatamente uma dessas operações é escolhida e executada para fornecer G_{t+1} . Denotamos a operação realizada no instante t por op_t . Assim, um algoritmo fica bem definido pelo conjunto das operações Op_i e uma ordem de prioridade global especificada.

Algoritmo 7(a): com parâmetros $n \geq 4$, $d \geq 3$ e $C(0)$.

Entrada: Um grafo G d -regular com n vértices.

Saída: $C(T_D)$

```

1  $t = 0$ ;
2  $G_0 = G$ ;
3 enquanto  $Y_1(t) + \dots + Y_s(t) > 0$  faça
4    $i = \min\{m : Y_m(t) > 0\}$ ;
5   Execute  $Op_i$ , produzindo  $G_{t+1}$ ;
6    $t = t + 1$ ;
7 fim
8  $T_D = t$ 

```

Um exemplo simples pode ser visto no Capítulo 6. Relembre que inicialmente selecionávamos vértices de grau 0 para colocar no conjunto 2-dominante. Isso pode ser formalizado como uma operação Op_0 . Quando os vértices de grau zero se tornaram raros, fizemos a simples modificação de escolher somente vértices de grau 1 em vez dos de grau 0. Essa operação poderia ser denotada como Op_1 . Assim, poderíamos formalizar uma decomposição desse algoritmo pelo ordenamento $C(t) := (c_1(t), c_2(t), \dots, c_r(t))$, definindo que c_i é o conjunto de vértices de grau $i - 1$ em G_t e onde c_i tem prioridade sobre c_j se, e somente se, $i < j$. Note que $r = d$ e $s = 2$ (pois somente $c_1(t)$ e $c_2(t)$ são selecionáveis). Note que uma aplicação desse algoritmo produz uma lista ordenada $(op_0, op_1, op_2, \dots, op_{T_D})$ que, tipicamente, teria o seguinte aspecto $(Op_0, Op_0, \dots, Op_0, Op_1, Op_1, \dots, Op_1)$.

Na prática, essa decomposição é útil quando diferentes operações podem ser aplicadas em uma mesma fase, algo que não aconteceu no algoritmo do Capítulo 5. De fato, essa é a principal complicação que abordaremos aqui.

7.2.2 Algoritmos despriorizados: uma forma de simular um processo com prioridade global

Para abordar esse problema com o MED, devemos pensar nas recorrências das variações esperadas das variáveis Y_i 's também em função de Op_i . Cada operação, quando aplicada, causará suas próprias variações de forma que poderemos calcular

$$\mathbb{E}[Y_j(t+1) - Y_j(t) | G_t, op_t = Op_i] = f_j^{(i)}(Y(t)/n) + o(1)$$

onde $Y(t)$ é o vetor

$$(Y_1(t), \dots, Y_r(t)),$$

e as funções $f_j^{(i)}(z(x))$, com $1 \leq j \leq r$ e $1 \leq i \leq s$, estão bem definidas para cada vetor $z(x) := (z_1(x), \dots, z_r(x))$ num domínio Ω e satisfazem as hipóteses do Teorema 4.2.

Teoricamente, em um instante t específico, para que uma operação Op_i seja executada é necessário que $i = \min\{m : Y_m(t) > 0\}$. Após a aplicação de uma operação Op_i , vértices de classes c_j com $j < i$ poderão ser criados, sendo necessário executar as respectivas operações até que sejam eliminados todos os vértices das classes de índices menores do que i antes de aplicar novamente uma operação Op_i .

É importante mencionar que a análise das variações esperadas das variáveis se torna difícil devido às restrições de prioridade. De fato, se fôssemos escrever as equações diferenciais, seria necessário incluir funções delta de Dirac que determinam o grau mínimo de cada passo, o que produziria descontinuidades nas derivadas das soluções dessas equações. Uma abordagem alternativa foi feita por Wormald [51]. Neste trabalho, em vez de analisar diretamente a prioridade global, ele considerou um algoritmo para o qual um vetor de probabilidade

$$\alpha(x) = (\alpha_1(x), \alpha_2(x), \dots, \alpha_s(x))$$

é preescrito para cada G_t . Esse vetor depende apenas do número n de vértices do grafo original e de um parâmetro x , que está relacionado com o passo t do algoritmo. As coordenadas de $\alpha(x)$ são contínuas com relação a x .

Este novo algoritmo, em cada instante t , primeiro decide qual operação será executada, onde cada Op_i possui probabilidade $\alpha_i(x)$ de ser escolhida. Quando $op_t = Op_i$, um vértice da classe $c_i(t)$ é escolhido uniformemente para implementar a etapa de seleção. Chamamos esse novo algoritmo de despriorizado, pois o vértice da etapa de seleção não é mais determinado pela ordem de prioridades, as classes apenas possuem diferentes probabilidades de serem escolhidas. Apesar de algumas complicações serem geradas, tais como a necessidade de assegurar que G_t contenha um vértice da classe i caso a etapa de seleção peça por um vértice dessa classe, a análise se torna possível com um método similar ao usado anteriormente. A principal razão para isso é que as variações esperadas podem ser calculadas, em cada passo t , por

$$\mathbb{E}[Y_j(t+1) - Y_j(t) | G_t] = \sum_{i=1}^s \mathbb{E}[Y_j(t+1) - Y_j(t) | G_t, op_t = Op_i] \alpha_i^k(x),$$

onde $\alpha_i^k(x) = \mathbb{P}[op_t = Op_i | G_t]$.

Para que esta versão despriorizada aproxime o algoritmo priorizado original, Wormald define essas probabilidades em termos de propriedades que deveriam ser satisfeitas pelo algoritmo priorizado. Contudo, essa comparação é tecnicamente independente da análise do algoritmo despriorizado. Nosso objetivo é simular o algoritmo original, com prioridades, por uma versão despriorizada adequada. Para isso, nós o modelaremos diretamente a partir de algumas propriedades que o algoritmo despriorizado deveria satisfazer. O mesmo tipo de abordagem já foi usada anteriormente no contexto de largura de bisecção de grafos aleatórios d -regulares

por Díaz, Serna e Wormald [18] e no contexto de grafos aleatórios regulares por Shi e Wormald [47].

Primeiramente, está claro que

$$\sum_{i=1}^s \alpha_i^k(x) = 1, \quad \text{para cada } x,$$

pois, em cada passo, o algoritmo deve sortear uma classe, de onde será escolhido o vértice da etapa de seleção. Se o algoritmo está na fase k , todas as classes sorteadas devem possuir índice $j \leq k$, de forma que

$$\alpha_j^k(x) = 0 \quad \text{se } j > k.$$

Finalmente, durante a fase k , os vértices de classes com maior prioridade do que k não devem ser acumulados. Em termos quantitativos, isso significa que se $j < k$, devemos ter

$$\mathbb{E}[Y_j(t+1) - Y_j(t) | G_t] = \sum_{i=1}^k \mathbb{E}[Y_j(t+1) - Y_j(t) | G_t, \text{opt} = \text{Op}_i] \alpha_i^k(x) = 0.$$

Assim, se o algoritmo está na fase k , as probabilidades $\alpha_i^k(x)$ devem satisfazer

$$\frac{dz_j(x)}{dx} = \sum_{i=1}^k f_j^{(i)}(x) \alpha_i^k(x), \quad j = 1, 2, \dots, r \quad (7.1)$$

onde o vetor $\alpha^k(x) = (\alpha_1^k(x), \alpha_2^k(x), \dots, \alpha_k^k(x))$ é uma combinação convexa, que é solução do sistema matricial (7.2).

$$\begin{bmatrix} 1 & 1 & \cdots & 1 \\ f_1^{(1)}(z(x)) & f_1^{(2)}(z(x)) & \cdots & f_1^{(k)}(z(x)) \\ f_2^{(1)}(z(x)) & f_2^{(2)}(z(x)) & \cdots & f_2^{(k)}(z(x)) \\ \vdots & \vdots & \ddots & \vdots \\ f_{k-1}^{(1)}(z(x)) & f_{k-1}^{(2)}(z(x)) & \cdots & f_{k-1}^{(k)}(z(x)) \end{bmatrix} \begin{bmatrix} \alpha_1^k(x) \\ \alpha_2^k(x) \\ \alpha_3^k(x) \\ \vdots \\ \alpha_k^k(x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (7.2)$$

Se as probabilidades $\alpha_i^k(x)$ forem escolhidas de forma que satisfaçam o sistema linear (7.2), garantiremos que cada $z_1(x), z_2(x), \dots, z_{k-1}(x)$ tenha derivadas nulas, o que as manterá em zero caso sejam iniciadas assim. Isso só define o interior da fase quando as funções f forem “bem comportadas”. De fato, não está claro quando o sistema terá solução. Por isso, devemos estabelecer condições para que essa fase esteja bem definida.

Para as condições iniciais, devemos ter o seguinte:

Seja $M_k(x)$ a matriz do sistema (7.2).

(1) $|\det M_k(x_k)| > \gamma$ para algum $\gamma > 0$.

Justificativa. Isso garante que o sistema (7.2) tenha solução e que ela seja única. Além disso, a Regra de Cramer deixa bem definidas as funções

$$\alpha_i^k(x_k) = \frac{(-1)^{i-1} \det M_{k,i}(z(x_k))}{\det M_k(z(x_k))}, \quad i = 1, \dots, k,$$

onde $M_{k,i}(z(x))$ é a matriz obtida de $M_k(z(x))$ pela remoção da primeira linha e i -ésima coluna.

(2) *Pelo menos um dos seguintes itens é verdadeiro:*

(i) $\alpha_i^k(x_k) > \gamma$ para algum $\gamma > 0$ e para todo $i = 1, 2, \dots, k-1$

(ii) $\alpha_i^k(x_k) \geq 0$ para todo $i = 1, 2, \dots, k-1$ e sempre que $\alpha_i^k(x_k) = 0$, devemos ter $A_{i,k}(x_k)$ vazio ou então $\frac{d^{\nu_{i,k}(x_k)} \alpha_i^k(x_k)}{dx^{\nu_{i,k}(x_k)}} > 0$, onde

$$A_{i,k}(x) := \left\{ s \geq 1 : \frac{d^s \alpha_i^k(x)}{dx^s} \neq 0 \right\} \quad e \quad \nu_{i,k}(x) := \inf A_{i,k}(x).$$

Justificativa. Essa condição serve para garantir que as proporções α formem uma combinação convexa em um intervalo de comprimento mínimo não nulo $[x_k, \epsilon]$ para algum $\epsilon > 0$.

(3) *Ambos os itens devem ser verdadeiros:*

(i) $1 - \gamma > z_j(x_k) > \gamma$ para algum $\gamma > 0$ e para todo $j = 1, 2, \dots, k - 1$

(ii) $1 \geq z_j(x_k) \geq 0$ para todo $i = k, k + 1, \dots, r$ e sempre que $z_j(x_k) = 0$, devemos ter $B_{i,k}(x_k)$ vazio ou então $\frac{d^{X_{i,k}(x_k)} z_j(x_k)}{dx^{X_{i,k}(x_k)}} > 0$, e se $z_j(x_k) = 1$ devemos ter $B_{i,k}(x_k)$ vazio ou então $\frac{d^{X_{i,k}(x_k)} z_j(x_k)}{dx^{X_{i,k}(x_k)}} < 0$, onde

$$B_{i,k}(x) := \left\{ s \geq 1 : \frac{d^s z_j(x)}{dx^s} \neq 0 \right\} \quad e \quad \chi_{i,k}(x) := \inf B_{i,k}(x).$$

Justificativa. O primeiro item garante a existência de uma quantidade positiva das classes consumidas pelas operações (essas variáveis z deveriam ser nulas, mas por questões técnicas lipschitzianas, pode ser problemático permití-las assumir esse valor, por outro lado se esse valor for suficientemente pequeno a estabilidade garante que nos aproximemos tanto quanto desejamos da situação ideal) o segundo item apenas evita que as demais curvas z_j excedam a fronteira do intervalo $[0, 1]$ por pelo menos um instante arbitrariamente pequeno $[x_k, \epsilon]$.

Seja $M_k(t)$ a matriz do sistema (7.2) aplicada no ponto $z := (Y_1(t)/n, \dots, Y_r(t)/n)$ e seja $M_{k,i}(t)$ a matriz $M_k(t)$ após remover a primeira linha e a i -ésima coluna. Finalmente, o algoritmo da simulação fica com um aspecto do Algoritmo 7(b).

Algoritmo 7(b): com parâmetros $n \geq 4$, $d \geq 3$ e $C(0)$.

Entrada: Uma estrutura vazia G_0 com n vértices.

Saída: $C(T_D)$

1 $t = 0$;

2 **enquanto** $Y_1(t) + \dots + Y_s(t) > \epsilon n$ e $\det M_k(t) > \epsilon n$ **faça**

3 $k = \min\{m : Y_m(t) > \epsilon n\}$;

4 Sejam $\alpha_i^k(t) := \frac{(-1)^{i-1} \det M_{k,i}(t)}{\det M_k(t)}$, $i = 1, \dots, k$;

5 Sorteie j segundo a distribuição $\mathbb{P}[j = i] = \alpha_i^k(t)$;

6 Execute Op_j , produzindo G_{t+1} ;

7 $t = t + 1$;

8 **fim**

9 $T_D = t$

8 MELHORANDO AS COTAS PARA 2-DOMINÂNCIA EM GRAFOS ALEATÓRIOS REGULARES

No caso dos grafos 3-regulares, as cotas fornecidas pelo algoritmo anterior não são boas. De fato, seu valor é aproximadamente 0,507, que está acima da melhor cota geral conhecida que é 0,5 (lembre que essa cota é justa, pois temos, por exemplo, $K_{3,3}$ onde o menor conjunto 2-dominante possui três vértices).

Por outro lado, isso não significa que o método apresentado não se aplica a grafos 3-regulares, mas somente que o algoritmo proposto no Capítulo 6 não é adequado. Quando $n \rightarrow \infty$, a probabilidade de um grafo aleatório 3-regular ter número de 2-dominância entre 0,48 e 0,5 tende a um número maior do que zero? Poderíamos pensar que sim devido a existência de famílias bem conhecidas de grafos 3-regulares com número de 2-dominância igual a $0,5n$ e também para números de 2-dominância ligeiramente menores. No entanto, fornecemos uma resposta negativa a essa questão provando que essa probabilidade converge para zero. A prova disso é uma das consequências do que mostraremos nessa seção.

De fato, durante nossa busca por algoritmos eficientes para 2-dominância, encontramos alguns que, em particular, são mais eficientes do que o do Capítulo 6, porém são mais elaborados. Dentre eles, escolhemos aquele que ofereceu os melhores valores para apresentar aqui.

