



UNIVERSIDADE FEDERAL DO CEARÁ
CENTRO DE CIÊNCIAS
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CLÁUDIO SOARES DE CARVALHO NETO

PROBLEMAS DE DECOMPOSIÇÃO DE FLUXOS EM REDES

FORTALEZA
2025

CLÁUDIO SOARES DE CARVALHO NETO

PROBLEMAS DE DECOMPOSIÇÃO DE FLUXOS EM REDES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Doutor em Ciência da Computação. Área de concentração: Teoria dos Grafos (Teoria da Computação).

Orientadora: Dra. Cláudia Linhares Sales

Coorientadora: Dra. Ana Karolinna Maia de Oliveira

FORTALEZA

2025

Dados Internacionais de Catalogação na Publicação

Universidade Federal do Ceará

Sistema de Bibliotecas

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C322p Carvalho Neto, Cláudio Soares de.

Problemas de Decomposição de Fluxos em Redes / Cláudio Soares de Carvalho Neto. –
2025.

107 f. : il. color.

Tese (doutorado) – Universidade Federal do Ceará, Centro de Ciências, Programa de
Pós-Graduação em Ciência da Computação , Fortaleza, 2025.

Orientação: Profa. Dra. Cláudia Linhares Sales.

Coorientação: Profa. Dra. Ana Karolinna Maia de Oliveira.

1. Fluxos. 2. Redes arco-coloridas. 3. Decomposição. 4. Ramificações. 5. Complexidade.

I. Título.

CDD 005

CLÁUDIO SOARES DE CARVALHO NETO

PROBLEMAS DE DECOMPOSIÇÃO DE FLUXOS EM REDES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação do Centro de Ciências da Universidade Federal do Ceará, como requisito parcial para obtenção do título de Doutor em Ciência da Computação. Área de concentração: Teoria dos Grafos (Teoria da Computação).

Aprovada em: 10/03/2025

BANCA EXAMINADORA

Dra. Cláudia Linhares Sales (Orientadora)
Universidade Federal do Ceará (UFC)

Dra. Ana Karolinna Maia de Oliveira (Coorientadora)
Universidade Federal do Ceará (UFC)

Dr. Manoel Bezerra Campêlo Neto
Universidade Federal do Ceará (UFC)

Dr. Carlos Vinícius Gomes Costa Lima
Universidade Federal do Cariri (UFCA)

Dr. Fábio Protti
Universidade Federal Fluminense (UFF)

Aos meus pais, Luiz Freitas Carvalho e Maria
José Pedrosa Carvalho, *in memoriam*.

AGRADECIMENTOS

Em primeiro lugar, a Deus, por me guiar na realização deste trabalho. Apesar das dificuldades, acreditar que Deus está sempre à frente de tudo me fortalece e me ajuda a seguir com meus projetos e a enfrentar os desafios.

Aos meus familiares (irmãos, sobrinhos, cunhados), principalmente às minhas irmãs Cláudia, Cristiane e Liliane e ao meu cunhado Romildo, com quem tive mais contato durante o doutorado, pelo apoio e incentivo.

A todos os professores do grupo de pesquisa ParGO, pelo conhecimento transmitido, pelo profissionalismo. Principalmente, agradeço às minhas orientadoras, Cláudia Linhares Sales e Ana Karolinna Maia de Oliveira, pela disponibilidade para me orientarem, pelo incentivo, compreensão, cuidado, pela amizade e confiança depositada em mim. É uma honra ter sido orientado por elas.

Aos demais membros da Comissão Examinadora, Professores Manoel Campêlo, Carlos Vinícius e Fábio Protti, pelo tempo dedicado, pelas sugestões e comentários, muito valiosos para a conclusão desse trabalho.

Aos meus colegas de doutorado Raul Lopes e Jonas Costa, com quem tive o prazer de trabalhar durante o curso, por me ajudarem a obter os resultados desta tese.

À minha amiga Hilda Torres, com quem trabalho em Sobral, por todo o cuidado e suporte, por me incentivar a avançar cada vez mais durante o curso de doutorado. O seu apoio foi fundamental para que eu continuasse firme na minha pesquisa, nos meus estudos. Também agradeço aos meus amigos Erickson Vieira e Valdizar Forte, que estiveram mais próximos durante esse período, por todo apoio e suporte.

Ao Professor Glaudenir Brasil, Pró-Reitor de Pesquisa e Pós-Graduação da Universidade Estadual Vale do Acaraú (UVA), universidade em que trabalho, e ao Professor Rudini Sampaio, da UFC, por terem articulado a criação do Doutorado Interinstitucional (DINTER), o que me possibilitou fazer esse curso. Também agradeço aos meus colegas de trabalho e de DINTER, José Alex, Gilzamir Gomes, Éder Porfírio e Lourival Júnior por todo o apoio e incentivo durante a realização do doutorado.

RESUMO

Fluxos em redes são uma ferramenta fundamental na Teoria dos Grafos, com muitas aplicações práticas. uma rede \mathcal{N} é formada por um (multi)digrafo D juntamente com uma função de capacidade $u : A(D) \rightarrow \mathbb{R}_+$, e é denotada por $\mathcal{N} = (D, u)$. Um fluxo em \mathcal{N} é uma função $x : A(D) \rightarrow \mathbb{R}_+$ de modo que $x(a) \leq u(a)$ para todo $a \in A(D)$. Dizemos que um fluxo é λ -uniforme se seu valor em cada arco com fluxo positivo é exatamente λ , para algum $\lambda \in \mathbb{R}_+^*$. Segundo Granata et al. (2013), redes arco-coloridas são usadas para modelar diferenças qualitativas entre as diversas regiões por onde o fluxo será enviado. Elas têm aplicações em diversas áreas tais como redes de comunicação, transportes multimodais, empacotamento etc. Neste trabalho, consideramos dois tipos de fluxos - (s, t) -fluxo e fluxo s -ramificado.

Um (s, t) -fluxo representa a quantidade de fluxo que pode ser enviada de s a t em uma rede $\mathcal{N} = (D, u)$. De acordo com Baier, Köhler e Skutella (2005), um (s, t) -fluxo é k -divisível se ele pode ser decomposto em até k caminhos. Nós consideramos o problema de decompor um (s, t) -fluxo em uma rede arco-colorida com custo mínimo, isto é, com a menor soma do custo de seus caminhos, onde o custo de cada caminho é dado pelo seu número de cores. Mostramos que esse problema é \mathcal{NP} -Difícil para (s, t) -fluxos em geral. Quando restringimos o problema a (s, t) -fluxos λ -uniformes, mostramos que ele pode ser resolvido em tempo polinomial em redes com no máximo duas cores, e é \mathcal{NP} -Difícil para redes em geral com três cores e para redes acíclicas com pelo menos cinco cores.

Um fluxo s -ramificado deve alcançar todos os outros vértices de uma rede $\mathcal{N} = (D, u)$ a partir de um vértice s , perdendo exatamente uma unidade de fluxo em cada vértice diferente de s . Segundo Bang-Jensen e Bessy (2014), quando $u \equiv n - 1$, a rede admite k fluxos s -ramificados arco-disjuntos se e somente se D contém k s -ramificações arco-disjuntas. Assim, um resultado clássico de Edmonds (1973), que diz que um digrafo contém k s -ramificações se e somente se o grau de entrada de todo conjunto $X \subseteq V(D) \setminus \{s\}$ for pelo menos k , também caracteriza a existência de k fluxos s -ramificados arco-disjuntos nessas redes. Neste trabalho, investigamos como uma propriedade que é uma extensão natural da caracterização de Edmonds está relacionada à existência de k fluxos s -ramificados arco-disjuntos em redes. Embora essa propriedade seja sempre necessária para a existência de tais fluxos, nós mostramos que ela nem sempre é suficiente e que é difícil decidir se os fluxos desejados existem, mesmo se tivermos o conhecimento prévio de que ela é satisfeita.

Palavras-chave: fluxos; redes arco-coloridas; decomposição; ramificações; complexidade.

ABSTRACT

Flows on networks are a fundamental tool in the Graph Theory, with several practical applications. A *network* \mathcal{N} is formed by a (multi)digraph D together with a *capacity function* $u : A(D) \rightarrow \mathbb{R}_+$, and it is denoted by $\mathcal{N} = (D, u)$. A *flow on* \mathcal{N} is a function $x : A(D) \rightarrow \mathbb{R}_+$ such that $x(a) \leq u(a)$ for all $a \in A(D)$. We say that a flow is λ -uniform if its value on each arc of the network with positive flow value is exactly λ , for some $\lambda \in \mathbb{R}_+^*$. According to Granata et al. (2013), arc-coloured networks are used to model qualitative differences among different regions through which the flow will be sent. They have applications in several areas such as communication networks, multimodal transportation, packing etc. In this work, we deal with two types of flows - (s, t) -flow and s -branching flow.