Nessa seção, vamos descrever e analisar esse algoritmo com o Método das Equações Diferenciais a fim de formalizar nossos resultados.

8.1 Definindo as operações do nosso algoritmo

Para definir e analisar um algoritmo que produz conjuntos 2-dominantes, além de controlar o número de vértices de cada grau, estamos interessados em saber quais vértices já foram 1-dominados. Por isso introduzimos duas cores para os vértices: a cor azul a , que indica que o vértice não foi dominado nenhuma vez, e vermelho v , que indica que ele já foi dominado uma vez. Portanto, definimos a variável $Y_{i,c}(t)$, onde $\{i, c\} \in I := (\{0, 1, 2, \dots, d-1\} \times \{v, a\}) \setminus (0, v)$, como o número de vértices de grau i e cor c no instante t .

O algoritmo inicia com conjuntos vazios $D_2(0)$ e $D(0)$ que servem para identificar os vértices já incluídos no conjunto 2-dominante e aqueles que já foram 2-dominados, respectivamente. Da mesma maneira que nas seções anteriores, podemos considerar duas versões do algoritmo, uma de deleção e outra de construção. A estrutura vazia G_0 é a mesma definida em algoritmos de capítulos anteriores.

Em cada passo, a etapa de seleção do algoritmo seleciona um vértice vermelho de grau máximo disponível e escolhe um vizinho azul de grau mínimo desse vértice para remover e pôr no conjunto 2-dominante. Se o vértice selecionado na etapa de seleção não possui vizinhos azuis, então ele próprio é adicionado ao conjunto 2-dominante. Em seguida, os vizinhos azuis do vértice adicionado ao conjunto 2-dominante passam a ter cor vermelha e os que já eram vermelhos são removidos do grafo, terminando o passo. Esse processo pode durar enquanto houver vértices vermelhos disponíveis para o passo de seleção. Todos os vértices removidos recebem uma cor de saída verde e são contabilizados por uma variável $Q(t)$. Vértices verdes são vértices que já foram 2-dominados.

Algoritmo 8(a): com parâmetros $n \geq 4$, $d \geq 3$.

Entrada: Um grafo d -regular G com n vértices e com seus vértices todos da cor azul.

Saída: Um conjunto 2-dominante D_2 de G .

```
1  $t = 0$ ;  
2  $H_0 = G$ ;  
3 enquanto  $Y_{1,v}(t) + \dots + Y_{d-1,v}(t) > 0$  faça  
4    $i = \min\{m : Y_{m,v}(t) > 0\}$ ;  
5   Escolha um vértice  $v_t$  de grau  $i$  com probabilidade uniforme;  
6   Revele os vizinhos de  $v_t$   
7   se  $v_t$  possui vizinho de cor azul então  
8     Selecione um vizinho azul de grau mínimo qualquer  $u_t$  de  
9      $v_t$ ;  
9     Defina  $D_2(t+1) := D_2(t) \cup \{u_t\}$ ;  
10    Revele todos os vizinhos de  $u_t$ ;  
11    Defina  $D(t+1) := D(t) \cup \{u_t$  e seus vizinhos vermelhos};  
12    Remova os vizinhos vermelhos de  $u_t$  ;  
13    Altere a cor dos vizinhos azuis de  $u_t$  para vermelho;  
14  senão  
15    Defina  $D_2(t+1) := D_2(t) \cup \{v_t\}$ ;  
16    Defina  $D(t+1) := D(t) \cup \{v_t$  e seus vizinhos};  
17    Remova  $v_t$  e seus vizinhos ;  
18  fim  
19  Denote a estrutura obtida após esses emparelhamentos por  
20     $H_{t+1}$ ;  
20     $t = t + 1$ ;  
21 fim  
22  $D_2 = D_2(t) \cup V(G_0) \setminus D(t)$ ;
```

Este algoritmo admite a versão de construção a seguir.

Algoritmo 8(b): com parâmetros $n \geq 4$, $d \geq 3$.

Entrada: Uma estrutura vazia G_0 com n vértices e com seus vértices todos da cor azul.

Saída: Um multigrafo aleatório d -regular G com n vértices e um conjunto 2-dominante D_2 de G .

```
1  $t = 0$ ;  
2  $G_0 = G$ ;  
3 enquanto  $Y_{1,v}(t) + \dots + Y_{d-1,v}(t) > 0$  faça  
4    $i = \max\{m : Y_{m,v}(t) > 0\}$ ;  
5   Escolha um vértice  $v_t$  de grau  $i$  com probabilidade uniforme;  
6   Emparelhe  $v_t$   
7   se  $v_t$  possui vizinho de cor azul então  
8     Selecione um vizinho azul de grau mínimo qualquer  $u_t$  de  
9      $v_t$ ;  
10    Defina  $D_2(t+1) := D_2(t) \cup \{u_t\}$ ;  
11    Emparelhe  $u_t$  totalmente;  
12    Defina  $D(t+1) := D(t) \cup \{u_t$  e seus vizinhos vermelhos};  
13    Emparelhe totalmente os vizinhos vermelhos de  $u_t$  ;  
14    Altere as cores de  $u_t$  e seus vizinhos vermelhos para verde;  
15    Altere a cor dos vizinhos azuis de  $u_t$  para vermelho;  
16  senão  
17    Defina  $D_2(t+1) := D_2(t) \cup \{v_t\}$ ;  
18    Defina  $D(t+1) := D(t) \cup \{v_t$  e seus vizinhos};  
19    Emparelhe totalmente os vizinhos de  $v_t$  ;  
20    Altere as cores de  $v_t$  e seus vizinhos para verde;  
21  fim  
22  Denote a estrutura obtida após esses emparelhamentos por  
23   $G_{t+1}$ ;  
24   $t = t + 1$ ;  
25 fim  
26  $D_2 = D_2(t) \cup V(G_0) \setminus D(t)$ ;
```

Chamaremos de aplicação da operação Op_i o fragmento do algoritmo 8(a) que está entre as linhas 5 e 18.

O vértice da etapa de seleção é um dos vértices que será 2-dominado ao final de cada passo. Intuitivamente, parece interessante que o vértice adicionado ao conjunto dominante seja azul, pois isso faz com que ele seja “dominado duas vezes” no próprio passo e evita o desperdício de uma primeira dominação herdada de passos anteriores. Também é interessante que ele seja de grau mínimo para que o número de novos vértices dominados por ele seja máximo. Quando o vértice da etapa de seleção não possui vizinhos azuis, optamos por colocar no conjunto 2-dominante o próprio vértice da etapa de seleção para garantir um número grande de vértices 2-dominados ao final de um passo desse tipo.

8.2 Encontrando as recorrências e as equações diferenciais associadas às operações Op_i do algoritmo dado

Primeiramente devemos encontrar as recorrências associadas a cada operação Op_i para cada i fixo. Vamos denotar por $Y(t)$ o vetor

$$(Y_{0,a}(t), Y_{1,v}(t), Y_{1,a}(t), \dots, Y_{d-1,v}(t), Y_{d-1,a}(t)).$$

Seja

$$S_c(Y(t)) := \sum_{m=0}^{d-m} (d-i) Y_{m,c}(t).$$

Quando um vértice v_t de grau i é escolhido na etapa de seleção, $(d-i)$ novos vizinhos de v_t são revelados. A probabilidade de que ele tenha ℓ vizinhos azuis é de

$$A_{i,\ell}^d(Y(t)) := \binom{d-i}{\ell} \cdot \frac{S_a(Y(t))^\ell S_v(Y(t))^{d-i-\ell}}{(S_a(Y(t)) + S_v(Y(t)))^{d-i}} + o(1).$$

Se v_t possui ℓ vizinhos azuis, então a probabilidade de que o menor grau entre esses azuis seja n é de

$$B_{k,\ell}^d(Y(t)) := \left(\frac{\sum_{m=k}^d (d-m)Y_{m,a}(t)}{S_a(Y(t))} \right)^\ell - \left(\frac{\sum_{m=k+1}^d (d-m)Y_{m,a}(t)}{S_a(Y(t))} \right)^\ell + o(1).$$

Aqui definimos $Y_{d,a}(t) \equiv Y_{d,v}(t) \equiv Y_{-1,a}(t) \equiv Y_{-1,v}(t) \equiv Y_{1,v}(t) \equiv 0$ por simplicidade.

Agora, vamos supor que v_t tem grau i , $\ell \geq 1$ vizinhos azuis e o vizinho azul escolhido para ser u_t tem grau k . Então o valor esperado da variação do número de vértices de grau j e cor c é de

$$\begin{aligned} C_{i,k,\ell}^{d,j,c}(Y(t)) &:= -\delta_{j,i}\delta_{c,v} - \delta_{j,k}\delta_{c,a} \\ &+ (d-k-1) \left(\frac{\delta_{c,v}(d-j+1)Y_{j-1,a}(t)}{S_v(Y(t)) + S_a(Y(t))} - \frac{(d-j)Y_{j,c}(t)}{S_v(Y(t)) + S_a(Y(t))} \right) \\ &+ (d-k-1) \sum_{m=1}^{d-2} \frac{(d-m)Y_{m,v}(t)(d-m-1)}{S_v(Y(t)) + S_a(Y(t))} \left(\frac{(d-j+1)Y_{j-1,c}(t) - (d-j)Y_{j,c}(t)}{S_v(Y(t)) + S_a(Y(t))} \right) \\ &+ \delta_{c,v}(d-i-\ell) \left(\frac{(d-j+1)Y_{j-1,v}(t) - (d-j)Y_{j,v}(t)}{S_v(Y(t))} \right) \\ &+ \delta_{c,a} \left(\sum_{m=k}^{d-1} \delta_{m,j} \right) (\ell-1) \left(\frac{(1-\delta_{k,j})(d-j+1)Y_{j-1,a}(t) - (d-j)Y_{j,a}(t)}{\sum_{m=k}^{d-1} (d-m)Y_{m,a}(t)} \right) + o(1). \end{aligned}$$

Note que acima há seis termos. O primeiro termo é devido à possibilidade de o vértice v_t ser do mesmo tipo do vértice que está sendo medido. O segundo termo vem da possibilidade de u_t ser do mesmo tipo do vértice sendo medido. O terceiro termo é devido a mudança de cor e grau dos demais vizinhos de u_t revelados no passo (o que desconsidera v_t). Aqui, o termo negativo é a probabilidade de um vizinho de u_t ser do mesmo tipo que os vértices que estão sendo medidos, enquanto que o termo positivo é devido à possibilidade de estar medindo um vértice vermelho e o vizinho atingido por u_t ser um vértice azul cujo grau seja uma unidade menor do que o grau dos vértices que estão sendo medidos. O quarto termo é devido

à mudança de grau dos vizinhos vermelhos de u_t considerados no terceiro termo, já que esses vizinhos são emparelhados completamente nesse passo. O quinto e o sexto termos são devidos aos vizinhos de v_t sem considerar u_t . O quinto termo trata da mudança do grau dos vizinhos vermelhos de v_t e só entra na expressão quando o vértice medido for vermelho, e o sexto termo, que trata da mudança de grau dos vizinhos azuis de v_t , só entra na expressão quando o vértice medido for azul.

Por outro lado, se v_t não tem vizinhos azuis, o valor esperado dessa variação é

$$D_i^{d,j,c}(Y(t)) := -\delta_{j,i}\delta_{c,v} - \delta_{c,v}(d-i)\frac{(d-m)Y_{j,v}(t)}{S_v(Y(t))}$$

$$(d-i)\sum_{m=1}^{d-2}\frac{(d-m)Y_{m,v}(t)}{S_v(Y(t))}\left((d-m-1)\frac{(d-j+1)Y_{j-1,c}(t) - (d-j)Y_{j,c}(t)}{S_v(Y(t)) + S_a(Y(t))}\right)$$

$$+o(1).$$

O primeiro termo é devido à remoção de v_t , o segundo é devido à remoção dos vizinhos de v_t e o terceiro é devido à mudança nos graus na vizinhança dos vizinhos de v_t sem contar v_t .

Com isso podemos finalmente obter que $\mathbb{E}[Y_{j,c}(t+1) - Y_{j,c}(t)|G_t, op_t = Op_i]$ é igual a

$$\sum_{\ell=1}^{d-i}\left(A_{i,\ell}^d(Y(t))\sum_{k=0}^{d-1}B_{k,\ell}^d(Y(t))C_{i,k,\ell}^{d,j,c}(Y(t))\right) + A_{i,0}^d(Y(t))D_i^{d,j,c}(Y(t))$$

Será interessante nesta análise controlar o tamanho do conjunto 2-dominado $D(t)$. Assim, introduzimos a variável $Q(t) := |D(t)|$. Precisamos encontrar $\mathbb{E}[Q(t+1) - Q(t)|G_t, op_t = Op_i]$. Suponhamos que v_t tem pelo menos um vizinho azul e o vizinho azul escolhido para ser u_t tem grau k . Então o valor esperado da variação

de $Q(t)$ é

$$E_k^d(Y(t)) := 2 + (d - k - 1) \frac{S_v(Y(t))}{S_v(Y(t)) + S_a(Y(t))}.$$

Mas se v_t tem apenas vizinhos vermelhos, então esse valor esperado é de $(d - i + 1)$.

Isso permite concluir que $\mathbb{E}[Q(t + 1) - Q(t) | G_t, op_t = Op_i]$ é igual a

$$\sum_{\ell=1}^{d-i} \left(A_{i,\ell}^d(Y(t)) \sum_{k=0}^{d-1} B_{k,\ell}^d(Y(t)) E_k^d(Y(t)) \right) + A_{i,0}^d(Y(t))(d - i + 1)$$

Agora, fazemos as substituições $x := t/n$, $q(x) := Q(xn)/n$ e $z_{j,c}(x) := Y_{j,c}(xn)/n$, o que transforma, por exemplo, as expressões $S_c(Y(x))$, $A_{i,\ell}^d(Y(x))$ e $B_{k,\ell}^d(Y(x))$ anteriores em

$$s_c(z(x)) := \sum_{m=0}^{d-m} (d - i) z_{m,c}(x),$$

$$a_{i,\ell}^d(z(x)) := \binom{d-i}{\ell} \cdot \frac{s_a(z(x))^\ell s_v(z(x))^{d-i-\ell}}{(s_a(z(x)) + s_v(z(x)))^{d-i}} + o(1) \quad \text{e}$$

$$b_{k,\ell}^d(z(x)) := \left(\frac{\sum_{m=k}^{d-1} (d-m) z_{m,a}(x)}{s_a(z(x))} \right)^\ell - \left(\frac{\sum_{m=k+1}^{d-1} (d-m) z_{m,a}(x)}{s_a(z(x))} \right)^\ell + o(1),$$

respectivamente, onde

$$z(x) := (z_{0,a}(x), z_{1,v}(x), z_{1,a}(x), \dots, z_{d-1,v}(x), z_{d-1,a}(x)).$$

De forma análoga construímos $c_{i,k,\ell}^{d,j,c}(z(x))$ a partir de $C_{i,k,\ell}^{d,j,c}(Y(t))$, $d_i^{d,j,c}(z(x))$ a partir de $D_i^{d,j,c}(Y(t))$ e $e_k^d(z(x))$ a partir de $E_k^d(Y(t))$.

Finalmente, para todo $0 \leq j \leq d - 1$ e $c \in \{v, a\}$, podemos definir funções $f_{j,c}^{(i)}(z(x))$ para cada uma das operações Op_i , por

$$f_{j,c}^{(i)}(z(x)) := \sum_{\ell=1}^{d-i} \left(a_{i,\ell}^d(z(x)) \sum_{k=0}^{d-1} b_{k,\ell}^d(z(x)) c_{i,k,\ell}^{d,j,c}(z(x)) \right) + a_{i,0}^d(z(x)) d_i^{d,j,c}(z(x)).$$

Para $q(x)$ obtemos também

$$f_q^{(i)}(z(x)) := \sum_{\ell=1}^{d-i} \left(a_{i,\ell}^d(z(x)) \sum_{k=0}^{d-1} b_{k,\ell}^d(z(x)) e_k^d(z(x)) \right) + a_{i,0}^d(z(x))(d - i + 1).$$

8.3 Definindo e analisando um algoritmo eficiente

Até aqui, obtemos as expressões das funções f associadas a cada variável de cada operação Op_i . Resta definir a regra que decide qual operação será realizada em cada passo. Vamos definir que a operação realizada é a operação do tipo (i, v) onde i é o menor possível. Então, o que o algoritmo faz é selecionar um vértice vermelho de grau máximo v_t e, caso ele possua algum vizinho azul, acrescentar um vizinho azul de grau mínimo u_t no conjunto 2-dominante e atualiza a coloração e os emparelhamentos (isto é, os vizinhos vermelhos do vértice adicionado ao conjunto 2-dominante são totalmente emparelhados e os vizinhos azuis têm sua cor alterada para vermelho). Se v_t não possui vizinho azul, o algoritmo adiciona o próprio vértice no conjunto 2-dominante e faz a subsequente atualização. Isso completa a descrição do melhor algoritmo obtido para grafos 3 e 4- regulares.

Para d em geral esse algoritmo passa por vários estágios. No começo do processo, cada vez que executamos Op_1 podemos gerar alguns poucos vértices vermelhos de graus maiores do que 1, ficando, digamos, com $Y_{i,c}(x) = o(n)$, para certos $i > 1$. Isso faz com que tenhamos de executar uma sequência de Op_i com $i > 1$ até eliminar todos esses tipos de vértices e daí voltamos a executar Op_1 novamente. Inicialmente, os vértices vermelhos de grau $i > 1$ são eliminados imediatamente de forma que $z_i(x)$ permanece igual a zero para todo $i > 1$. Mas, conforme o processo avança, a remoção de um vértice vermelho de grau 2 tenderá a produzir novos vértices vermelhos de grau 2, de forma que seja inevitável que alguma variável $z_i(x)$ com $i > 1$, digamos $i = 2$, se torne positiva. Quando isso acontece, a fase 1 termina e uma nova fase se inicia com os vértices de grau 2 assumindo o papel dos vértices de grau 1.

Conforme vimos no Capítulo 7, esse tipo de algoritmo apresenta prioridades locais e globais. Portanto, o que faremos é analisar um processo que simula a execução do nosso algoritmo, isto é, analisaremos um processo no qual sabemos a probabilidade de cada operação ser executada em cada instante x . A principal hipótese dessa simulação é que, na fase k , $z_{i,v}(x) = 0$ para todo $i > k$, o que pode ser modelado escolhendo uma distribuição de probabilidades conveniente para cada Op_i em cada instante x . Se estamos na fase k , queremos uma distribuição de probabilidades $(\alpha_k^k(x), \alpha_{k+1}^k(x), \dots, \alpha_{d-1}^k(x))$ tal que $\alpha_k^k(x) + \dots + \alpha_{d-1}^k(x) = 1$ e que, para cada $i \in \{k+1, k+2, \dots, d-1\}$, tenhamos que

$$\alpha_k^k(x) f_{i,v}^k(z(x)) + \alpha_{k+1}^k(x) f_{i,v}^{k+1}(z(x)) + \dots + \alpha_{d-1}^k(x) f_{i,v}^{d-1}(z(x)) = 0.$$

Em outras palavras, queremos determinar o vetor $(d-k)$ -dimensional

$$(\alpha_k^k(x), \alpha_{k+1}^k(x), \dots, \alpha_{d-1}^k(x))$$

que é solução da equação matricial

$$\begin{bmatrix} 1 & 1 & \dots & 1 \\ f_{k+1,v}^{(k)}(z(x)) & f_{k+1,v}^{(k+1)}(z(x)) & \dots & f_{k+1,v}^{(d-1)}(z(x)) \\ f_{k+2,v}^{(k)}(z(x)) & f_{k+2,v}^{(k+1)}(z(x)) & \dots & f_{k+2,v}^{(d-1)}(z(x)) \\ \vdots & \vdots & \ddots & \vdots \\ f_{d-1,v}^{(k)}(z(x)) & f_{d-1,v}^{(k+1)}(z(x)) & \dots & f_{d-1,v}^{(d-1)}(z(x)) \end{bmatrix} \begin{bmatrix} \alpha_k^k(x) \\ \alpha_{k+1}^k(x) \\ \alpha_{k+2}^k(x) \\ \vdots \\ \alpha_{d-1}^k(x) \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (8.1)$$

para cada x . Assim, se $M_k(z(x))$ é a matriz da equação acima e $M_{k,i}(z(x))$ é a matriz obtida pela remoção da primeira linha e da i -ésima coluna de $M_k(z(x))$, então a Regra de Cramer estabelece que

$$\alpha_i^k(x) = \frac{(-1)^{i-k} \det M_{k,i}(z(x))}{\det M_k(z(x))}, \quad i = k, \dots, d-1, \quad (8.2)$$

desde que

$$\det M_k(z(x)) \neq 0. \quad (8.3)$$

Vamos supor que (8.3) é verdadeira para todos os pontos $z(x)$ de interesse. Neste caso o vetor $\alpha^1(x)$ está bem definido e podemos considerar o PVI

$$\begin{aligned}\frac{dz_{j,c}(x)}{dx} &= \sum_{i=1}^{d-1} \alpha_i^1(x) f_{j,c}^{(i)}(z(x)), \\ \frac{dq(x)}{dx} &= \sum_{i=1}^{d-1} \alpha_i^1(x) f_q^{(i)}(z(x)), \\ q(0) &= \hat{q}^0, \quad z_{j,c}(0) = \hat{z}_{j,c}^0, \quad (j, c) \in I,\end{aligned}\tag{8.4}$$

onde os valores $\hat{z}_{j,c}^0$ e \hat{q}^0 serão discutidos a seguir.

8.4 Obtendo condições iniciais adequadas

Note que não podemos usar as condições iniciais triviais $z(x) = (1, 0, \dots, 0)$ por causa da necessidade de existirem vértices vermelhos para a etapa de seleção de uma operação. Também não é interessante que algum $z_{j,c}$ intercepte a fronteira do intervalo $[0, 1]$. Assim definimos uma etapa de pré-processamento que produz vértices de todos os tipos em uma quantidade pequena, mas não-negligível e antecede o algoritmo definido anteriormente.

As entradas do algoritmo são um parâmetro p_0 e um grafo G'_0 vazio com n vértices. O que a etapa de pré-processamento faz é apenas acrescentar em D_2 cada vértice de G'_0 independentemente com probabilidade p_0 e realiza as atualizações de coloração e emparelhamentos típicas do algoritmo anterior: vértices em D_2 são emparelhados completamente e sua cor é alterada para verde, vértices com 1 vizinho em D_2 têm sua cor alterada para vermelho, e vértices com pelo menos 2 vizinhos em D_2 são emparelhados completamente e sua cor é alterada para verde. Com isso, a etapa de pré-processamento termina e o grafo de saída é denominado por $G'(p_0)$, que servirá de entrada para o algoritmo anterior.

Algoritmo 8(c): com parâmetros $n \geq 4$, $d \geq 3$.

Entrada: Um multigrafo colorido com n vértices $G'(p_0)$ produzido pela etapa de pré-processamento.

Saída: Um multigrafo aleatório d -regular G com n vértices e um conjunto 2-dominante D_2 de G .

1 $t = 0$;

2 $G_0 = G'(p_0)$;

3 **enquanto** $Y_{1,v}(t) > \gamma_1 n, \dots, Y_{d-1,v}(t) > \gamma_{d-1} n$ e $\det M_k(t) > \epsilon n$

faça

4 $k = \max\{m : Y_m(t) > 2\gamma_m n\}$;

5 Sejam $\alpha_i^k(t) := \frac{(-1)^{i-1} \det M_{k,i}(Y(t)/n)}{\det M_k(Y(t)/n)}$, $i = 1, \dots, k$;

6 Sorteie j segundo a distribuição $\mathbb{P}[j = i] = \alpha_i^k(t)$;

7 Escolha um vértice v_t de grau j com probabilidade uniforme;

8 Emparelhe v_t ;

9 **se** v_t possui vizinho de cor azul **então**

10 Escolha um vizinho azul de grau mínimo u_t de v_t ;

11 Defina $D_2(t+1) := D_2(t) \cup \{u_t\}$ e emparelhe u_t totalmente;

12 Defina $D(t+1) := D(t) \cup \{u_t$ e seus vizinhos vermelhos};

13 Emparelhe totalmente os vizinhos vermelhos de u_t ;

14 Altere as cores de u_t e seus vizinhos vermelhos para verde;

15 Altere as cores dos vizinhos azuis de u_t para vermelho;

16 **senão**

17 Defina $D_2(t+1) := D_2(t) \cup \{v_t\}$;

18 Defina $D(t+1) := D(t) \cup \{v_t$ e seus vizinhos};

19 Emparelhe totalmente os vizinhos de v_t ;

20 Altere as cores de v_t e seus vizinhos para verde;

21 **fim**

22 Denote a estrutura obtida após esses emparelhamentos por

G_{t+1} ;

23 $t = t + 1$;

24 **fim**

25 $D_2 = D_2(t) \cup V(G_0) \setminus D(t)$;

Aqui, vamos denotar o número de vértices de G_0 com grau j e cor c por $\widehat{Y}_{j,c}^0$, com a respectiva proporção denotada por $\widehat{z}_{j,c}^0 := \widehat{Y}_{j,c}^0/n$. Com um certo abuso de notação, diremos que $u \in \widehat{Y}_{j,c}^0$ é o evento no qual u tem grau j e cor c .

Teorema 8.1. *Temos que*

$$(a) \lim_{p_0 \rightarrow 0^+} \mathbb{E}[\widehat{z}_{j,c}^0(p_0)] = \begin{cases} 1 & \text{se } (j, c) = (0, a) \\ 0 & \text{c.c.} \end{cases} \quad \text{e esta é uma convergência} \\ \text{monótona.}$$

$$(b) \widehat{z}_{j,c}^0(p_0) = \mathbb{E}[\widehat{z}_{j,c}^0(p_0)] + o(1), \text{ a.q.c.}$$

$$(c) \text{ Dado } p_0 > 0 \text{ existe } \epsilon(p_0) > 0 \text{ tal que } \epsilon(p_0) \leq \widehat{z}_{j,c}^0(p_0) \leq 1 - \epsilon(p_0) \text{ a.q.c.}$$

Demonstração. Primeiramente, note que

$$\mathbb{E}[\widehat{z}_{j,c}^0(p_0)] = \mathbb{E}\left[\frac{\widehat{Y}_{j,c}^0(p_0)}{n}\right] = \frac{1}{n} \mathbb{E}\left[\sum_{u \in V(G)} 1_{u \in \widehat{Y}_{j,c}^0(p_0)}\right] = \mathbb{P}\left[u \in \widehat{Y}_{j,c}^0(p_0)\right]. \quad (8.5)$$

Agora, para que um vértice u tenha grau $j < d$ e cor v em G_0 é necessário que não esteja em D_2 , o que tem probabilidade $(1 - p_0)$, mas deve possuir exatamente um vizinho em D_2 , o que ocorre com probabilidade $p_0(1 - p_0)^{d-1}$. Dos $d - 1$ vizinhos fora de D_2 , $j - 1$ devem estar em D . Há $\binom{d}{1} \binom{d-1}{j-1}$ formas disso ocorrer. Note que um vértice só é responsável por um emparelhamento se for colocado em D_2 ou se possuir pelo menos dois vizinhos em D_2 . Assim, cada um dos vizinhos de u em D deve possuir pelo menos dois vizinhos em D_2 para que sejam emparelhados com u . A probabilidade de cada uma dessas ocorrências é de $1 - (1 - p_0)^{d-1} - \binom{d-1}{1} p_0(1 - p_0)^{d-2}$. Quanto aos seus vizinhos não revelados, cada um desses pode possuir no máximo um vizinho em D_2 , pois não são emparelhados com u . Essa probabilidade é de $(1 - p_0)^{d-1} + \binom{d-1}{1} p_0(1 - p_0)^{d-2}$. Finalmente, podemos escrever

$$\mathbb{P}[v \in \widehat{Y}_{j,v}^0(p_0)] = (1-p_0) \binom{d}{1} \binom{d-1}{j-1} p_0 (1-p_0)^{d-1} \quad (8.6)$$

$$\begin{aligned} & \times \left(1 - (1-p_0)^{d-1} - \binom{d-1}{1} p_0 (1-p_0)^{d-2} \right)^{j-1} \\ & \times \left((1-p_0)^{d-1} + \binom{d-1}{1} p_0 (1-p_0)^{d-2} \right)^{d-j} \\ & = \left(\binom{d}{1} \binom{d-1}{j-1} p_0 + p_0^2 \text{pol}(p_0) \right) \quad (8.7) \\ & \times \left(\binom{d-1}{2} p_0^2 + p_0^3 \text{pol}(p_0) \right)^{j-1} \\ & \times (1 + p_0 \text{pol}(p_0))^{d-j} \\ & = \binom{d-1}{2}^{j-1} \binom{d}{1} \binom{d-1}{j-1} p_0^{2j-1} + p_0^{2j} \text{pol}(p_0) \end{aligned}$$

De forma similar obtemos que

$$\begin{aligned} \mathbb{P}[v \in \widehat{Y}_{j,a}^0(p_0)] & = (1-p_0) \binom{d}{j} (1-p_0)^d \quad (8.8) \\ & \times \left(1 - (1-p_0)^{d-1} - \binom{d-1}{1} p_0 (1-p_0)^{d-2} \right)^j \\ & \times \left((1-p_0)^{d-1} + \binom{d-1}{1} p_0 (1-p_0)^{d-2} \right)^{d-j} \\ & = \binom{d-1}{2}^j \binom{d}{j} p_0^{2j} + p_0^{2j+1} \text{pol}(p_0) \end{aligned}$$

As igualdades para $\mathbb{P}[v \in \widehat{Y}_{j,c}^0(p_0)]$ acima fornecem prontamente o item (a).