An (s, t) -flow represents the amount of flow that can be sent from s to t in a network $\mathcal{N} = (D, u)$. According to Baier, Köhler e Skutella (2005), an (s, t) -flow is k -splittable if it can be decomposed into up to k paths. We consider the problem of decomposing a flow over an arc-coloured network with minimum cost, that is, with minimum sum of the cost of its paths, where the cost of each path is given by its number of colours. We show that this problem is \mathcal{NP} -Hard for general flows. When we restrict the problem to λ -uniform flows, we show that it can be solved in polynomial time for networks with at most two colours, and it is \mathcal{NP} -Hard for general networks with three colours and for acyclic networks with at least five colours.

An s -branching flow must reach every vertex of a network $\mathcal{N} = (D, u)$ from a vertex s while loosing exactly one unit of flow in each vertex other than s . According to Bang-Jensen e Bessy (2014), when $u \equiv n - 1$, the network admits k arc-disjoint s -branching flows if and only if its associated digraph D contains k arc-disjoint s -branchings. Thus, a classical result by Edmonds (1973) stating that a digraph contains k arc-disjoint s -branchings if and only if the indegree of every set $X \subseteq V(D) \setminus \{s\}$ is at least k also characterizes the existence of k arc-disjoint s -branching flows in those networks. In this work, we investigate how a property that is a natural extension of the characterization by Edmonds is related to the existence of k arc-disjoint s -branching flows in networks. Although this property is always necessary for the existence of such flows, we show that it is not always sufficient and that it is hard to decide if the desired flows exist even if we know beforehand that the network satisfies it.

Keywords: flows; arc-coloured networks; decomposition; branchings; complexity.

LISTA DE FIGURAS

Figura 1 – Exemplos de grafo e multigrafo.	16
Figura 2 – Remoção de vértices e arestas.	17
Figura 3 – Multidigrafo, digrafo simples e grafo orientado.	19
Figura 4 – Subdivisão de um digrafo.	19
Figura 5 – Exemplo de fluxo viável e o seu custo em uma rede.	21
Figura 6 – Exemplo de decomposição de um (s, t) -fluxo.	23
Figura 7 – Rede residual com respeito a um (s, t) -fluxo x em uma rede \mathcal{N} .	24
Figura 8 – Exemplo de adição de fluxos $x \oplus x'$.	25
Figura 9 – Efeito causado no fluxo original por um caminho aumentante.	26
Figura 10 – Exemplos de (s, t) -corte.	27
Figura 11 – Exemplo de fluxo k-divisível.	29
Figura 12 – Exemplo de fluxo máximo 4-divisível não inteiro.	30
Figura 13 – Exemplo de fluxo 3-divisível.	33
Figura 14 – Redução do <i>SAT</i> para o Fluxo Máximo 2-Divisível (primeira etapa).	34
Figura 15 – Redução do <i>SAT</i> para o <i>FluxoMax2Div</i> .	35
Figura 16 – Rede com fluxo máximo 4-divisível uniforme não inteiro.	35
Figura 17 – Construção de rede a partir de uma instância do 3-Partição.	37
Figura 18 – Influência da circulação na decomposição de um fluxo.	39
Figura 19 – Exemplo ruim para a decomposição de fluxo de forma gulosa.	40
Figura 20 – Redução do <i>3SAT</i> para o <i>FluxoKDiv</i> com 3 valores de fluxo.	41
Figura 21 – Exemplos de (s, t) -corte em redes arco-coloridas.	44
Figura 22 – Exemplo de redução do <i>Conjunto Dominante</i> para <i>KColSpanningTree</i> .	47
Figura 23 – Exemplo de Redução do <i>MCSpaningTree</i> para <i>MCMMaxFlow</i> .	48
Figura 24 – Exemplo de fluxo em rede sobre um digrafo arco-colorido.	49
Figura 25 – Exemplo de redução do <i>SetCover</i> para o <i>KColourPath</i> .	51
Figura 26 – Construção de rede a partir de uma instância do 3-Partição.	52
Figura 27 – Exemplo da decomposição produzida pelo Algoritmo 1.	54
Figura 28 – Influência de ciclos na decomposição de fluxo uniforme em redes coloridas.	56
Figura 29 – Rede bicromática e acíclica com dois valores de fluxo nos arcos.	57
Figura 30 – Redução do Problema <i>FluxoKDiv</i> para o <i>DFColCustoK</i> .	58
Figura 31 – Exemplo de redução do <i>Weak 2-Linkage</i> para o <i>DFColCustoK</i> .	60
Figura 32 – Exemplo de rede sobre um <i>DAG</i> com 3 cores e fluxo uniforme.	61
Figura 33 – Exemplo de Redução do 1-em-3 <i>SAT</i> para o <i>DFColCustoK</i> .	64
Figura 34 – Exemplos de fluxos s -ramificados em uma rede $\mathcal{N} = (D, u \equiv 4)$.	67
Figura 35 – Exemplo de rede \mathcal{N} e sua equivalente \mathcal{N}' com fonte e sumidouro únicos.	68
Figura 36 – Contraexemplo para a Conjectura 5.6, para $k = \lambda = 2$.	71

Figura 37 – Contraexemplo para $k \geq 2$ par e uma constante $\lambda \geq 2$.	72
Figura 38 – Contraexemplo para $k > 3$, $c > 2$ e $\lambda = n - c$.	75
Figura 39 – Exemplos de multiarborescência e multispindle.	76
Figura 40 – Rede construída a partir de uma instância do 3-Partição.	82

LISTA DE QUADROS

Quadro 1 – Resultados de complexidade para o problema $DFColCustoK$ 65

SUMÁRIO

1	INTRODUÇÃO	12
1.1	Decomposição de (s,t) -fluxos em redes arco-coloridas	13
1.2	Fluxos s -ramificados arco-disjuntos	14
1.3	Organização do texto	15
2	DEFINIÇÕES GERAIS	16
2.1	Grafos	16
2.2	Digrafos	18
2.3	Redes e Fluxos	20
2.3.1	<i>Decomposição de fluxos</i>	22
2.3.2	<i>Rede residual</i>	24
2.3.3	<i>Fluxo máximo e corte mínimo</i>	26
3	FLUXOS K-DIVISÍVEIS	28
3.1	O Problema e suas variações	28
3.2	Maximizando o fluxo enviado por um número dado de caminhos	30
3.3	Minimizando o número de caminhos de um dado fluxo	36
4	FLUXOS EM REDES ARCO-COLORIDAS	44
4.1	O Problema e suas variações	44
4.2	Encontrando o fluxo máximo com o menor número de cores	45
4.3	Minimizando o custo da decomposição de um fluxo colorido	49
4.3.1	<i>Redes com apenas uma cor</i>	53
4.3.2	<i>Redes com apenas duas cores</i>	56
4.3.3	<i>Redes com pelo menos três cores</i>	59
5	FLUXOS RAMIFICADOS ARCO-DISJUNTOS	66
5.1	Ramificações	66
5.2	Fluxos s -ramificados	66
6	FLUXOS RAMIFICADOS ARCO-DISJUNTOS EM REDES (k, λ, s)-SUFICIENTES	71
6.1	Casos Negativos	71
6.2	Casos Positivos	76
6.3	Complexidade	81
7	CONCLUSÃO	84
7.1	Resultados obtidos	84
7.2	Próximas etapas	85
	REFERÊNCIAS	87

APÊNDICE A – COMPLEXIDADE DE ALGORITMOS	91
APÊNDICE B – ALGORITMOS DE FLUXO MÁXIMO	96

1 INTRODUÇÃO

Segundo Bang-Jensen e Gutin (2008), fluxos em redes são provavelmente, de forma isolada, a mais importante ferramenta para aplicações em Grafos Direcionados. Do ponto de vista teórico, constituem uma generalização para o problema de caminho mais curto e o de caminhos internamente arco-disjuntos entre um dado par de vértices. Do ponto de vista prático, eles permitem, com certa elegância e simplicidade, modelar problemas em diferentes áreas de estudo tais como transportes, logística e telecomunicações. A teoria é relativamente simples e, combinada à enorme aplicabilidade a problemas da vida real, torna o assunto muito atrativo para estudo.