Com (a) verdadeiro e supondo que (b) seja verdadeiro, o item (c) é imediato.

Para provar a concentração do item (b) usamos a Desigualdade de Chebyshev. Seja $\delta > 0$. Então

$$\begin{aligned} \mathbb{P}[|\hat{z}_{j,c}^0(p_0) - \mathbb{E}[\hat{z}_{j,c}^0(p_0)]| \geq \delta] &\leq \frac{1}{\delta^2} \text{Var} [\hat{z}_{j,c}^0(p_0)] = \frac{1}{\delta^2} \text{Var} \left[\frac{\hat{Y}_{j,c}^0(p_0)}{n} \right] \\ &= \frac{1}{(n\delta)^2} \sum_{u \in V(G)} \text{Var} [1_{u \in \hat{Y}_{j,c}^0(p_0)}] + \frac{1}{(n\delta)^2} \sum_{u_1 \neq u_2} \text{Cov} [1_{u_1 \in \hat{Y}_{j,c}^0(p_0)}, 1_{u_2 \in \hat{Y}_{j,c}^0(p_0)}], \end{aligned}$$

onde o termo da esquerda converge para zero quando $n \rightarrow \infty$ pois

$$\left| \text{Var} [1_{u \in \hat{Y}_{j,c}^0(p_0)}] \right| \leq 1.$$

Para provar que o termo da direita converge para zero, observe que

$$\left| \text{Cov} [1_{u_1 \in \hat{Y}_{j,c}^0(p_0)}, 1_{u_2 \in \hat{Y}_{j,c}^0(p_0)}] \right| \leq 1$$

e que esta covariância é igual a zero sempre que $1_{u_1 \in \hat{Y}_{j,c}^0(p_0)}$ for independente de $1_{u_2 \in \hat{Y}_{j,c}^0(p_0)}$. Mas essas variáveis indicadoras serão independentes sempre que a distância entre u_1 e u_2 for maior do que 4. Por outro lado o número de vértices a distância menor ou igual a 4 de um $u \in V(G)$ específico é no máximo d^4 . Portanto, o número de pares (u_1, u_2) com distância menor ou igual a 4 é no máximo $nd^4/2$. Isso completa a prova do item (b). \square

Um aspecto importante da simulação que facilita a análise é que as coordenadas da matriz $M_k(z(x))$ sejam estritamente positivas na região de interesse com exceção, possivelmente, da diagonal principal. Por isso, faremos uma breve análise dessas coordenadas. Primeiramente, vamos relembrar o significado das funções que aparecem nas coordenadas da matriz $M_k(z(x))$. Cada $f_{j,v}^{(i)}(0)$ representa a taxa de variação esperada dos vértices vermelhos de grau j no instante 0 pela operação Op_i . Essa esperança depende do número de vértices vermelhos de graus j , $j - 1$ e azuis de grau $j - 1$, sendo que os vértices de grau j vermelhos promovem o decréscimo e os outros dois tipos promovem o crescimento do número de vértices

vermelhos de grau j , independentemente da operação aplicada. Adicionalmente, a operação Op_i tem efeito de decréscimo de 1 unidade sobre vértices vermelhos de grau i . Na fase k da simulação, as curvas que representam os vértices vermelhos de grau maior que k , $z_{j,v}^i$ com $i > k$, permanecem nulas, conforme já discutimos. Isso implica que as coordenadas $f_{j,v}^{(i)}$, com $i > k$, sejam estritamente positivas exceto, possivelmente, as da diagonal principal. Uma maneira de obter isso de forma rigorosa é olhar para a expressão de $f_{j,v}^{(i)}(x)$ e notar que todos os termos negativos são multiplicados por $z_{j,v}(x)$.

Para que a nossa simulação tenha essa propriedade temos que escolher adequadamente o parâmetro p_0 da etapa de pré-processamento de forma que esses termos sejam pequenos em comparação com os termos que tornem a expressão positiva e controlar o domínio das funções f . Basta olhar para as expressões de $c_{i,k,\ell}^{d,j,c}(z(x))$ e $d_i^{d,j,c}(z(x))$ para ver que os termos positivos são multiplicados por $z_{j-1,v}(x)$ e $z_{j-1,a}(x)$ e que, em $x = 0$, estes termos podem ser tomados proporcionalmente maiores do que $z_{j,v}(x)$ tanto quanto se queira desde que p_0 seja suficientemente pequeno. Abaixo segue uma versão formal desse fato.

Consequência 8.2. *(da prova do Teorema 8.1) Existe p_0 tal que $f_{j,v}^{(i)}(0) + \delta_{i,j} > 0$ para todo $i \geq 1$ e $j \geq 2$.*

Demonstração. Das equações (8.5), (8.6) e (8.8) obtemos que

$$\lim_{p_0 \rightarrow 0} \frac{\mathbb{E}[\hat{z}_{j,v}^0(p_0)]}{\mathbb{E}[\hat{z}_{j-1,c}^0(p_0)]} = 0, \quad c \in \{a, v\}.$$

Agora, basta escolher p_0 tal que as funções $c_{i,k,\ell}^{d,j,c}(z(x))$ e $d_i^{d,j,c}(z(x))$ sejam positivas em $x = 0$. Ao olhar para as expressões dessas funções podemos concluir que isso será verdadeiro se p_0 for escolhido pequeno o suficiente tal que

$$4\mathbb{E}[\hat{z}_{j,v}^0(p_0)] < \inf\{\mathbb{E}[\hat{z}_{j-1,c}^0(p_0)] : c \in \{a, v\}\},$$

o que conclui a prova. □

No PVI (8.4) os denominadores das funções f não podem ter polos na região de seus domínios. Além disso, queremos que as coordenadas da matriz $M_k(z(x))$ que estão fora da diagonal principal sejam positivas no decorrer de todas as fases. Por isso, devemos definir um domínio que contenha em seu interior as condições iniciais resultantes do pré-processamento, e que haja uma certa tolerância para a movimentação das variáveis que ocorre no decorrer do processo sem que as coordenadas referidas deixem de ser positivas em algum momento.

Assim, seja p_0 pequeno o suficiente tal que

$$8\mathbb{E}[\hat{z}_{j,v}^0(p_0)] < \min\{\mathbb{E}[\hat{z}_{j-1,c}^0(p_0)] : c \in \{a, v\}\},$$

e seja $\gamma_{j,c} := \mathbb{E}[\hat{z}_{j,c}^0(p_0)]/2$. Note que $\gamma_{j,c}$ é uma cota inferior para $z_{j,c}$ no domínio de cada $f_{j,c}^i$ que garante a positividade de todas as funções f fora da diagonal principal da matriz quadrada da equação (8.1).

Dado $\varepsilon = \min_{j,c} \gamma_{j,c}/2$, seja

$$\Omega'(p_0) = \left\{ \begin{array}{l} -\varepsilon < x < 1, \\ (x, z(x), q(x)) \in \mathbb{R}^{2d+1} : \gamma_{j,c} < z_{j,c}(x) < 1 - \varepsilon, \quad (j, c) \in I \\ \varepsilon < q(x) < 1 - \varepsilon \end{array} \right\}.$$

Está claro que cada função $f_{j,v}^i(x) + \delta_{i,j}$ permanece positiva em $\Omega'(p_0)$.

Por outro lado, as funções α devem estar bem definidas neste domínio, o que será garantido se seu denominador $\det M_k(z(x))$ nunca for nulo nesse domínio. Assim, dado $\gamma > 0$ definimos o conjunto aberto

$$\Omega'_k(\gamma) := \{(x, z(x), q(x)) \in \mathbb{R}^{2d+1} : \det M_k(z(x)) > \gamma\}.$$

O domínio da fase k poderá ser definido como a componente conexa Ω_k do conjunto aberto $\Omega'(p_0) \cap \Omega'_k(\gamma)$ que contém as condições iniciais da fase .

Lema 8.3. *Seja $p_0 > 0$ e considere os valores $\hat{z}_{j,c}^0(p_0)$ obtidos pela etapa de pré-processamento como entradas das funções f em $x = 0$, então é verdade que*

$$f_{j,v}^{(i)}(0) - \delta_{j,i} = H_{j,v}^{(i)} p_0^{2(j-1)-1+\delta_{i,d-1}} + p_0^{2(j-1)+\delta_{i,d-1}} \text{pol}(p_0),$$

onde $H_{j,v}^{(i)} > 0$ é uma constante real e $\text{pol}(p_0)$ denota um polinômio com indeterminada p_0 .

Demonstração. Relembre que

$$f_{j,c}^{(i)}(z(x)) := \sum_{\ell=1}^{d-i} \left(a_{i,\ell}^d(z(x)) \sum_{k=0}^{d-1} b_{k,\ell}^d(z(x)) c_{i,k,\ell}^{d,j,c}(z(x)) \right) + a_{i,0}^d(z(x)) d_i^{d,j,c}(z(x)),$$

onde as funções a , b , c e d foram definidas na Seção 8.2. Note que podemos usar as equações 8.6 e 8.8 para descobrir que, nas condições iniciais, as funções a , b , c e d têm valores da forma

$$\begin{aligned} c_{i,k,\ell}^{d,j,c}(z(0)) &= (d-k-1) \left(U_1 p_0^{2(j-1)} + p_0^{2(j-1)+1} \text{pol}(p_0) \right) \\ &\quad + (d-i-\ell) \left(U_2 p_0^{2(j-1)-2} + p_0^{2(j-1)-1} \text{pol}(p_0) \right), \\ b_{k,\ell}^d(z(0)) &= \left(U_3 p_0^{2k} + p_0^{2k+1} \text{pol}(p_0) \right)^\ell, \\ a_{i,\ell}^d(z(0)) &= U_4 p_0^{d-i-\ell} + p_0^{d-i-\ell+1} \text{pol}(p_0), \\ d_i^{d,j,c}(z(0)) &= (d-i) \left(U_5 p_0^{2(j-1)} + p_0^{2(j-1)+1} \text{pol}(p_0) \right), \end{aligned}$$

onde U_1, \dots, U_5 são constantes reais positivas.

Agora, note que, no primeiro caso, em que $i \leq d-2$, a expressão de $f_{j,c}^{(i)}(z(0))$ sempre tem exatamente um termo com um fator p_0 de grau $2(j-1)-1$ e coeficiente positivo, que corresponde ao termo dos somatórios com $\ell = d-i-1$ e $k = 0$. Claramente, não há fatores p_0 com grau menor do que $2(j-1)-1$.

No segundo caso $i = d-1$. Isso implica que o segundo termo de $c_{i,k,\ell}^{d,j,c}(z(0))$ na expressão acima pode ser descartado. Com isso, os termos com fator p_0 de menor

grau tem grau $2(j - 1)$, que são referentes ao termo dos somatórios associados a $\ell = d - i$ e $k = 0$ e ao termo da direita de $f_{j,c}^{(i)}(z(0))$, que se refere a $\ell = 0$. Ambos com coeficientes positivos. \square

8.5 Estabelecendo condições para a consistência da simulação

É necessário verificar algumas hipóteses para garantir a existência e unicidade das soluções do PVI (8.4) e um comportamento adequado à análise do MED. Abaixo temos condições suficientes para garantir isso no começo do processo.

Proposição 8.4. *Existe $\gamma > 0$ tal que*

$$(1.1) \quad |\det M_1(z(0))| > \gamma.$$

$$(2.1) \quad |\alpha_i^1(0)| > \gamma \text{ para todo } 1 \leq i \leq d - 1.$$

$$(3.1) \quad (0, z(0), q(0)) \in \Omega_1.$$

O primeiro item serve para garantir a existência e a unicidade das proporções α 's nas condições iniciais e também nas proximidades destas. A segunda garante que cada α inicial esteja no interior do intervalo $(0, 1)$ (pois a soma desses α 's é igual a 1). A terceira garante que as funções f não tenham polos inicialmente.

Demonstração. (Proposição 8.4). Como vimos antes, o número $f_{j,v}^{(i)}(0)$ depende do número de vértices vermelhos de graus j , $j - 1$ e azuis de grau $j - 1$, sendo que os vértices vermelhos de grau j causam um efeito de decrescimento e os outros dois tipos causam crescimento dos vermelhos de grau j , independentemente da operação aplicada. Adicionalmente, a operação Op_i tem efeito de decrescimento

de 1 unidade permanente sobre vértices vermelhos de grau i . Por outro lado, se desconsiderarmos esse efeito das Op_i o Lema 8.3 permite afirmar que as expressões $f_{j,v}^{(i)} + \delta_{i,j}$ não são substancialmente diferentes para um mesmo j .

No instante $t = 0$ essas quantidades são fornecidas pela etapa de pré-processamento definida anteriormente e podem ser concentradas tão próximas de zero quanto desejado (escolhendo p_0 suficientemente pequeno, conforme o Teorema 8.1). Assim, não é difícil concluir que todos os elementos da matriz $M_k(z(0))$ podem ser tornados tão pequenos em módulo quanto desejados, com exceção dos elementos da primeira linha e da diagonal principal, que terão valores absolutos próximos de 1.

Por outro lado a matriz

$$\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 0 & -1 & 0 & \cdots & 0 \\ 0 & 0 & -1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{bmatrix} \quad (8.9)$$

tem o valor absoluto do seu determinante igual a 1. Como o determinante de uma matriz é função contínua das suas coordenadas, $M_1(z(0))$ pode ser tomada tão próxima dessa matriz quanto desejado pela escolha de um p_0 suficientemente pequeno para a etapa de pré-processamento, o que permite que ela tenha a propriedade do item (1.1).

Para obter (2.1), basta mostrar que o numerador e o denominador de (8.2) têm o mesmo sinal e que o numerador é não nulo. Com efeito, relembre da Álgebra Linear elementar que se $B = (b_{i,j})_{i,j=1}^n$ é uma matriz então o determinante de B é dado por

$$\det B = \sum_{\sigma} \operatorname{sgn}(\sigma) \prod_{i=1}^n b_{i,\sigma(i)}$$

onde σ varia entre todas as permutações de $(1, 2, \dots, n)$. A função $\text{sgn}(\sigma)$, conhecida como o *signal* da permutação, atribui o valor 1 para permutações pares, isto é, permutações que podem ser obtidas a partir de $(1, 2, \dots, n)$ por um número par de operações de transposição, e atribui o valor -1 caso contrário. Além disso, o numerador de (8.2) é o determinante da matriz dada por $M^1(z(0))$ com a i -ésima coluna substituída pelo vetor $(1, 0, 0, \dots, 0)$.

Afirmamos que o termo da transposição $(i, 2, 3, \dots, i-1, 1, i+1, \dots, d-k)$ domina a soma do determinante do numerador de $\alpha_i^1(0)$. Com efeito, lembre que a matriz do numerador têm elementos arbitrariamente pequenos com exceção da primeira linha e da diagonal principal. Note que as coordenadas de uma mesma linha, com exceção dos elementos da diagonal principal, têm a mesma ordem de grandeza (em termos de p_0). De fato, o Lema 8.3 garante que, se $1 < j < i$ ou $0 < i < j$, então $f_{j,v}^{(i)}(0)$ tem a mesma ordem de grandeza de $p_0^{2(j-1)-1+\delta_{i,d-1}}$ e, se $2 \leq i = j$, então

$$f_{i,v}^{(i)}(0) = -1 + H_{i,v}^{(i)} p_0^{2(i-1)-1+\delta_{i,d-1}} + p_0^{2(i-1)+\delta_{i,d-1}} \text{pol}(p_0),$$

onde $H_{j,v}^{(i)} > 0$ é uma constante real e $\text{pol}(p_0)$ denota um polinômio com indeterminada p_0 . Finalmente, note que a permutação $(i, 2, 3, \dots, i-1, 1, i+1, \dots, d-k)$ envolve apenas uma coordenada fora da diagonal principal e da primeira linha que, pelo Lema 8.3, é um polinômio em p_0 cujo termo com coeficiente não nulo de menor grau tem grau $2(i-1)-1$. Qualquer outra permutação envolve pelo menos duas coordenadas fora da primeira linha e da diagonal principal, sendo pelo menos uma na i -ésima linha. Isso produz um polinômio em p_0 cujo termo de menor grau tem grau maior do que $2j-3$. Assim, o sinal do $\det M_{1,i}(z(x))$ é determinado pelo sinal do termo referente à transposição $(i, 2, 3, \dots, i-1, 1, i+1, \dots, d-k)$ para p_0 suficientemente pequeno. Isso também fornece um numerador não nulo para $\alpha_i^1(0)$.

De forma análoga concluímos que o sinal do $\det M_1(z(0))$ é determinado pela permutação identidade $(1, 2, \dots, d - k)$. Finalmente, para ver que ambos têm o mesmo sinal, note que a transposição $(i, 2, 3, \dots, i - 1, 1, i + 1, \dots, d - k)$ tem sinal contrário à permutação identidade e que o termo associado a permutação identidade corresponde ao produto de coordenadas onde há exatamente uma coordenada negativa a mais do que o termo associado a transposição. Isso completa o item (2.1).

O item (3.1) segue da prova do Teorema 8.1 e da definição de Ω_1 . \square

A Proposição 8.4 garante que o processo tenha um início e que a duração do processo é não nula. Agora, podemos estender as soluções do PVI (8.4) para todo $x \in [0, x_1]$ onde x_1 é o ínfimo de todos os $x > 0$ tal que pelo menos uma das seguintes *condições* seja satisfeita.

$$(i.1) \quad (x, z(x), q(x)) \notin \Omega_1.$$

$$(ii.1) \quad \text{Para algum } 1 \leq i \leq d - 1, \text{ vale que } \alpha_i^1(x) = 0, A_{i,1} \text{ é não vazio e } \frac{d^{\nu_{i,1}} \alpha_i^1(x)}{dx^{\nu_{i,1}}} < 0, \text{ onde}$$

$$A_{i,k} := \left\{ s \geq 1 : \frac{d^s \alpha_i^k(x)}{dx^s} \neq 0 \right\} \quad \text{e} \quad \nu_{i,k} := \inf A_{i,k}.$$

Esse x_1 está bem definido por que a condição (i.1) tem que ser satisfeita em tempo finito. Com efeito, em cada passo, pelo menos um vértice (vermelho) é subtraído da soma de vértices de grau $\leq d - 1$. Como o número total de vértices é finito, alguma classe de vértices deve se anular. Isso se traduz em termos analíticos da seguinte forma. A soma das variáveis z tem derivada negativa (menor do que -1) e como cada z é menor do que a soma de todas elas, alguma curva $z_{j,c}(x)$ deve se anular em algum $x \leq 1$. Assim o PVI (8.4) está bem definido em $[0, x_1]$ e x_1 será o final da fase 1.

Se a condição (i.1) é satisfeita em x_1 , a fase 1 é dita a fase *final* do processo.

Afirmamos que, se apenas a condição (ii.1) é satisfeita em x_1 , então o PVI (8.4) satisfaz as hipóteses do Teorema 4.2 em Ω_1 . Com efeito, a Hipótese de limitação vale com $\beta(n) := d^2$ e $\gamma(n) = 0$. A Hipótese de tendência vale para as funções à direita das igualdades da equação (8.4) porque é válida para cada $f_{j,c}^{(i)}$ (isso foi verificado no momento em que determinamos as expressões das funções f). Quanto à Hipótese de Lipschitz, note que as funções $f_{j,c}^{(i)}$ são Lipschitz em Ω_1 devido a própria definição desse conjunto. Resta verificar que as proporções α são todas Lipschitz nesse domínio. De fato, pela equação (8.2), cada α é uma função racional contínua das funções f , onde o $\det M_1(z(x))$ não se anula, pois estamos supondo que a condição (i.1) não é satisfeita no intervalo fechado $[0, x_1]$, de forma que $|\det M_1(z(x))| > \gamma$ para algum $\gamma > 0$ no intervalo fechado $[0, x_1]$. Isso conclui a verificação da Hipótese de Lipschitz. O fato de a condição (ii.1) ser a única restrição ativa no final da fase 1 implica que o determinante da matriz $M_1(z(x))$ (referente à fase 1) está afastado de 0. Para que a fase 2 comece e se mantenha pelo menos em um pequeno intervalo, é necessário que, no início da fase 2, o determinante da matriz (referente à fase 2) esteja afastado de 0. Para demonstrar isso, não basta apenas dizer que o determinante está afastado de 0 ao final da fase 1, pois a matriz na fase 2 mudou para $M_2(z(x))$, logo ainda temos que argumentar que esse determinante está afastado de 0 de alguma forma.

Como mencionamos antes, estamos supondo que (ii.1) deve ser satisfeita em x_1 . Neste caso afirmamos o seguinte.

Proposição 8.5. *Se apenas a condição (ii.1) é satisfeita em x_1 , então $\alpha_1^1(x_1) = 0$ e $\alpha_i^1(x_1) > 0$ para todo $i = 2, 3, \dots, d - 1$.*

Demonstração. De fato, nenhuma das proporções α é negativa em $[0, x_1]$ pela definição de x_1 . Além disso, se para algum $i = 2, 3, \dots, d-1$ for verdade que $\alpha_i^1(x) = 0$ em algum ponto de $[0, x_1]$, será necessário que

$$\left(\sum_{k=1}^{i-1} \alpha_k^1(x) f_{i,v}^{(k)}(z(x)) \right) + \alpha_i^1(x) f_{i,v}^i(z(x)) + \left(\sum_{k=i+1}^{d-1} \alpha_k^1(x) f_{i,v}^{(k)}(z(x)) \right) > 0,$$

pois o termo central é nulo e o que sobra é uma combinação convexa de quantidades estritamente positivas. Mas isso contradiz o fato de que essas proporções α são uma solução da equação matricial (8.1), o que deveria satisfazer que

$$\left(\sum_{k=1}^{i-1} \alpha_k^1(x) f_{i,v}^{(k)}(z(x)) \right) + \alpha_i^1(x) f_{i,v}^i(z(x)) + \left(\sum_{k=i+1}^{d-1} \alpha_k^1(x) f_{i,v}^{(k)}(z(x)) \right) = 0,$$

□

A Proposição 8.5 permite que saibamos que $\alpha_1^1(x_1)$ é o único α nulo quando houver a possibilidade de uma segunda fase.

O problema de valor inicial associado à fase $k \geq 2$ é o seguinte.

$$\begin{aligned} \frac{dz_{j,c}(x)}{dx} &= \sum_{i=k}^{d-1} \alpha_i^k(x) f_{j,c}^{(i)}(z(x)), \\ \frac{dq(x)}{dx} &= \sum_{i=k}^{d-1} \alpha_i^k(x) f_q^{(i)}(z(x)), \\ q(0) &= \hat{q}^{k-1}, \quad z_{j,c}(0) = \hat{z}_{j,c}^{k-1}, \quad (j, c) \in I, \end{aligned} \tag{8.10}$$

onde \hat{q}^{k-1} e $\hat{z}_{j,c}^{k-1}$ são iguais aos $q(x_{k-1})$ e $z_{j,c}(x_{k-1})$ da fase $k-1$, respectivamente. Note que fizemos uma translação na variável tempo x de forma que as condições iniciais da fase 2 fiquem na origem $x = 0$ (ao invés do ponto $x = x_{k-1}$). Isso será feito para as demais fases e serve para que cada fase esteja no formato do Teorema 4.2. Isso não afeta o desempenho final porque as funções f não dependem diretamente de x , mas sim dos valores das variáveis z . O domínio Ω_k pode ser redefinido para outro valor de $\gamma > 0$ se necessário.

As hipóteses que garantem uma duração mínima para a fase k e as condições de término desta são análogos às da fase 1. A diferença está na forma como as hipóteses de início são verificadas, pois dependem da situação na qual a fase anterior terminou. Para que se inicie a fase $k \geq 2$ é suficiente que exista $\gamma > 0$ tal que

$$(1.k) \quad |\det M_k(z(0))| > \gamma.$$

$$(2.k) \quad |\alpha_i^k(0)| > \gamma \text{ para todo } k \leq i \leq d-1.$$

$$(3.k) \quad (0, z(0), q(0)) \in \Omega_{M_k}.$$

O PVI (8.10) se estende para todo $x \in [0, x_k]$ onde x_k é o ínfimo de todos os $x > 0$ tal que pelo menos uma das seguintes condições seja satisfeita.

$$(i.k) \quad (x, z(x), q(x)) \notin \Omega_{M_k}.$$

$$(ii.k) \quad \alpha_k^k(x) = 0, A_{k,k} \text{ é não vazio e } \frac{d^{\nu_{k,k}} \alpha_k^k(x)}{dx^{\nu_{k,k}}} < 0$$

Note que o item $(ii.k)$ foi simplificado em relação a $(ii.1)$, pois, da Proposição 8.6 a seguir, para que algum outro α se anule antes de $\alpha_2^2(x)$, é preciso que $\det M_k(z(x)) = 0$ o que ativaria o item $(i.1)$.

Proposição 8.6. *Se, em x_{k-1} , $(ii.k-1)$ é satisfeita e $(i.k-1)$ não é satisfeita, então $\alpha_{k-1}^{k-1}(x_1) = 0$ e $\alpha_i^{k-1}(x_1) > 0$ para todo $i = k, 3, \dots, d-1$.*

Isso prova que as mudanças de fase ocorrem precisamente na sequência: fase 1, fase 2, fase 3 ...

Se apenas a condição $(ii.k-1)$ for satisfeita em x_{k-1} , então podemos considerar o início da fase k . Para determinar se a fase k tem duração não nula não é

necessário verificar todas as condições de início, só precisamos verificar (1.k). Os itens (2.k) e (3.k) serão consequência. A proposição que garante isso segue abaixo.

Proposição 8.7. *Se (i.k - 1) não é satisfeita, e (iii.k - 1) e (1.k) são satisfeitos, então os itens (2.k) e (3.k) são satisfeitos. Portanto existe uma fase k com duração não nula e x_k pode ser definido.*

Demonstração. Como a condição (i.k - 1) não foi satisfeita no final da fase $k - 1$, então $(0, z(0), q(0)) \in \Omega_k$, fornecendo o item (3.k).

Para provar que (2.k) é verdadeiro, lembre que o vetor

$$(\alpha_{k-1}^{k-1}(x_k), \alpha_k^{k-1}(x_{k-1}), \dots, \alpha_{d-1}^{k-1}(x_{k-1}))$$

é solução da equação matricial (8.1) na versão para $k - 1$ no ponto $z(x) = z(x_{k-1})$.

Como $\alpha_{k-1}^{k-1}(x_{k-1}) = 0$ o vetor

$$(\alpha_k^{k-1}(x_{k-1}), \alpha_{k+1}^{k-1}(x_{k-1}), \dots, \alpha_{d-1}^{k-1}(x_{k-1}))$$

também é solução da equação (8.1) mas desta vez na versão para k e no mesmo ponto $z(x) = z(x_{k-1})$ (pois, da versão $k - 1$ para k , tudo o que se faz é excluir a primeira coluna e a segunda linha). Agora, como estamos supondo (1.k) verdadeiro, essa solução deve ser única, de forma que

$$\alpha_i^k(0) = \alpha_i^{k-1}(x_{k-1}), \quad i = k, 3, \dots, d - 1.$$

onde as limitações $0 < \alpha_i^k(x_{k-1}) < 1$ seguem da Proposição 8.6. Isso prova (2.k).

□

Até aqui, vimos que a fase 1 é a fase inicial e sempre existe por causa das propriedades das saídas da etapa de pré-processamento. Também vimos que, para que exista uma fase $k \geq 2$, é suficiente que exista a fase $k - 1$ e que as seguintes condições sejam satisfeitas:

(a) $\det M_k(z(x_{k-1})) \neq 0$, isto é, $(1, k)$ é satisfeita;

(b) $(x_{k-1}, z(x_{k-1}), q(x_{k-1})) \in \Omega_{M_{k-1}}$, isto é, $(i, k-1)$ não é satisfeita;

A última fase r será definida como o menor r tal que (i, r) for satisfeita ou $(1, r+1)$ não for satisfeita.