Uma *rede* é definida formalmente em Bang-Jensen e Gutin (2008) como um (multi)digrafo $D = (V, A)$ associado com as seguintes funções de $V \times V \rightarrow \mathbb{R}_+$: um *limite inferior* l_{ij} , uma *capacidade* u_{ij} e um *custo* c_{ij} . Também é possível usar uma função $b : V \rightarrow \mathbb{R}$, chamada de *vetor de balanço*. De forma abreviada, $\mathcal{N} = (D, l, u, b, c)$ denota uma rede sobre um digrafo $D = (V, A)$ e com parâmetros l , u , b e c . Se não há limites inferiores, vetor de balanço ou custos, as respectivas letras são omitidas. Por exemplo, em $\mathcal{N} = (D, u)$ é considerada apenas a capacidade dos arcos. Usamos $\mathcal{N} = (D, u \equiv \lambda)$, para indicar que todo arco da rede \mathcal{N} tem capacidade igual a λ .

Um *fluxo* em uma rede \mathcal{N} é uma função que associa a cada arco desta um valor não negativo. Dois fluxos x e y em uma rede \mathcal{N} são ditos arco-disjuntos se $x(a) \cdot y(a) = 0$, para todo arco a da rede. Dizemos que um fluxo é λ -uniforme se o valor de fluxo em cada arco da rede está em $\{0, \lambda\}$. O *balanço* de um vértice v em relação a um fluxo x , denotado por $b_x(v)$, é a diferença entre a quantidade de fluxo que sai de v , e a que entra em v . Um fluxo pode ser descrito em função do seu vetor de balanço. Neste trabalho, abordamos basicamente dois tipos de fluxo – o (s, t) -fluxo e o fluxo s -ramificado – definidos a seguir.

Sejam s e t vértices distintos de uma rede \mathcal{N} . Um (s, t) -*fluxo* é um fluxo x tal que, para algum $k \in \mathbb{R}_+$, $b_x(s) = -b_x(t) = k$, e o balanço dos demais vértices é zero, e $x_{ij} \leq u_{ij}$ para todo arco ij . O valor do fluxo é representado por $|x| = b_x(s)$. O *Problema do Fluxo Máximo* consiste em determinar o maior valor de fluxo que pode ser enviado da origem s ao destino t na rede.

Seja uma rede \mathcal{N} com um vértice especial s . Um fluxo s -*ramificado* é aquele em que $b_x(s) = n - 1$ e o balanço dos demais vértices é -1 . Isso pode ser visto como o problema em que se deseja enviar $n - 1$ unidades de uma dada mercadoria, a partir de uma fonte s , para cada um dos outros pontos da rede. Cada ponto deve reter uma unidade e passar a quantidade excedente, caso haja, adiante.

Segundo Bang-Jensen e Gutin (2008), há equivalência entre os modelos de fluxo. Assim, é possível restringir a definição geral de fluxo em redes de forma considerável e ainda assim conservar a sua capacidade de modelagem.

Um fluxo pode ser visto como uma coleção de caminhos da origem ao destino. Dizemos que um fluxo é *k-divisível* se ele puder ser decomposto em *k* fluxos caminhos (fluxos ao longo de caminhos) não necessariamente disjuntos. Dois problemas que podem surgir nesse contexto são: dados uma rede \mathcal{N} e um inteiro positivo *k*, determinar o fluxo máximo *k*-divisível; e dados uma rede \mathcal{N} e um fluxo *x* nesta, determinar o menor valor de *k* tal que *x* é *k*-divisível. Segundo Baier, Köhler e Skutella (2005), o primeiro problema é \mathcal{NP} -Difícil. Nós mostramos que o segundo também é \mathcal{NP} -Difícil.

O foco desta tese são dois problemas relacionados à decomposição de fluxos. O primeiro trata da decomposição de (s,t) -fluxos em redes arco-coloridas, e o segundo trata de fluxos *s*-ramificados arco-disjuntos. Segue uma breve introdução a cada um deles.

1.1 Decomposição de (s,t) -fluxos em redes arco-coloridas

Em algumas situações, pode ser necessário representar características específicas de determinadas regiões da rede. Nesse contexto surge o conceito de redes arco-coloridas, em que é atribuída uma cor a cada arco da rede. As cores podem representar tipos de transporte, riscos, substâncias com as quais o produto do fluxo pode entrar em contato etc. A cor do arco traz um aspecto qualitativo, ao passo que sua capacidade, seu custo e valor de fluxo são aspectos quantitativos. Redes desse tipo têm aplicações, por exemplo, em redes de telecomunicação, sistemas de transporte multimodal etc.

Como observado por Coudert et al. (2007), os problemas de fluxo em redes arco-coloridas são diferentes, em termos de complexidade, dos seus correspondentes nas redes tradicionais. Por exemplo, determinar o fluxo máximo que pode ser enviado em uma rede de modo que o número de cores distintas dos arcos utilizados pelo fluxo seja limitado, segundo Granata et al. (2013), é um problema \mathcal{NP} -Difícil; ao passo que o problema convencional de fluxo máximo pode ser solucionado em tempo polinomial.

Pensando nas cores como sendo riscos, encontrar um caminho com o menor número de cores corresponde a encontrar uma rota mais segura para enviar o fluxo. Definimos o *custo de um caminho* como sendo o número de cores distintas nele. Em Yuan, Varma e Jue (2005), foi mostrado que encontrar um caminho de *s* a *t* com o menor número de cores em uma rede arco-colorida é um problema \mathcal{NP} -Difícil.

O estudo de fluxos em redes arco-coloridas e problemas relacionados, tais como o fluxo máximo com o menor número de cores e o corte mínimo também com o menor número de cores, não é original. Entre os trabalhos sobre cortes em grafos aresta-coloridos (versão não direcionada), podemos destacar Coudert et al. (2007), Faria et al. (2016), Bordini et al. (2019), Morawietz et al. (2020). Entretanto, trazemos aqui os primeiros estudos de decomposição de fluxos em redes arco-coloridas. Nesse contexto, definimos vários problemas, alguns dos quais não foram explorados e serão mencionados nas conclusões. Nós nos concentramos em um destes problemas, enunciado a seguir, trazendo resultados de complexidade e algoritmos polinomiais, quando possível.

O primeiro problema abordado nesta tese é uma junção de aspectos do problema de fluxos divisíveis com o de encontrar um caminho com o menor número de cores. Dado um fluxo x sobre uma rede arco-colorida, desejamos calcular o menor custo de uma decomposição de x em caminhos, que é dado pela soma dos custos de cada caminho. A menos que a rede seja monocromática, a decomposição que minimiza o custo da solução não é necessariamente a que tem o menor número de caminhos. Nós mostramos que esse problema é \mathcal{NP} -Difícil (Carvalho et al. (2023a), Carvalho et al. (2025)). Nós mostramos também que, mesmo quando o fluxo é uniforme, o problema permanece \mathcal{NP} -Difícil para redes em geral com 3 cores e para redes acíclicas com pelo menos 5 cores (Carvalho et al. (2024), Carvalho et al. (2025)).

1.2 Fluxos s -ramificados arco-disjuntos

Um digrafo com um vértice especial s é uma s -ramificação se há um caminho direcionado de s para cada um dos seus outros vértices. De acordo com Bang-Jensen e Bessy (2014), toda rede $\mathcal{N} = (D, u \equiv n - 1)$ com um vértice especial s admite k fluxos s -ramificados arco-disjuntos se e somente se o digrafo sobre o qual ela está definida contém k s -ramificações arco-disjuntas, ou seja, que não possuem arcos em comum.

Seja um digrafo $D = (V, A)$, com $s \in V$ e $\emptyset \neq X \subseteq V \setminus s$. O grau de entrada de X , denotado por $d_D^-(X)$, é a quantidade de arcos de $V \setminus X$ para X . Um resultado clássico de Edmonds (1973) diz que um digrafo D contém k s -ramificações arco-disjuntas se e somente se o grau de entrada de X é pelo menos k . Isso também caracteriza, pelo resultado de Bang-Jensen e Bessy (2014), a existência de k fluxos s -ramificados em redes com capacidade $n - 1$ e sugere que quanto maior é a capacidade, mais próximo de uma s -ramificação é o *digrafo suporte* (o que contém apenas os arcos com fluxo positivo) de um fluxo s -ramificado.

Em Costa et al. (2019), os autores propuseram uma extensão natural da caracterização de Edmonds (1973) relacionada à existência de k fluxos s -ramificados arco-disjuntos. Eles conjecturaram que toda rede $\mathcal{N} = (D, u \equiv \lambda)$ admite k fluxos s -ramificados arco-disjuntos se e somente se $d_D^-(X) \geq k \cdot \lceil \frac{|X|}{\lambda} \rceil$, para todo $\emptyset \neq X \subseteq V \setminus \{s\}$. Dizemos que redes que satisfazem essa propriedade são (k, λ, s) -suficientes.

Nesta tese, nós examinamos a conjectura de Costa et al. (2019). Nós mostramos um contraexemplo para essa conjectura, ou seja, que essa propriedade nem sempre garante a existência dos k fluxos s -ramificados arco-disjuntos (Carvalho et al. (2020)). Adicionalmente, nós mostramos como construir uma família de redes (k, λ, s) -suficientes que não admitem tais fluxos, e que é difícil decidir se esses fluxos existem, mesmo sabendo previamente que a rede é (k, λ, s) -suficiente. No lado positivo, apresentamos que ela garante a existência dos fluxos desejados em alguns casos particulares, dependendo da escolha da função de capacidade e da estrutura do digrafo sobre o qual a rede está definida (Carvalho et al. (2023b)).

1.3 Organização do texto

O texto está organizado em 7 capítulos e 2 apêndices. No Capítulo 2, são apresentados conceitos básicos de grafos, digrafos, redes e fluxos e suas notações. No Capítulo 3 são apresentados os conceitos relacionados a fluxos indivisíveis e divisíveis, e os resultados de complexidade relevantes para o estudo. Damos a nossa contribuição, na Seção 3.3, com o resultado de complexidade para o problema de decomposição de um fluxo no menor número de caminhos. No Capítulo 4, apresentamos, na Seção 4.1, as definições de redes arco-coloridas e algumas variações dos problemas de fluxo nesse tipo de rede; na Seção 4.2 abordamos a complexidade do problema de encontrar um fluxo máximo usando o menor número de cores; e na Seção 4.3 formalizamos o problema de decompor um fluxo dado em uma rede arco-colorida, minimizando o custo da decomposição, e apresentamos os nossos resultados. No Capítulo 5, apresentamos alguns resultados sobre fluxos s -ramificados, que embasaram o nosso estudo, juntamente com a conjectura de Costa et al. (2019) sobre a caracterização de redes que admitem k fluxos s -ramificados arco-disjuntos. No Capítulo 6, damos as nossas contribuições para o estudo dessa conjectura. Por fim, no Capítulo 7, são feitas as considerações finais e são dadas algumas direções para continuidade dos estudos nos problemas apresentados.

O Apêndice A traz uma revisão dos principais conceitos referentes à complexidade de algoritmos, que são usados ao longo do texto.

O Apêndice B traz uma visão geral sobre o Método de Ford-Fulkerson para o Problema do Fluxo Máximo, e alguns dos principais algoritmos na literatura para esse problema, abordando as limitações e a complexidade de cada um.

2 DEFINIÇÕES GERAIS

Este capítulo traz os conceitos e notações de grafos, digrafos, redes e fluxos, que são utilizados no restante do texto.

2.1 Grafos

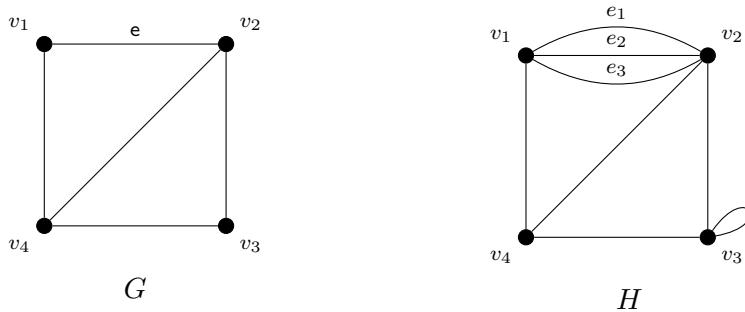
Os conceitos e notações adotados nesta seção estão em Bondy e Murty (2008).

Um *grafo* é definido por um par ordenado $G = (V, E)$, sendo V o conjunto não vazio de vértices e E o conjunto de arestas de G , e por uma função de incidência de G , ψ_G , que associa cada aresta com um par não ordenado de vértices de G . É comum usar as notações $V(G)$ e $E(G)$ para fazer referência, respectivamente, aos conjuntos de vértices e de arestas de um grafo G . Sejam e uma aresta, u e v vértices tais que $\psi_G(e) = \{u, v\}$. Diz-se que u e v são as extremidades de e . A *ordem* de um grafo, denotada por n , é o seu número de vértices; e o *tamanho*, denotado por m , é o seu número de arestas.

As extremidades de uma aresta são ditas incidentes a esta. Dois vértices são *adjacentes* se incidem sobre uma mesma aresta e duas arestas são adjacentes se incidem sobre um mesmo vértice. Vértices adjacentes distintos são chamados de *vizinhos*. O conjunto dos vizinhos de um vértice v em um grafo G é representado por $N_G(v)$.

Um *laço* é uma aresta cujas extremidades são iguais. Se duas ou mais arestas incidem sobre o mesmo par de vértices, estas são chamadas de *múltiplas*. Um grafo com arestas desse tipo é chamado de *multigrafo*. Um *grafo simples* é aquele que não possui laços ou arestas múltiplas e possui um número finito de vértices. Nesse caso, uma aresta e , com $\psi_G(e) = \{u, v\}$, pode ser referenciada por uv . Na Figura 1, G é um grafo e H é um multigrafo. Em G , a aresta e pode ser referenciada por v_1v_2 . Em H temos $\psi_H(e_1) = \psi_H(e_2) = \psi_H(e_3) = v_1v_2$.

Figura 1 – Exemplos de grafo e multigrafo.



Fonte: Elaborada pelo autor.

O grau de um vértice v em um grafo G , denotado por $d_G(v)$, corresponde ao número de arestas incidentes com v ; os laços contando 2 vezes.

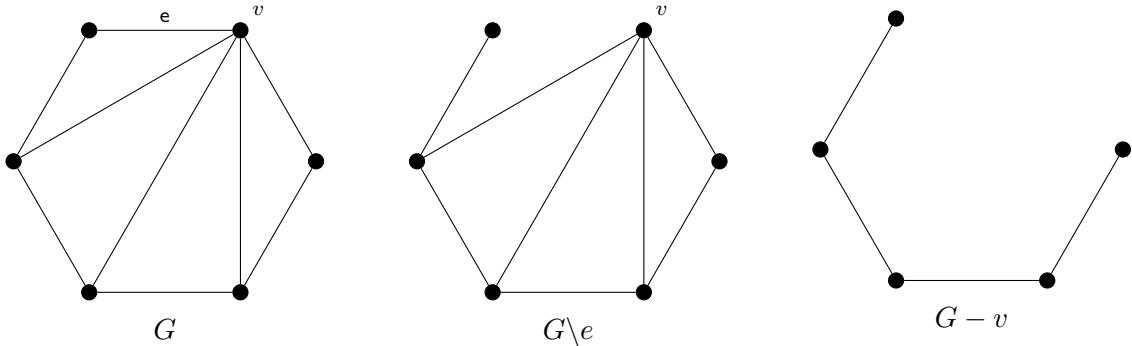
Um *passeio* em um (multi)grafo G é uma sequência $P = v_0e_1v_1 \dots v_{k-1}e_kv_k$, onde $\{v_0, \dots, v_k\} \subseteq V(G)$ e $\psi_G(e_i) = \{v_{i-1}, v_i\}$, com $1 \leq i \leq k$. No caso de grafos simples, um passeio pode ser visto como uma sequência de vértices $P = v_0v_1 \dots v_k$, onde $v_{i-1}v_i \in E(G)$ para $1 \leq i \leq k$. Os vértices v_0 e v_k são as *extremidades* de P e v_1 a v_{k-1} são os *vértices internos* de P . O tamanho de um passeio é a quantidade de arestas deste. Se $v_0 = v_k$, então P é um *passeio fechado*. Caso contrário, P é um *passeio aberto*. Um *caminho* é um passeio aberto sem repetição de vértices; e um *ciclo* é um passeio fechado sem repetição de vértices internos. A *distância* entre dois vértices x e y em um grafo G é o tamanho do menor caminho entre x e y , caso exista um; ou infinita, caso contrário.