A concentração obtida dessa simulação para esse x_r final se deve, como já mencionado antes, à estabilidade das funções do nosso PVI com relação às condições iniciais. Essa estabilidade é garantida pela Condição de Lipschitz em cada domínio Ω_{M_k} .

Note que o tamanho do conjunto dominante construído no final do processo é igual à soma das durações das fases x_1, x_2, \dots, x_r com o tamanho do conjunto de vértices não dominados no final da fase r , $1 - q(x_r)$. Isso conclui a prova do Teorema 3.11.

Testes computacionais sugerem que a última fase que pode ser iniciada é a fase $d-2$. Em todos os testes a condição que encerrou a última fase foi que a quantidade de vértices do tipo que estava sendo consumido, os vértices vermelhos de grau $d-2$, se anulou antes que α_{d-2}^{d-2} ou $\det M_{d-2}(z(x))$ se anulassem, o que caracterizou a ativação da condição $(i, d-2)$.

Simulações numéricas foram incluídas no apêndice para $3 \leq d \leq 6$ e os casos $d=3$ e $d=5$ estão comentados.

8.6 Prova do Teorema 3.12

Resta interpretar o Algoritmo 8(c) como um algoritmo local de deleção. Este algoritmo tem duas cores transientes, azul e vermelho, e duas cores de saída, verde

e amarelo. O algoritmo tem como entrada um grafo colorido pela etapa de pré-processamento ou pela fase anterior, obedecendo a seguinte coloração. Vértices 2-dominados ou do conjunto 2-dominantes têm a cor verde. Vértices que têm todos os vizinhos 2-dominados com no máximo um vizinho no conjunto 2-dominante recebem a cor amarela. Vertices que não se enquadram nos critérios das cores verde e amarela recebem cor azul se não têm nenhum vizinho no conjunto 2-dominante e recebem a cor vermelha se têm um vizinho no conjunto 2-dominante. Se estamos processando a fase k , a regra de seleção escolhe um vértice vermelho v_t seguindo as probabilidades e procedimentos descritos nas linhas 5, 6 e 7 do Algoritmo 8(c), o que claramente atende os requisitos da regra de seleção. A regra de exploração local opera da seguinte forma. O algoritmo identifica todas as adjacências dos vizinhos de v_t . No primeiro caso, v_t possui vizinho azul. Neste caso, o algoritmo revela um vizinho azul u_t dentre os vizinhos azuis de menor grau e identifica todas as adjacências de u_t . Depois, revela todos os vértices correspondentes às adjacências de u_t que foram identificadas com a cor vermelha e identifica todas as adjacências desses vértices. No segundo caso, quando v_t não tem vizinho azul. Neste caso, o algoritmo revela todos os vizinhos de v_t e identifica todas as adjacências dos vértices revelados, o que encerra a regra de exploração local de raio $r = 2$. A função de recoloração atribui cor verde para todos os vértices revelados pela regra de exploração local. Se v_t têm um vizinho azul revelado, esse vértice, que precisa ser o vértice u_t da regra de exploração local recém aplicada, tem seus vizinhos identificados com a cor azul e grau menor do que $d - 1$ alteradas para vermelho. Para os vértices que foram identificados com grau $d - 1$ mas não revelados pela regra de exploração local, a função de recoloração os revela e atribui cor amarela. Isso encerra nossa interpretação. O Teorema 3.12 segue após $d - 2$ aplicações do Teorema 5.8 ao nosso algoritmo.

9 CONCLUSÃO

Neste trabalho, usamos o Método das Equações Diferenciais para analisar o desempenho de algoritmos locais que produzem conjuntos dominantes específicos em grafos aleatórios d -regulares. Estudamos os parâmetros de dominância total e 2-dominância, que ainda não haviam sido explorados com esse método, e o desempenho obtido pela análise dos algoritmos forneceu cotas superiores probabilísticas para esses parâmetros que são consideravelmente melhores do que aquelas que estavam disponíveis na literatura até então, que foram enunciadas para todos os grafos ou para grafos que satisfazem uma certa propriedade.

Além disso, uma revisão da teoria desenvolvida em Hoppen e Wormald [36] permitiu traduzir as cotas obtidas nesse contexto probabilístico para cotas determinísticas válidas para todos os grafos d -regulares com cintura suficientemente grande. Assim, enunciamos dois teoremas para cada resultado obtido, um para grafos com cintura suficientemente grande e outro para grafos aleatórios.

Como uma complementação de nossos resultados, provamos na Seção 3.4.1 uma desigualdade entre cotas para os dois tipos de resultado. Essa desigualdade é importante porque garante que qualquer cota superior para um resultado na versão para a cintura grande também é cota superior para o respectivo resultado na versão para grafos aleatórios, independentemente da origem da cota. No entanto, cabe destacar que a teoria de Hoppen e Wormald só garante que a cota superior de um resultado na versão para grafos aleatórios seja também cota superior para um resultado na versão para cintura grande se a cota foi obtida pela análise de um algoritmo local, como feito em nosso trabalho.

No caso da dominância total, os dois resultados mencionados são os Teoremas 3.7 e 3.8, demonstrados no Capítulo 5, que correspondem, respectivamente, à versão para grafos aleatórios e à versão para a cintura grande. Resultados parciais desta pesquisa foram apresentados na conferência da Society for Industrial and Applied Mathematics (SIAM) [35] e os resultados completos foram apresentados no X Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS 2019). Está previsto que o resumo expandido seja publicado em uma edição especial da revista *Electronic Notes in Theoretical Computer Science*. Os resultados foram disponibilizados via Internet (Veja [34]). Em breve, também esperamos submeter a versão completa à publicação.

O parâmetro de 2-dominância contou com resultados derivados de dois algoritmos distintos, onde cada um destes algoritmos gerou dois teoremas, um para grafos aleatórios e outro para grafos com cintura grande. Os Teoremas 3.9 e 3.10 correspondem, respectivamente, às versões para grafos aleatórios e para grafos com cintura grande do primeiro algoritmo e foram provados no Capítulo 6. O algoritmo que gerou estes resultados é bem simples e permitiu que as cotas destes teoremas sejam dadas como uma função algébrica explícita que depende apenas de d .

Com relação ao segundo algoritmo, os Teoremas 3.11 e 3.12 representam, respectivamente as versões para grafos aleatórios e para grafos com cintura grande e foram provados no Capítulo 8. O algoritmo que gerou essas cotas é mais elaborado, usa prioridades e tem cotas melhores do que as obtidas no primeiro algoritmo. Neste caso, as cotas não puderam ser obtidas explicitamente e dependem de implementações computacionais para obter aproximações de seus valores. Algumas dessas aproximações foram apresentadas na Tabela 3.4. Pretendemos enviar os resultados do Capítulo 6 e 8 para publicação em um periódico. Os resultados do

Capítulo 6 também foram mencionados na conferência da Society for Industrial and Applied Mathematics [35].

Acreditamos que as técnicas estudadas nesse trabalho permitem abordar uma grande variedade de problemas em Teoria dos Grafos e em processos aleatórios em geral. Essas técnicas permitem encontrar cotas probabilísticas que geralmente são bem melhores do que as cotas gerais obtidas com o mesmo algoritmo. Isso ocorre porque as cotas gerais se baseiam no pior caso possível de aplicação do algoritmo, enquanto que o método das equações diferenciais se baseia nos valor médio das quantidades que o algoritmo retorna.

No caso dos parâmetros estudados nesse trabalho, não encontramos, na literatura, cotas inferiores para quantidades como $\gamma_t^R(d)$ e $\gamma_t^g(d, \infty)$ além das cotas inferiores triviais das equações (3.1) e (3.3). Potencialmente, algoritmos mais complicados poderiam melhorar as cotas que obtivemos, no entanto não se deve esperar que as melhores cotas possíveis (isto é, determinar o valor de quantidades como $\gamma_t^R(d)$ e $\gamma_t^g(d, \infty)$, definidas na Seção 3.4.1) possam ser encontradas dessa forma. Conforme já mencionamos, há trabalhos recentes, como o de Gamarnik e Sudan [25], onde se prova que, se d é suficientemente grande, os algoritmos locais não são capazes de fornecer a melhor cota para conjuntos independentes máximos, o que sugere que o mesmo seja válido para para valores de d pequenos e para outros parâmetros da teoria dos grafos. Além disso, Bayati, Gamarnik e Tetali [5] provaram que, no caso de conjuntos independentes para grafos d -regulares, a proporção de vértices em um conjunto independente máximo é assintoticamente quase certamente igual a uma constante α_d . Se um resultado semelhante valesse para parâmetros de dominância, teríamos que $\gamma_t^R(d)$ seria a.q.c. a proporção de vértices em um conjunto totalmente dominante mínimo.

É importante mencionar que há vários parâmetros da teoria dos grafos que ainda não foram estudados com as técnicas abordadas aqui, por exemplo, as noções de k -dominância e de k -dominância total. Não é difícil estender os algoritmos da 2-dominância tratados aqui para versões de k -dominância para $k > 2$, simplesmente pela adição de mais cores ou mais fases. As técnicas estudadas aqui certamente se aplicariam a essa situação, e os resultados dependem do estudo de soluções de um sistema de equações diferenciais específico. No futuro, podemos trabalhar com outros parâmetros da Teoria dos Grafos para grafos d -regulares, além daqueles voltados para dominância.

Vale também ressaltar que o Método das Equações Diferenciais foi usado em muitos contextos, como o da resiliência ao contágio de redes financeiras por Amini, Cont e Minca [3] ou mesmo no trabalho Griffiths e Morris [43] que não trata especificamente de grafos d -regulares como o nosso. Além disso, esse método já recebeu aprimoramentos voltados para a questão determinística, como o Trabalho de Wormald e Hoppen [36]. Ele é capaz de estudar processos aleatórios em grafos em geral de maneira eficiente e tem se mostrado uma ferramenta muito útil e promissora. Acreditamos que esse estudo permita abordar problemas de outra natureza no futuro.

Referências Bibliográficas

- [1] N. Alon e J. Spencer, *The Probabilistic Method*, Wiley Series in Discrete Mathematics and Optimization, Wiley, 2011.
- [2] H. Amini, Bootstrap percolation and diffusion in random graphs with given vertex degrees, *Electronic Journal of Combinatorics*, 17(1):25 pp, 2010.
- [3] H. Amini, R. Cont e A. Minca, Resilience to contagion in financial networks, *Mathematical Finance*, 26(2):329-365, 2016.
- [4] D. Archdeacon, J. Ellis-Monaghan, D. Fischer, D. Froncek, P. C. B. Lam, S. Seager, B. Wei e R. Yuster, Some remarks on domination. *J. Graph Theory*, 46:207-210, 2004.
- [5] M. Bayati, D. Gamarnik e P. Tetali, Combinatorial approach to the interpolation method and scaling limits in sparse random graphs, *The Annals of Probability*, 41(6):4080-4115, 2013.
- [6] E. A. Bender e E. R. Canfield, The asymptotic number of non-negative integer matrices with given row and column sums, *Journal of Combinatorial Theory, Series A*, 24:296-307, 1978.
- [7] A. Backhausz e B. Szegedy, On large-girth regular graphs and random processes on trees, *Random Structures and Algorithms*, 53(3):389-416, 2018.
- [8] S. Bermudo, J. C. Hernández-Gómez e J. M. Sigarreta, On the total k -domination in graphs, *Discussiones Mathematicae Graph Theory*, 38(1):301-317, 2018.

- [9] T. Bohman e P. Keevash, Dynamic concentration of the triangle-free process. *The Seventh European Conference on Combinatorics, Graph Theory and Applications*, Edizioni della Normale, Pisa, 489-495, 2013.
- [10] B. Bollobás, A probabilistic proof of an asymptotic formula for the number of labelled regular graphs, *European Journal of Combinatorics*, 1:311-316, 1980.
- [11] B. Bollobás, *Random Graphs*, Academic Press, London, 1985.
- [12] B. Bollobás e E. J. Cockayne, Graph-theoretic parameters concerning domination, independence, and irredundance. *Journal of Graph Theory*, 3:241-249, 1979.
- [13] G. Brightwell, S. Janson e M. Łuczak, The greedy independent set in a random graph with given degrees. *Random Structures and Algorithms*, 51(4):565-586, 2017.
- [14] C. Bujtás e S. Jaskó, Bounds on the 2-domination number. *Discrete Applied Mathematics*, 242:4-15, 2018.
- [15] Y. Caro e Y. Roditty, A note on the k-domination number of a graph. *International Journal of Mathematics and Mathematical Sciences*, 13:205-206, 1990.
- [16] W. Chen, D. Gamarnik, D. Panchenko e M. Rahman, Suboptimality of local algorithms for a class of max-cut problems, *The Annals of Probability*, 47(3):1587-1618, 2019.
- [17] E. J. Cockayne, R. Dawes e S. T. Hedetniemi, Total domination in graphs, *Networks*, 10(3):211-219, 1980.

- [18] J. Díaz, M. Serna e N. C. Wormald, Computation of the bisection width for random d -regular graphs, *Lecture Notes in Computer Science, LATIN 2004: Theoretical informatics*, 49-58, 2004.
- [19] N. A. Cook, Discrepancy properties for random regular digraphs, *Random Structures and Algorithms*, 50:23-58, 2016.
- [20] M. Dorfling e M. A. Henning, Transversals in 5-uniform hypergraphs and total domination in graphs with minimum degree five, *Quaestiones Mathematicae*, 38(2):155-180, 2015.
- [21] W. Duckworth e B. Mans, Randomized greedy algorithms for finding small k -dominating sets of random regular graphs, *Random Structures and Algorithms*, 27(3):401-412, 2005.
- [22] W. Duckworth e N. C. Wormald, On the independent domination number of random regular graphs, *Combinatorics, Probability and Computing*, 15:513-522, 2006.
- [23] P. Erdős, Graph theory and probability, *Canadian Journal of Mathematics* 11:34-38, 1959.
- [24] P. Erdős e A. Rényi, On Random Graphs I., *Publicationes Mathematicae*, 6:290-297, 1959.
- [25] D. Gamarnik e M. Sudan, Limits of local algorithms over sparse random graphs, *Proceedings of the 5th Conference on Innovations in Theoretical Computer Science*, ACM, 369-376, 2014.
- [26] H. Grötzsch, Ein dreifarbensatz für dreikreisfreie netze auf der kugel, *Wiss. Z. Martin-Luther-Univ. Halle-Wittenberg Math.-Natur. Reihe*, 8:109-120, 1959.