Um grafo $G = (V, E)$ é *conexo* se há um caminho entre quaisquer dois de seus vértices. Um grafo conexo que não possui ciclos é chamado de *árvore*. Em uma árvore, uma *folha* é um vértice cujo grau é igual a um.

Sejam G e H dois grafos. Diz-se que H é *subgrafo* de G se $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ e ψ_H é a restrição de ψ_G para $E(H)$. Seja H um subgrafo de G . Diz-se que H é uma *componente conexa* (ou apenas *componente*) de G se não existe um subgrafo de G , diferente de H , que seja conexo e o contenha.

Dado um grafo G , se e é uma aresta de G , então pode-se obter um subgrafo de G com $m - 1$ arestas removendo-se e de G . O grafo resultante desta operação é denotado por $G \setminus e$. Se v é um vértice de G , pode-se obter um subgrafo de G com $n - 1$ vértices removendo-se v e todas as arestas incidentes a ele em G . O grafo resultante desta operação é denotado por $G \setminus \{v\}$ ou $G - v$. Essas operações estão ilustradas na Figura 2.

Figura 2 – Remoção de vértices e arestas.



Fonte: Elaborada pelo autor.

Um subgrafo obtido apenas pela remoção de vértices é chamado de *subgrafo induzido*. Seja X é o conjunto de vértices removidos, o subgrafo resultante é denotado por $G - X$. Considere o conjunto de vértices restantes $Y = V \setminus X$. O subgrafo é denotado por $G[Y]$ e é referenciado como o subgrafo de G induzido por Y . O conjunto de arestas é formado pelas arestas de G que possuem as duas extremidades em Y . Um *subgrafo gerador* de um grafo G é obtido apenas por remoções de arestas. Se este é conexo e acíclico, então ele é chamado de *árvore geradora* de G .

2.2 Digrafos

Os conceitos e notações adotados nesta seção estão de acordo com Bondy e Murty (2008) e Bang-Jensen e Gutin (2008).

Um *digrafo* é definido por um par ordenado $D = (V, A)$, sendo V o conjunto finito de vértices e A o conjunto de arcos de D , e por uma função de incidência de D , $\psi_D : A \rightarrow V \times V$. De forma análoga aos grafos, usa-se comumente $V(D)$ e $A(D)$ para fazer referência aos conjuntos, respectivamente, de vértices e de arcos de um digrafo D . A *ordem* e o *tamanho* de um digrafo são definidos de forma semelhante aos grafos.

Se a é um arco e $\psi_D(a) = (u, v)$, diz-se que u *domina* v ou que u é a *cauda* e v é a *cabeça* de a . Os vértices que dominam um vértice v são seus *vizinhos de entrada*. Já os dominados por v são seus *vizinhos de saída*. Esses conjuntos são denotados, respectivamente, por $N_D^-(v)$ e $N_D^+(v)$.

Um *laço* é um arco cuja cabeça é igual à cauda. Dois *arcos múltiplos* possuem a mesma cabeça e a mesma cauda. Se um digrafo possui laços ou arcos múltiplos é chamado de *multidigrafo*. Caso contrário, é chamado de *digrafo simples* ou apenas digrafo. Em um digrafo simples, um arco com cauda u e cabeça v pode ser representado por uv .

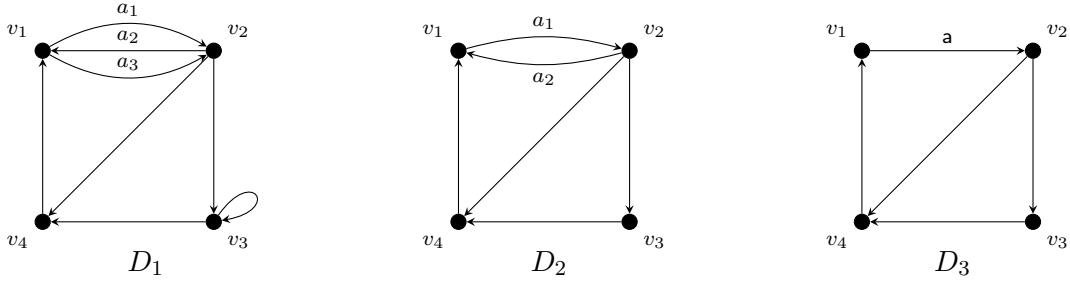
Considere um digrafo D e um vértice $v \in V(D)$. O *grau de entrada* (resp. *saída*) de v em D , denotado por $d_D^-(v)$ (resp. $d_D^+(v)$), é o número de arcos com cabeça (resp. cauda) v . O grau de v em D é a soma dos seus graus de entrada e de saída. Se $d_D^-(v) = 0$, então v é uma *fonte*. Se $d_D^+(v) = 0$, então v é um *sumidouro*. Um digrafo D é *Euleriano* se $d_D^+(v) = d_D^-(v)$, para todo $v \in V(D)$.

Os conceitos de grau de entrada e de saída podem ser estendidos para conjuntos de vértices. Assim, para um digrafo $D = (V, A)$ e um conjunto $X \subseteq V$, temos: $d_D^-(X) = |\{uv \in A \mid u \in V \setminus X \text{ e } v \in X\}|$ e $d_D^+(X) = |\{uv \in A \mid u \in X \text{ e } v \in V \setminus X\}|$.

A qualquer digrafo D pode ser associado um grafo G com o mesmo conjunto de vértices, substituindo-se cada arco por uma aresta com as mesmas extremidades e removendo-se as arestas múltiplas e laços. Este é o *grafo subjacente* de D , denotado por $G(D)$. Para qualquer grafo G é possível também obter um digrafo D equivalente, substituindo cada aresta uv de G por um par de arcos uv e vu em D . Este é o *digrafo associado* de G e é denotado por $D(G)$. Em uma *orientação* de G , cada aresta é substituída por apenas um arco. Um *grafo orientado* é uma orientação de um grafo simples.

A Figura 3 traz um multidigrafo D_1 , um digrafo simples D_2 e um grafo orientado D_3 . Os três possuem o mesmo grafo subjacente, ou seja, $G(D_1) = G(D_2) = G(D_3)$, que corresponde ao grafo G da Figura 1. Em D_1 , observe que $\psi_{D_1}(a_1) = \psi_{D_1}(a_3) = (v_1, v_2) \neq \psi_{D_1}(a_2) = (v_2, v_1)$. O digrafo D_2 é simples, pois não possui um par de arcos com a mesma cabeça e a mesma cauda e nem laços. D_3 é um grafo orientado, pois não possui um par de arcos com as mesmas extremidades. Como exemplo de grau de entrada e de saída de um conjunto de vértices, temos $d_{D_2}^+(\{v_1, v_4\}) = 1$ e $d_{D_2}^-(\{v_1, v_4\}) = 3$.

Figura 3 – Multidigrafo, digrafo simples e grafo orientado.

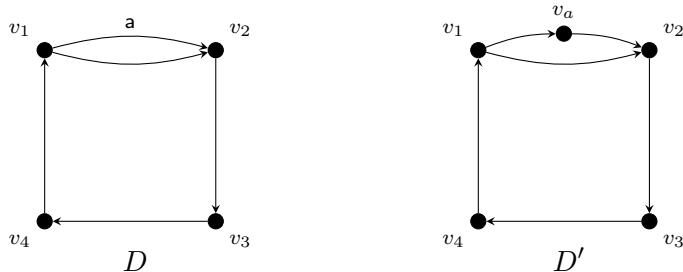


Fonte: Elaborada pelo autor.

Um *caminho direcionado* ou um *ciclo direcionado* é uma orientação de um caminho ou de um ciclo, em que cada vértice domina o seu sucessor na sequência. Se P é um caminho direcionado com extremidades x e y , diz-se que ele é um (x, y) -caminho. A *distância* de x para y , que não necessariamente é a mesma de y para x , é o tamanho do menor (x, y) -caminho, caso exista um; ou infinita, caso contrário.

A *subdivisão* de um arco a em um digrafo D , com $\psi_D(a) = (u, v)$, consiste na substituição de a por um (u, v) -caminho de tamanho pelo menos 2. Assim, é possível obter um digrafo simples D' a partir de um multidigrafo D , pela subdivisão de arcos paralelos. Diz-se que D' é uma subdivisão de D . A Figura 4 traz um exemplo disso.

Figura 4 – Subdivisão de um digrafo.



Fonte: Elaborada pelo autor.

Um digrafo D é *conexo* se $G(D)$ é conexo, e é *fortemente conexo* se, para todo par de vértices distintos x e y , há um caminho direcionado de x para y . Se D não possui ciclos direcionados, ele é chamado de *DAG* (de *Directed Acyclic Graph*). Uma árvore orientada é um *DAG* T conexo com n vértices e $n - 1$ arcos. Se um vértice s é escolhido para ser a raiz de T , esta é chamada de s -enraizada. Além disso, se há um caminho direcionado de s para cada um dos outros vértices de T , esta é uma s -*arborescência* (de saída). Neste caso, $d_T^-(s) = 0$ e $d_T^-(v) = 1$, para todo vértice $v \in V(T) \setminus \{s\}$.

O prefixo *multi* será utilizado nesse texto para indicar um tipo de digrafo em que são permitidos arcos múltiplos (com mesma cabeça e mesma cauda). Por exemplo, *multicaminho*, *multiarborescência* etc. Os conceitos de *subdigrafo*, *subdigrafo induzido*, *subdigrafo gerador* e *componente* são análogos aos respectivos conceitos em grafos.

2.3 Redes e Fluxos

Nesta seção, serão apresentados os conceitos e notações para redes e fluxos adotados por Bang-Jensen e Gutin (2008), com algumas adaptações.

Redes e fluxos em digrafos estão amplamente presentes em problemas do mundo real. Pode-se pensar, por exemplo, em uma rede de tubulações utilizada para transportar fluidos (gás, água, óleo), em que os arcos representam as tubulações e os vértices as suas junções ou reservatórios. Nas redes de transportes, os arcos podem ser vistos como vias e os vértices como cruzamentos ou pontos de distribuição. Nas redes de comunicação, os arcos representam cabos com diferentes larguras de banda conectando um conjunto de terminais, representados pelos vértices.

Formalmente, uma *rede* é um (multi)digrafo $D = (V, A)$ associado com as seguintes funções de $V \times V \rightarrow \mathbb{R}_+$: um *limite inferior* l_{ij} , uma *capacidade* u_{ij} e um *custo* c_{ij} . Se $(i, j) \notin A$, então $l_{ij} = u_{ij} = c_{ij} = 0$. Também é possível usar uma função $b : V \rightarrow \mathbb{R}$, chamada de *vetor de balanço* da rede. Se alguma dessas funções f é constante, igual a v , escrevemos $f \equiv v$. Por exemplo, $\mathcal{N} = (D, l \equiv 0, u \equiv \lambda)$ representa uma rede \mathcal{N} sobre um digrafo D em que todos os arcos possuem limite inferior 0 e capacidade λ .

De forma abreviada, $\mathcal{N} = (D, l, u, b, c)$ denota uma rede sobre um digrafo $D = (V, A)$ e com parâmetros l , u , b e c . Se não há limites inferiores, ou custos, ou um vetor de balanço prescrito, as respectivas letras serão omitidas da notação.

Seja dada uma rede $\mathcal{N} = (D, l, u, b, c)$, com $D = (V, A)$. Para um par de subconjuntos U e W de V e uma função f sobre $V \times V$, onde $f_{ij} = f(i, j)$, define-se

$$f(U, W) = \sum_{i \in U, j \in W} f_{ij} \quad (2.1)$$

Um *fluxo* em uma rede $\mathcal{N}(D, l, u, b, c)$, com $D = (V, A)$, é uma função $x : A \rightarrow \mathbb{R}_+$. O valor de um fluxo x em um arco ij é denotado por x_{ij} . Um *fluxo inteiro* x é aquele em que $x_{ij} \in \mathbb{Z}_+$ para todo arco ij . Diz-se que um fluxo x é λ -uniforme se, para um dado $\lambda \in \mathbb{R}_+$, $x_{ij} = \lambda$ para todo arco ij com fluxo positivo.

Para um fluxo x em uma rede \mathcal{N} , denota-se por $x^+(v)$ e $x^-(v)$ as somas dos valores de fluxo nos arcos que possuem, respectivamente, cauda e cabeça v . O *vetor de balanço* de x é dado por:

$$b_x(v) = x^+(v) - x^-(v), \quad \forall v \in V \quad (2.2)$$

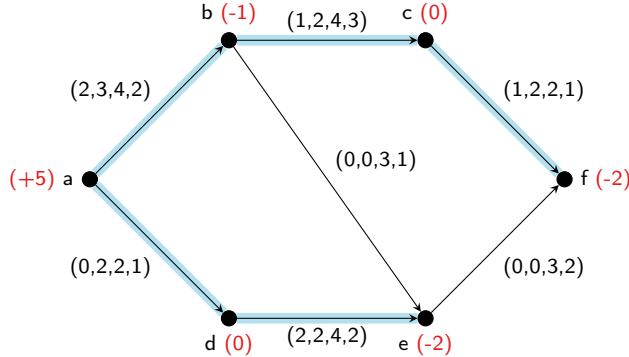
De acordo com o seu valor de balanço, v pode ser classificado como *fonte*, se $b_x(v) > 0$; *sumidouro*, se $b_x(v) < 0$; e *balanceado*, se $b_x(v) = 0$.

Um fluxo x em uma rede $\mathcal{N} = (D, l, u, b, c)$ é *viável* se $l_{ij} \leq x_{ij} \leq u_{ij}$ para todo arco ij e $b_x(v) = b(v)$ para todo vértice v da rede. Se não for especificado um vetor de balanço, desconsidera-se essa última restrição. O *custo do fluxo* x é dado por:

$$c_x^T = \sum_{ij \in A} c_{ij} \cdot x_{ij} \quad (2.3)$$

A Figura 5 traz uma rede \mathcal{N} e um fluxo x . Nela, cada arco ij está associado a uma quádrupla $(l_{ij}, x_{ij}, u_{ij}, c_{ij})$. Próximo ao rótulo de cada vértice, entre parênteses, está o seu balanço no fluxo x . Observe ainda que o custo total de x é 20. Neste trabalho, será utilizado apenas o termo fluxo para se referir a um fluxo viável em uma rede e não será levado em conta o custo em cada arco.

Figura 5 – Exemplo de fluxo viável e o seu custo em uma rede.



Fonte: Elaborada pelo autor.

Considere uma rede $\mathcal{N} = (D, l, u, b, c)$, com $D = (V, A)$, e um fluxo x nesta. Se $l \equiv 0$ e os arcos ij e ji estão em A , assume-se que $x_{ij} \times x_{ji} = 0$. Nesse caso, x é chamado de *fluxo líquido*. A motivação para isso é que geralmente se usa fluxo para modelar itens que se movem de um lugar para outro na rede. Por exemplo, enviar três unidades de i para j e duas de j para i equivale a enviar uma unidade de i para j e nada de j para i ; ou seja, o fluxo líquido de i para j é de uma unidade. O *digrafo suporte* de um fluxo x em uma rede \mathcal{N} é o subdigrafo de D que possui apenas os arcos de fluxo positivo. Na rede apresentada na Figura 5, o digrafo suporte de x não inclui apenas os arcos be e ef .

Sejam dois fluxos x e y em uma rede $\mathcal{N} = (D, l, u, b, c)$, com $D = (V, A)$. Eles são *arco-disjuntos* se $x_{ij} \times y_{ij} = 0$, para todo $ij \in A$. A *soma* de x e y é feita arco a arco, ou seja, produz um fluxo z , de modo que $z_{ij} = x_{ij} + y_{ij}$, para todo $ij \in A$.

Sejam s e t vértices distintos de uma rede $\mathcal{N} = (D, l \equiv 0, u)$. Um (s, t) -*fluxo* nesta é um fluxo x tal que, para algum $k \in \mathbb{R}_+$, tem-se:

$$b_x(v) = \begin{cases} k & , \text{ se } v = s \\ -k & , \text{ se } v = t \\ 0 & , \text{ caso contrário} \end{cases} \quad (2.4)$$

Nesse caso, o *valor* do fluxo x é definido por $|x| = b_x(s) = -b_x(t)$. Todo vértice $v \in V \setminus \{s, t\}$ é chamado de *intermediário*. A condição que impõe balanço 0 para todo vértice intermediário é chamada de *condição de conservação do fluxo*. Este é o tipo de fluxo tradicionalmente utilizado na literatura. Em Bang-Jensen e Gutin (2008, p. 131), os autores mostram que é possível restringir consideravelmente a definição geral de fluxo em redes e ainda assim conservar a sua capacidade de modelagem.