- [27] A. Hansberg e L. Volkmann, Upper bounds on the k -domination number and the k -Roman domination number, *Discrete Applied Mathematics*, 157:1634-1639, 2009.
- [28] T. Haynes, S. T. Hedetniemi e P. Slater, *Fundamentals of domination in graphs*. CRC Press, 1998.
- [29] S. T. Hedetniemi e R. C. Laskar, Bibliography on domination in graphs and some basic definitions of domination parameters, *Annals of Discrete Mathematics*, 48: 257-277, 1991.
- [30] M. A. Henning, Graphs with large total domination number. *Journal of Graph Theory* 35(1):21-45, 2000.
- [31] M. A. Henning e A. Yeo. *Total domination in graphs*. Springer, 2013.
- [32] M. A. Henning e A. Yeo. Total domination in graphs with given girth, *Graphs and Combinatorics* , 24(4):333-348, 2008.
- [33] C. Hoppen, *Properties of graphs with large girth*, Doctoral thesis, University of Waterloo, 2008.
- [34] C. Hoppen e G. Mansan, Total Domination in Regular Graphs, *e-print arXiv:1812.03560*, 2018.
- [35] C. Hoppen e G. Mansan, Domination in Random Regular Graphs and in Graphs with Large Girth, *Proceedings of the SIAM Conference on Discrete Mathematics 2018 (DM18)*, p. 14, 2018.
- [36] C. Hoppen e N. C. Wormald, Local algorithms, regular graphs of large girth, and random regular graphs. *Combinatorica*, 38(3):619-664, 2018.

- [37] C. Hoppen e N. Wormald, Properties of regular graphs with large girth via local algorithms, *Journal of Combinatorial Theory, Series B*, 121:367-397, 2016.
- [38] S. Janson, T. Łuczak e A. Ruciński. *Random Graphs*, Wiley Series in Discrete Mathematics and Optimization, Wiley, 2011.
- [39] R. M. Karp, Reducibility among combinatorial problems, *Complexity of computer computations*, 85-103, Springer, 1972.
- [40] J. H. Kim, The Ramsey number $R(3, t)$ has order of magnitude $t^2/\log(t)$, *Random Structures and Algorithms*, 7(3):173-207, 1995.
- [41] E. L. Lima, *Análise real*, IMPA, 2004.
- [42] J. Pfaff, R. C. Laskar e S. T. Hedetniemi. NP-completeness of total and connected domination and irredundance for bipartite graphs. *Technical Report 428*, Dept. Math. Sciences, Clemson University, 1983.
- [43] G. Fiz Pontiveros, S. Griffiths e R. Morris, The triangle-free process and the Ramsey number $R(3, k)$, *e-print arXiv:1302.6279*, 2013.
- [44] M. Rahman e B. Virág, Local algorithms for independent sets are half-optimal, *Annals of Probability*, 45(3):1543-1577, 2017.
- [45] R. W. Robinson e N. C. Wormald, Almost all regular graphs are Hamiltonian, *Random Structures and Algorithms*, 5(2):363-374, 1994.
- [46] A. Ruciński, N. C. Wormald, Random graph processes with degree restrictions, *Combinatorics, Probability and Computing*, 1(2):169-180, 1992.

- [47] L. Shi e N. C. Wormald, Colouring random 4-regular graphs, *Combinatorics, Probability and Computing* 16(2):309-344, 2007.
- [48] A. Shiryayev and S. Wilson, *Probability*, Graduate Texts in Mathematics, Springer, New York, 2013.
- [49] L. Sun, An upper bound for the total domination number, *J. Beijing Inst. Tech*, 4:111-114, 1995.
- [50] S. Thomassé e A. Yeo, Total domination of graphs and small transversals of hypergraphs, *Combinatorica*, 27:473-487, 2007.
- [51] N. C. Wormald, Analysis of greedy algorithms on graphs with bounded degrees, *Discrete Mathematics*, 273(1-3):235-260, 2003.
- [52] N. C. Wormald, *Models of random regular graphs. London Mathematical Society Lecture Note Series*, 267:239-298, 1999.
- [53] N. C. Wormald. *Some Problems in the Enumeration of Labelled Graphs*, Doctoral thesis, Newcastle University, 1978.
- [54] N. C. Wormald, The asymptotic distribution of short cycles in random regular graphs, *Journal of Combinatorial Theory, Series B*, 31:168-182, 1981.
- [55] N. C. Wormald, The diferential equation method for random graph processes and greedy algorithms. *Lectures on approximation and randomized algorithms*, 73-155, 1999.

Apêndice A EXPERIMENTOS COMPUTACIONAIS

Conforme mencionamos, o problema de valor inicial associado ao Algoritmo 8(c), cuja análise gerou os Teoremas 3.11 e 3.12, não foi resolvido analiticamente, mas aproximações para suas soluções foram obtidas computacionalmente e forneceram a Tabela 3.4. Nossos dados foram obtidos pela aplicação do método de Euler. Métodos de aproximação mais elaborados e precisos para soluções de equações diferenciais não foram necessários devido ao fato das funções f terem constantes de Lipschitz baixas, o que facilita a convergência. Por completude, incluiremos a seguir as linhas de comando, que foram executadas no programa Maple na versão 2015.1. Além disso, para o algoritmo com prioridades, apresentaremos experimentos computacionais que levaram aos valores aproximados obtidos nas tabelas que apresentemos no texto.

```
> with(LinearAlgebra):
> with(linalg):
> delta:=proc(i,j);
  if i=j then 1;
  else 0;
  end if;
end:
> deltam:=proc(i,j);
  if i<j then 0;
  else 1;
  end if;
end:
> Bin:=proc(d) local Q,i,k;
  Q:=Array(1..d+1,1..d+1,0);
  for k from 0 to d do
  for i from 0 to d-k do
  Q[i+1,k+1]:=binomial(d-k,i);
  end do;
  end do;
  Q;
end:
```

```

> Drop:=proc(d,M) local drop,j,i,r;
drop:=Array(1..d,1..d+4,0);
for j from 0 to d do
drop[1,j+2]:=drop[1,j+1]+M[d-j+2,1]*j;
drop[1,d+3]:=drop[1,d+3]+M[d-j+2,2]*j;
end do;
drop[1,d+4]:=drop[1,d+3]+drop[1,d+2];
for r from 1 to d+4 do
for i from 2 to d do
drop[i,r]:=drop[i-1,r]*drop[1,r];
end do
end do;
drop;
end:

> Dife:=proc(d,j,t,M,Dr) local q;
q:=((d-j+1)*M[j+1,t]-(d-j)*M[j+2,t])/Dr[1,d+4];
q;
end:

> Difejt:=proc(d,M,Dr) local Q,j,m,t;
Q:=Array(1..d+1,1..2,0);
for t from 1 to 2 do
for j from 0 to d do
Q[j+1,t]:=Dife(d,j,t,M,Dr)
end do;
end do;
Q;
end:

> Aux:=proc(d,M) local q,m;
q:=0;
for m from 0 to d-2 do
q:=q+(d-m)*(d-m-1)*M[m+2,2];
end do;
q;
end:

```

```

> Aux1:=proc(d,j,t,M,Dr,difejt,au) local q;
q:=((d-j+1)*M[j+1,1]*delta(t,2)-(d-j)*M[j+2,t]+au*difejt[j+1,t])
/Dr[1,d+4];
q;
end:
> Aux1jt:=proc(d,M,Dr,difejt,au) local Q,j,m,t;
Q:=Array(1..d+1,1..2,0);
for t from 1 to 2 do
for j from 0 to d do
Q[j+1,t]:=Aux1(d,j,t,M,Dr,difejt,au)
end do;
end do;
Q;
end:
> Aux2:=proc(d,j,t,i,k,Dr,difejt,aux1jt,M) local n,q;
q:=0;
for n from 0 to d-1 do
q:=q+(Dr[i,d+2-n]-Dr[i,d+1-n])*(-delta(j,n)*delta(t,1)+(d-n-1)*
aux1jt[j+1,t]+(d-k-i)*((d-j+1)*M[j+1,t]-(d-j)*M[j+2,t])/Dr[1,d+3]
*delta(t,2)+(i-1)*((d-j+1)*M[j+1,t]-(d-j)*M[j+2,t])/(Dr[1,d+2-n])
*delta(t,1)*deltam(j-1,n)+(i-1)*(-(d-j)*M[j+2,t])/(Dr[1,d+2-n])
*delta(t,1)*delta(j,n));
end do;
q;
end:
> Aux3:=proc(d,j,t,k,Dr,difejt,aux1jt,Bin,M) local i,q;
if k<d-1 then
q:=Aux2(d,j,t,d-k,k,Dr,difejt,aux1jt,M);
for i from 1 to d-k-1 do;
q:=q+Bin[i+1,k+1]*Dr[d-k-i,d+3]*Aux2(d,j,t,i,k,Dr,difejt,aux1jt,
M)
end do;
else q:=Bin[1+1,k+1]*Aux2(d,j,t,1,k,Dr,difejt,aux1jt,M);
end if;
q;
end:
> Fix:=proc(d,j,t,M,k,Dr,difejt,au) local q;
if k<d-1 then
q:=Dr[d-k-1,d+3]*((d-k)*au*difejt[j+1,t]-delta(t,2)*(d-k)*(d-j)*M
[j+2,2]);
else
q:=(d-k)*au*difejt[j+1,t]-delta(t,2)*(d-k)*(d-j)*M[j+2,2];
end if;
q;
end:
> f:=proc(d,j,t,M,k,Dr,difejt,au,aux1jt,Bin) local q;
q:=-delta(j,k)*delta(t,2)+(Aux3(d,j,t,k,Dr,difejt,aux1jt,Bin,M)+
Fix(d,j,t,M,k,Dr,difejt,au))/Dr[d-k,d+4];
end:
> Auxq2:=proc(d,i,Dr) local q,n;
q:=0;
for n from 0 to d-1 do
q:=q+(Dr[i,d+2-n]-Dr[i,d+1-n])*(2+((d-n-1)*Dr[1,d+3])/Dr[1,d+4]);
end do;
q;
end:

```

```

> Auxq3:=proc(d,k,Dr,Bin) local q,i;
  if k<d-1 then
    q:=Auxq2(d,d-k,Dr);
    for i from 1 to d-k-1 do;
      q:=q+Bin[i+1,k+1]*Dr[d-k-i,d+3]*Auxq2(d,i,Dr)
    end do;
  else q:=Bin[1+1,k+1]*Auxq2(d,1,Dr);
  end if;
  q;
end:
> Fixq:=proc(d,k,Dr) local q;
  q:=Dr[d-k,d+3]*(d-k+1);
  q;
end:
> fq:=proc(d,k,Dr,Bin) local q;
  q:=(Auxq3(d,k,Dr,Bin)+Fixq(d,k,Dr))/Dr[d-k,d+4];
  q;
end:
> eule:=proc(d,h,M,x0,k,quant) local C,R,j,i,x,di,au,aui,B,q,m,E,
  fjmkk,ffj2k,b,alpha,Bkk,gjm,gq; global A;
  x:=x0;
  q:=quant;
  C:=M;

  fjmkk:=Array(1..d+1,1..2,1..d-1,0);
  ffj2k:=Matrix(1..d-k,1..d-k,1);
  b:=Matrix(1..d-k,1,0);b[1,1]:=1;alpha:=Matrix(1..d-k,1,1);

  while alpha[1,1]>=0.000001 and C[k+2,2]>=0 do
    E:=evalf(C);
    R:=Drop(d,E);
    au:=Aux(d,E);
    di:=Difejt(d,E,R);
    aui:=Auxljt(d,E,R,di,au);
    B:=Bin(d);

    for kk from 1 to d-1 do
      for m from 1 to 2 do
        for j from 0 to d do
          fjmkk[j+1,m,kk]:=evalf(f(d,j,m,E,kk,R,di,au,aui,B));
        end do;
      end do;
    end do;

    for kk from 1 to d-k do
      for j from 2 to d-k do
        ffj2k[j,kk]:=fjmkk[j+k,2,kk+k-1];
      end do;
    end do;

    alpha:=MatrixInverse(ffj2k).b;
    gjm:=Array(1..d+1,1..2,0);
    for j from 1 to d+1 do
      for m from 1 to 2 do
        for kk from 1 to d-k do
          gjm[j,m]:=gjm[j,m]+alpha[kk,1]*fjmkk[j,m,kk+k-1]

```

```

end do
end do;
end do;

gq:=0;
for kk from 1 to d-k do
gq:=gq+alpha[kk,1]*fq(d, kk+k-1,R,B);
end do;

for m from 1 to 2 do
for j from 0 to d do
C[j+2,m]:=C[j+2,m]+h*gjm[j+1,m];
end do;
end do;
q:=q+h*gq;
x:=x+h;
end do;
A:=E;
print(alpha,q,C,x,1-q+x);
end:
> fp:=proc(d,p) local Q,j;
Q:=Array(1..d+2,1..2,0);
Q[0+2,1]:=(1-p)*binomial(d,0)*(1-p)^(d)*(1-(1-p)^(d-1)-binomial
(d-1,1)*p*(1-p)^(d-2))^0*((1-p)^(d-1)+binomial(d-1,1)*p*(1-p)^(
(d-2))^(d-0));
for j from 1 to d do
Q[j+2,2]:=(1-p)*binomial(d,1)*binomial(d-1,j-1)*p*(1-p)^(d-1)*(1-
(1-p)^(d-1)-binomial(d-1,1)*p*(1-p)^(d-2))^(j-1)*((1-p)^(d-1)+
binomial(d-1,1)*p*(1-p)^(d-2))^(d-j);
Q[j+2,1]:=(1-p)*binomial(d,j)*(1-p)^(d)*(1-(1-p)^(d-1)-binomial
(d-1,1)*p*(1-p)^(d-2))^(j)*((1-p)^(d-1)+binomial(d-1,1)*p*(1-p)^(
(d-2))^(d-j));
end do;
Q;
end:
> A:=fp(3,0.001)

```

$$A := \begin{bmatrix} 0 & 0 \\ 0.9960030080 & 0 \\ 0.000002988012012 & 0.002991003015 \\ 2.988015000 \cdot 10^{-12} & 5.982012012 \cdot 10^{-9} \\ 9.960059960 \cdot 10^{-19} & 2.991008997 \cdot 10^{-15} \end{bmatrix} \quad (1)$$

```

> eule(3,0.0001,A,0,1,0);

```

$$\begin{bmatrix} 0.699311925158584 \\ 0.300688074841417 \end{bmatrix}, 0.944739691533200, \begin{bmatrix} 0 & 0 \\ 0.000886296161521942 & 0 \\ 0.00579618791040607 & -0.0000499330569568065 \\ 0.0163302745114965 & 5.98201201196886 \cdot 10^{-9} \\ 0.0139669961617109 & 0.0173274858087237 \end{bmatrix}, 0.4219, 0.477160308466800 \quad (2)$$

A rotina (2) acima é a saída do comando $eule(3, 0.0001, A, 0, 1, 0)$, que implementa o Método de Euler na fase 1 do caso $d = 3$ com passo de Euler igual a 0,0001. Isso significa que cada passo da aproximação de Euler corresponde a $n/1000$ passos do algoritmo. A matriz A acima (Veja (1)) é a matriz inicial do processo e informa os números esperados de vértices vermelhos e azuis de cada grau obtidos após a etapa de pré-processamento com $p_0 = 0,001$. O elemento da linha i e coluna j é a aproximação de $z_{i+2,c}(0)$, onde $j = 1$ significa $c = a$ e $j = 2$ significa $c = v$. Nessa implementação, $z_{d,a}(0)$ e $z_{d,v}(0)$ contam de fato vértices azuis e vermelhos de grau d , mas $z_{-1,a}$, $z_{-1,v}$ e $z_{0,v}$ são sempre nulas em qualquer instante x .