2.3.1 Decomposição de fluxos

Dada uma rede $\mathcal{N} = (D, l, u)$, entende-se por caminho ou ciclo em \mathcal{N} um caminho ou ciclo no digrafo $D = (V, A)$. Um *fluxo caminho* ao longo de um caminho P em uma rede \mathcal{N} é um fluxo x que tem $x_{ij} = k$, para todo $ij \in A(P)$ e para um valor $k \in \mathbb{R}_+$, e $x_{ij} = 0$ se $ij \notin A(P)$. De forma análoga, define-se também um *fluxo ciclo*. Um conjunto de fluxos ciclos é chamado de *circulação*. Pelo resultado a seguir, originalmente proposto por Ford e Fulkerson (1956), adaptado por Bang-Jensen e Bessy (2014), todo fluxo pode ser decomposto em um número pequeno de fluxos simples.

Teorema 2.1 (Teorema da Decomposição de Fluxos - Ford e Fulkerson (1956)). *Todo fluxo x em uma rede \mathcal{N} é a soma de no máximo $n + m$ fluxos caminhos e ciclos. Além disso, os fluxos caminhos podem ser tomados ao longo de caminhos P_1, \dots, P_q tais que P_i inicia em um vértice s_i com $b_x(s_i) > 0$ e termina em um vértice t_i com $b_x(t_i) < 0$, para $1 \leq i \leq q$. Em particular, se $b_x \equiv 0$, não há caminhos e x é a soma de no máximo m fluxos ciclos. Dado um fluxo x , essa decomposição pode ser encontrada em tempo $O(mn)$.*

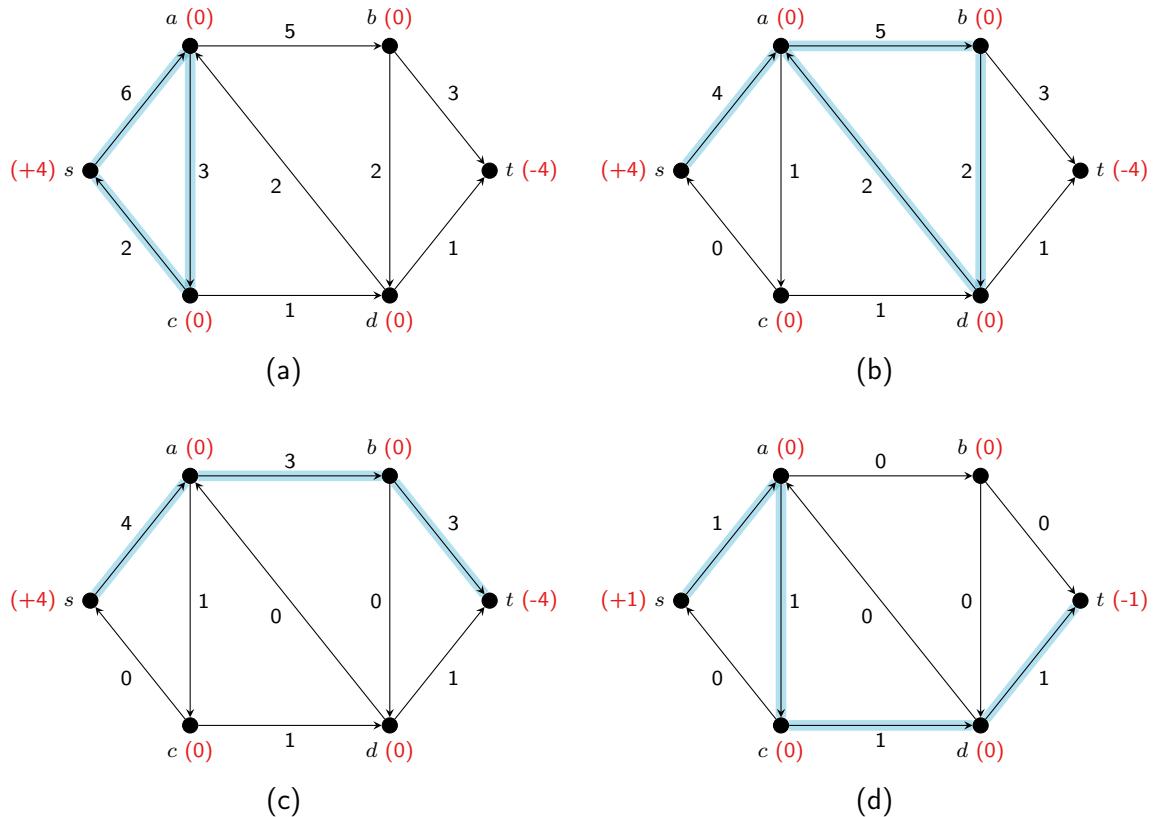
Demonstração. Seja x um fluxo não nulo em \mathcal{N} . Inicialmente, suponha que $b_x(v_0) > 0$ para algum $v_0 \in V$. Segue da Equação (2.2) que existe um arco com cauda v_0 , seja este v_0v_1 , tal que $x(v_0v_1) > 0$. Se $b_x(v_1) < 0$, temos um caminho de v_0 a v_1 . Caso contrário, $b_x(v_1) \geq 0$ e, como $x(v_0v_1) > 0$, existe um arco v_1v_2 com $x(v_1v_2) > 0$. Repetindo-se esse processo, deve-se alcançar um vértice v_k com $b_x(v_k) < 0$ ou que já tenha sido visitado. No primeiro caso, foi encontrado um caminho $P = v_0 \dots v_k$. No segundo, foi encontrado um ciclo $C = v_kv_{k+1} \dots v_k$. Note que, durante esse processo, foram examinados apenas arcos ij com $x(ij) > 0$. O próximo passo é modificar o fluxo x da seguinte maneira:

- i. Se foi encontrado um caminho P de v_0 a v_k , considere $\delta = \min\{x(v_i, v_{i+1}) \mid v_iv_{i+1} \in A(P)\}$ e $\mu = \min\{b_x(v_0), -b_x(v_k), \delta\}$. Tomamos x^P como um fluxo caminho de valor μ sobre P e atualizamos o fluxo x , decrementando μ unidades de fluxo em cada arco em $A(P)$.
- ii. Se foi encontrado um ciclo C , considere $\mu = \min\{x(v_i, v_{i+1}) \mid v_iv_{i+1} \in A(C)\}$. Toma-se x^C como um fluxo ciclo de valor μ sobre C e atualiza-se o fluxo x , decrementando μ unidades de fluxo em cada arco em $A(C)$.

Se após as alterações acima não houver arcos ij com $x(ij) > 0$, o processo é terminado. Caso contrário, é repetido. Se antes do fluxo x se tornar nulo ($x \equiv 0$), todos os vértices estiverem balanceados com respeito ao fluxo x atual, escolhe-se um v_0 tal que exista um arco v_0v_1 com fluxo positivo e, a partir desse ponto, apenas ciclos serão extraídos pelo processo acima. A cada iteração, ou um vértice se torna balanceado com respeito ao fluxo x atual, ou o fluxo em determinado arco fica nulo. Logo, o número total de iterações é no máximo $n + m$. Como a cada extração de fluxo ciclo, pelo menos um arco fica com fluxo nulo, no máximo m fluxos ciclos terão sido extraídos. \square

A Figura 6 traz um exemplo de decomposição de um (s, t) -fluxo de valor 4 em fluxos caminhos e fluxos ciclos. O rótulo de cada arco indica o seu valor de fluxo, e o valor entre parênteses próximo ao rótulo de cada vértice indica o seu balanço em relação ao fluxo. Como nesse tipo de fluxo, apenas o vértice s tem balanço positivo, toda busca é iniciada nele. Como descrito no Teorema 2.1, cada busca se encerra quando for alcançado um vértice já visitado (nesse caso, terá sido encontrado um ciclo) ou quando t (vértice com balanço negativo) for alcançado (nesse caso, terá sido encontrado um (s, t) -caminho). Na figura, os arcos destacados indicam as rotas seguidas a partir de s em cada etapa de uma possível decomposição do fluxo. Em (a) temos o fluxo inicial e uma rota que contém um círculo $(sacs)$. Após a remoção do fluxo ciclo de valor 2 ao longo deste, obtemos o fluxo em (b) e temos uma rota em que novamente foi encontrado um círculo $(abda)$. Após a remoção do fluxo ciclo de valor 2 ao longo deste, obtemos a rede em (c) e temos uma rota que é um caminho $(sabt)$. Após a remoção do fluxo caminho de valor 3 ao longo deste, obtemos o fluxo em (d), que é um fluxo caminho de valor 1.

Figura 6 – Exemplo de decomposição de um (s, t) -fluxo.



Fonte: Elaborada pelo autor.