Os dados de saída da rotina (2) acima podem ser interpretados da seguinte forma. A primeira matriz, que é do tipo 2×1 , fornece as proporções $\alpha_1^1(x_1)$ e $\alpha_2^1(x_1)$ no final da fase 1. O segundo dado de saída, o número 0,9447..., representa o tamanho do conjunto dominado $D(t)$ produzido até a linha 24 desse algoritmo, o que significa que o tamanho aproximado desse conjunto é de $0,9447n$. O quarto dado de saída, o número 0,4219, é a duração do processo, o que significa que o número de iterações da primeira fase do algoritmo de deleção é de $0,4219n$. Como cada iteração adiciona um vértice ao conjunto 2-dominante, esse também é o tamanho do conjunto 2-dominante produzido até a linha 24 do Algoritmo 8(c). A matriz 5×2 da saída da rotina (2) é a matriz que indica o fim da fase 1. Essa matriz representaria a proporção dos vértices de cada tipo (como na matriz A) após 4220 passos de Euler, mas verificamos que a condição de que as proporções sejam positivas foi violada, o que significa que o passo foi interrompido anteriormente (a matriz de saída é o primeiro passo em que isso ocorre). O elemento da linha i e coluna j da matriz A após o passo 4219 é a aproximação de $z_{i+2,c}(x_1)$, onde $j = 1$ significa $c = a$ e $j = 2$ significa $c = v$. O quinto e último dado acima, o número 0,4771... representa a proporção de vértices no conjunto D_2 , produzido

na linha 25 do Algoritmo 8(c). Para obter esse número, somamos as coordenadas não-nulas da matriz A no final da fase ao número 0,4219. Note que os dados de saída indicam que a fase 1 é a fase final desse processo. Isso ocorre porque a condição que interrompeu a implementação foi a coordenada negativa da matriz de saída $-0,000049\dots$, que representa os vértices vermelhos de grau 1. Esses dados indicam que todos os vértices vermelhos de grau 1 foram eliminados antes que a coordenada $\alpha_1^1(x_1)$ da matriz coluna fosse anulada. Com relação às condições técnicas apresentadas no Capítulo 8, isso significa que a fase 1 foi encerrada porque a condição (i.1) foi violada.

Note também que, ao final dessa fase, sobram alguns vértices azuis de grau menor do que d , cuja soma é aproximadamente 0,022. O que o passo da linha 25 faz é adicionar esses vértices ao conjunto 2 dominante, mas se a partir desse ponto, o vértice da regra de seleção fosse azul em vez de vermelho, esses vértices poderiam ainda ser processados melhorando a cota para o conjunto 2-dominante final em, no máximo, $0,022n$ (na verdade, melhorias decorrentes disso seriam muito inferiores a esse valor, e portanto não a incluímos em nosso algoritmo).

Os dados a seguir são referentes às aplicações do Algoritmo 8(c) para $d \in \{4, 5, 6\}$. Discutiremos em mais detalhes o caso $d = 5$.

$$\begin{array}{l}
 \text{> } A := \text{fp}(4, 0.001) \\
 A := \begin{bmatrix}
 0 & 0 \\
 0.9949980579 & 0 \\
 0.00001193205248 & 0.003983988152 \\
 5.365860088 \cdot 10^{-11} & 3.583209686 \cdot 10^{-8} \\
 1.072459785 \cdot 10^{-16} & 1.074249484 \cdot 10^{-13} \\
 8.038110185 \cdot 10^{-23} & 1.073536537 \cdot 10^{-19}
 \end{bmatrix}
 \end{array} \tag{3}$$

```

> eule(4,0.0001,A,0,1,0)
[ -0.000193142622367731
  0.963477635732836
  0.0367155068895319 ] , 0.267666278114937,
[ 0 0
  0.412744668897112 0.
  0.136543161315756 0.163878738017372
  0.0171732263409764 3.58320968599970 10-8
  0.000942336253229145 1.07424944599615 10-13
  0.0000191924573777131 0.0000263767612403789 ] , 0.1171, 0.849433721885063

```

(4)

```

> eule(4,0.0001,A,0.1171,2,0.267666278114937)
[ 0.777962553191819
  0.222037446808181 ] , 0.949020164363390,
[ 0. 0.
  0.0000404450532755453 0.
  0.000621544816632253 0.000684975701943058
  0.00417257090135694 -0.0000342498952397106
  0.0147002945968136 1.07425179134229 10-13
  0.0176616434190441 0.0123827120085460 ] , 0.3752, 0.426179835636610

```

(5)

```

> A:=fp(5,0.001)
A := [ 0 0
       0.9939851997 0
       0.00002977999490 0.004974930709
       3.568858356 10-10 1.192399333 10-7
       2.138474168 10-15 1.071735658 10-12
       6.406911276 10-21 4.281255219 10-18
       7.678093605 10-27 6.413363029 10-24 ]

```

(6)

```

> eule(5,0.0001,A,0,1,0)
[ -0.000549059610745513
  0.976132801499297
  0.0238491191809322
  0.000567138930515933 ] , 0.110539127916849,
[ 0 0
  0.636132773227487 0.
  0.116230817106048 0.127391030121084
  0.00839069676429675 1.19239933300012 10-7
  0.000300044640806550 1.07173565229991 10-12
  0.00000533758359875295 4.28132229549851 10-18
  3.78460742568237 10-8 4.55541968153976 10-8 ] , 0.0491, 0.938560872083151

```

(7)

```

> eule(5,0.0001,A,0.0491,2,0.110539127916849)
[ -0.000854924301591824
  0.947200939417432
  0.0536539848841595 ] , 0.463135279163349,
[ 0. 0.
  0.0999718087467794 0.
  0.130805682862426 0.118465017358426
  0.0713632028469575 0.0935182123627730
  0.0192127759766858 1.07173574053601 10-12
  0.00254050541961763 4.27702885489546 10-18
  0.000132854418115760 0.0000934691836206670 ] , 0.1690, 0.705864720836651

```

(8)

```

> eule(5,0.0001,A,0.1690,3,0.463135279163349)
[ 0.815756562468409
  0.184243437531591 ] , 0.955140853034347,
[ 0. 0.
  0.00000114949762068671 0.
  0.0000347921851332661 0.0000278991974049112
  0.000439444541102413 0.000755435219607321
  0.00307259007763807 -0.0000400172870116268
  0.0123685407422564 4.23886493842398 10-18
  0.0180602580233153 0.00969643951722304 ] , 0.3411, 0.385959146965653

```

(9)

As rotinas (7), (8) e (9) acima são as saídas do comandos $eule(5, 0.0001, A, 0.0491, 1, 0)$, $eule(5, 0.0001, A, 0.1690, 1, 0.110539127916849)$ e $eule(5, 0.0001, A, 0, 1, 0.463135279163349)$, que implementam o Método de Euler nas fases 1, 2 e 3, respectivamente, do caso $d = 5$ com passo de Euler igual a $0,0001$. A matriz A da saída (6) acima é a matriz inicial do processo para o caso em que $d = 5$ e suas coordenadas têm a mesma descrição já feita para a matriz A da saída (1).

Quanto aos dados de saída da rotina (7) a interpretação é a seguinte. A primeira matriz, que é do tipo 4×1 , fornece as proporções $\alpha_1^1(x_1)$, $\alpha_2^1(x_1)$, $\alpha_3^1(x_1)$ e $\alpha_4^1(x_1)$ que indicam o fim da fase 1. Essa matriz representaria as proporções α após 492 passos de Euler. Como uma das coordenadas é negativa, as condições são violadas e a fase é encerrada após 491 passos. O segundo dado de saída, o número $0,1105\dots$, indica que tamanho do conjunto dominado $D(t)$ produzido até o final da primeira fase tem tamanho aproximado de $0,1105n$. O quarto dado de saída, o número $0,0491$, é o instante no qual a fase 1 terminou, o que significa que o número de iterações da primeira fase do algoritmo de deleção é de $0,0491n$. Como cada iteração adiciona um vértice ao conjunto 2-dominante, esse também é o tamanho do conjunto 2-dominante produzido até o final dessa fase. A matriz 7×2 da saída da rotina (7) é a matriz de saída da fase 1 e suas coordenadas têm a mesma descrição já feita para o caso $d = 3$ em (2). O quinto e último dado acima, o número $0,9385\dots$ representa a proporção de vértices que o conjunto conjunto 2-dominante teria se essa fosse a fase final do processo. Para obter esse número, somamos as coordenadas não-nulas da matriz A no final da fase 1 ao número $0,0491$. Note que os dados de saída (7) indicam que a fase 1 não é a fase final do caso $d = 5$. Isso ocorre porque a condição que interrompeu a implementação foi a coordenada negativa $-0,0005\dots$ relativa ao valor de $\alpha_1^1(x_1)$ da matriz coluna de (7). A fase 2 tem como entrada a matriz A de saída da fase 1, que ocorre após 491 passos de Euler. Os dados da saída da fase 2 são apresentados em (8) acima

e têm descrição análoga à feita para a saída (7). Note que a fase 2 também não é a fase final por motivo análogo ao da fase 1. A fase 3 tem como entrada a matriz A da saída da fase 2, que ocorre após 1690 passos de Euler, e seus dados de saída são apresentados em (9). Neste caso, a fase 3 é a fase final porque a condição que interrompeu a implementação dessa fase foi a coordenada negativa $-0,00004\dots$ que indica que todos os vértices vermelhos de grau 3 foram eliminados antes que $\alpha_3^3(x_3)$ se anulasse. Isso significa que a condição $(i, 3)$ foi ativada.

Note também que, ao final dessa fase, sobra um total de aproximadamente 0,015 vértices azuis de grau menor do que d , o que permitiria que o algoritmo continuasse por um curto período caso a regra de seleção passasse a escolher vértices azuis a partir deste momento. Como a melhoria seria ainda muito menor do que esse valor optamos por não incluir no nosso algoritmo.

```
> A:=fp(6,0.001)
```

$$A := \begin{bmatrix} 0 & 0 \\ 0.9929615044 & 0 \\ 0.00005945924744 & 0.005963792278 \\ 1.483526003 \cdot 10^{-9} & 2.975968010 \cdot 10^{-7} \\ 1.974102258 \cdot 10^{-14} & 5.940103335 \cdot 10^{-12} \\ 1.477633248 \cdot 10^{-19} & 5.928294172 \cdot 10^{-17} \\ 5.898782617 \cdot 10^{-25} & 2.958254244 \cdot 10^{-22} \\ 9.811759337 \cdot 10^{-31} & 5.904746234 \cdot 10^{-28} \end{bmatrix} \quad (10)$$

```
> eule(6,0.0001,A,0,1,0);
```

$$\begin{bmatrix} -0.000462367286649766 \\ 0.981153903061952 \\ 0.0189026229415824 \\ 0.000401596496500744 \\ 0.00000424478661461722 \end{bmatrix}, 0.0606180114050757, \quad (11)$$

$$\begin{bmatrix} 0 & 0 \\ 0.736141642294385 & 0. \\ 0.0961884813111116 & 0.100720653360815 \\ 0.00516660982096329 & 2.97596801000068 \cdot 10^{-7} \\ 0.000146998584137928 & 5.94010333611994 \cdot 10^{-12} \\ 0.00000234069847262215 & 5.92828914833428 \cdot 10^{-17} \\ 1.97891610984817 \cdot 10^{-8} & 2.95651651388126 \cdot 10^{-22} \\ 6.94142278658510 \cdot 10^{-11} & 7.44395267978432 \cdot 10^{-11} \end{bmatrix}, 0.0270, 0.966381988594924$$

```
> eule(6, 0.0001, A, 0.0270, 2, 0.0606180114050757)
[ -0.000766527747917225
  0.958527895521752
  0.0396964666785352
  0.00254216554763035 ] , 0.260397402760955,
```

(12)

```

[ 0. 0.
  0.240113182404759 0.
  0.189670510931907 0.157039993885630
  0.0626525028146316 0.0774187870211042
  0.0108405829324473 5.94010315767633 10-12
  0.00104328202740231 5.92776943526767 10-17
  0.0000532101438439635 2.62104708665333 10-23
  0.00000112622418729755 7.59792166278540 10-7 ] , 0.0945, 0.834102597239045
```

```
> eule(6, 0.0001, A, 0.0945, 3, 0.260397402760955)
```

```
[ -0.000244289586055397
  0.942447447607293
  0.0577968419787627 ] , 0.565021092150497,
```

(13)

```

[ 0. 0.
  0.0217949536257503 0.
  0.0604040210813899 0.0421623567368111
  0.0744236625207885 0.0984109607423654
  0.0494452226363953 0.0657852094012955
  0.0181971844784581 5.92750895569573 10-17
  0.00351085455642238 3.29787079162472 10-22
  0.000278927888367766 0.000135985431951345 ] , 0.1799, 0.614878907849503
```

```
> eule(6, 0.0001, A, 0.1799, 4, 0.565021092150497)
```

```
[ 0.835242123340582  
  0.164757876659418 ], 0.960511870084885,
```

(14)

```
      0.          0.  
  2.46851858016573 10-8      0.  
  0.00000126780588727831  8.03524386655000 10-7  
  0.0000276396769128566  0.0000384589548425806  
  0.000329933963126607   0.000708504337503585  
  0.00239248890036052   -0.0000282302766510071  
  0.0105238153880462    -2.46502309749037 10-20  
  0.0174566579754198    0.00795530151288182  
], 0.3141, 0.353588129915115
```