Observe que, após a remoção de um fluxo ciclo, o balanço dos vértices não é alterado. Consequentemente, a remoção de uma circulação não afeta o valor do fluxo na rede. Este corresponde à soma dos valores dos fluxos caminhos obtidos na decomposição.

2.3.2 Rede residual

Dado um (s, t) -fluxo x em uma rede $\mathcal{N} = (D, l, u)$, com $D = (V, A)$, a *capacidade residual* de um arco $ij \in V \times V$ é o máximo de fluxo adicional que pode ser enviado de i para j . Ela possui duas componentes: a primeira, $(u_{ij} - x_{ij})$, representa a capacidade não utilizada do arco ij pelo fluxo x ; e a segunda, $(x_{ji} - l_{ji})$, a quantidade de fluxo de j para i que pode ser cancelada. Assim, essa capacidade é definida como:

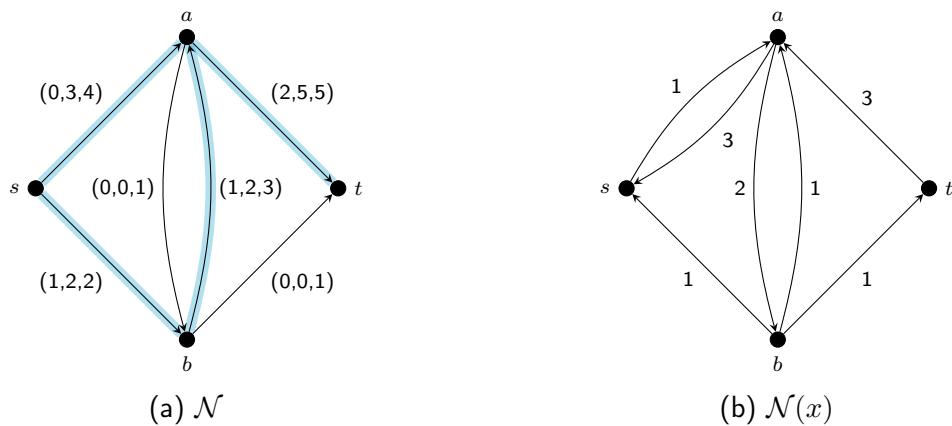
$$r_{ij} = (u_{ij} - x_{ij}) + (x_{ji} - l_{ji}) \quad (2.5)$$

A *rede residual* com respeito ao (s, t) -fluxo x na rede \mathcal{N} é definida como $\mathcal{N}(x) = (D', l' \equiv 0, r)$, onde r é a sua função de capacidade, $D' = (V, A')$ e $A' = \{ij \mid r_{ij} > 0\}$; ou seja, $\mathcal{N}(x)$ contém apenas os arcos com capacidade residual positiva. Esse conceito foi introduzido por Ford e Fulkerson (1956). Pode-se dizer que cada arco ij de \mathcal{N} gera uma capacidade residual em até dois arcos em $\mathcal{N}(x)$, da seguinte maneira:

- a) um arco de avanço ij com capacidade $u_{ij} - x_{ij}$, se $x_{ij} < u_{ij}$;
- b) um arco de retorno ji com capacidade $x_{ji} - l_{ji}$, se $x_{ij} > l_{ji}$.

A Figura 7 traz, em (a) um exemplo de uma rede $\mathcal{N} = (D, l, u)$ com um fluxo x , em que cada arco ij está associado a uma tripla (l_{ij}, x_{ij}, u_{ij}) ; e em (b) a rede $\mathcal{N}(x) = (D', l' \equiv 0, r)$, em que cada arco ij está associado à sua capacidade residual r_{ij} . Para detalhar o cálculo da capacidade residual de um arco, conforme Equação (2.5), observe o arco ab de $\mathcal{N}(x)$, em que $r_{ab} = (u_{ab} - x_{ab}) + (x_{ba} - l_{ba}) = (1 - 0) + (2 - 1) = 2$.

Figura 7 – Rede residual com respeito a um (s, t) -fluxo x em uma rede \mathcal{N} .



Fonte: Elaborada pelo autor.

Um *caminho aumentante* em relação ao um (s, t) -fluxo x em uma rede \mathcal{N} é um caminho direcionado de s a t em $\mathcal{N}(x)$. A *capacidade residual de um caminho aumentante* P é definida por $\delta(P) = \min\{r_{ij} \mid ij \in A(P)\}$, ou seja, é o maior valor que pode ser incrementado ao fluxo x ao longo de P , de modo que x continue sendo viável na rede. Diz-se que $ij \in A(P)$ é um *arco crítico*, se $r_{ij} = \delta(P)$. Na Figura 7, em $\mathcal{N}(x)$, $sabt$ é um caminho aumentante de capacidade residual 1 e seus arcos críticos são sa e bt .

Pela definição a seguir, se x é um fluxo em uma rede $\mathcal{N} = (D, u)$ e x' é um fluxo em $\mathcal{N}(x)$, então é possível “adicionar” x' a x e obter um novo fluxo em \mathcal{N} . A operação de adição é feita arco a arco e deve-se considerar que x e x' são fluxos líquidos.

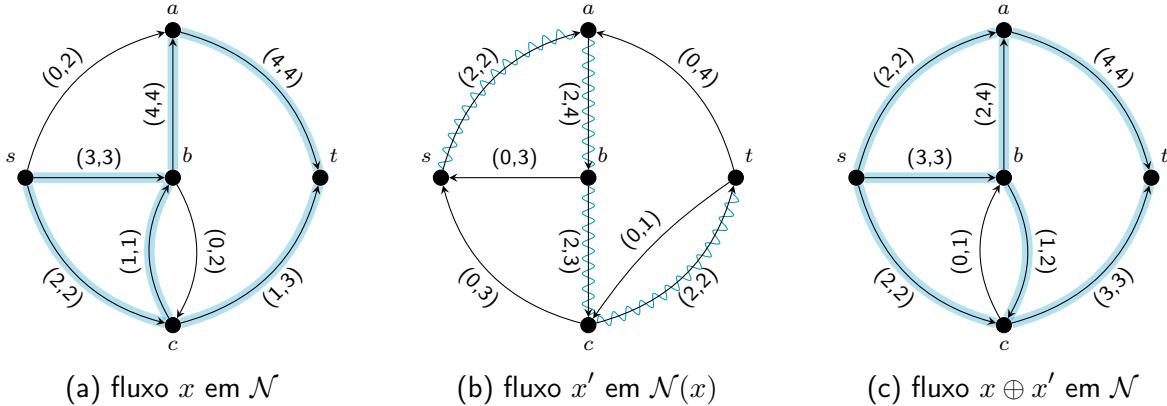
Sejam x um fluxo em uma rede \mathcal{N} e x' um fluxo em $\mathcal{N}(x)$. Define-se o fluxo $x^* = x \oplus x'$ em \mathcal{N} fazendo inicialmente $x_{ij}^* = x_{ij}$, para todo arco ij em \mathcal{N} , e para todo ij em $\mathcal{N}(x)$, tal que $x'_{ij} > 0$, alterando o fluxo x^* como segue:

- a) Se $x_{ji} = 0$, então $x_{ij}^* := x_{ij} + x'_{ij}$;
- b) Se $x_{ji} > 0$, $x_{ij} = 0$ e $x_{ji} < x'_{ij}$, então $x_{ij}^* := x'_{ij} - x_{ji}$ e $x_{ji}^* := 0$;
- c) Se $x_{ji} \geq x'_{ij}$, então $x_{ji}^* := x_{ji} - x'_{ij}$.

Caso x' não possua arcos de retorno de x , a operação $x \oplus x'$ consiste em somar os valores de fluxo de x' aos de x , arco a arco.

Os três diferentes casos da definição acima estão ilustrados na Figura 8. Ela traz, em (a), um fluxo x em \mathcal{N} ; em (b), um fluxo x' em $\mathcal{N}(x)$; e em (c), o fluxo resultante de $x \oplus x'$ em \mathcal{N} . Cada arco está associado a um par de valores. O primeiro indica o valor de fluxo e o segundo indica a capacidade (para a rede \mathcal{N}) ou a capacidade residual (para a rede $\mathcal{N}(x)$). Em (b), estão destacados os arcos ij em que $x'_{ij} > 0$. Nela, é possível observar todos os casos da definição acima. O primeiro caso acontece nos arcos sa e ct ; o segundo acontece no arco bc e o terceiro acontece no arco ab .

Figura 8 – Exemplo de adição de fluxos $x \oplus x'$.

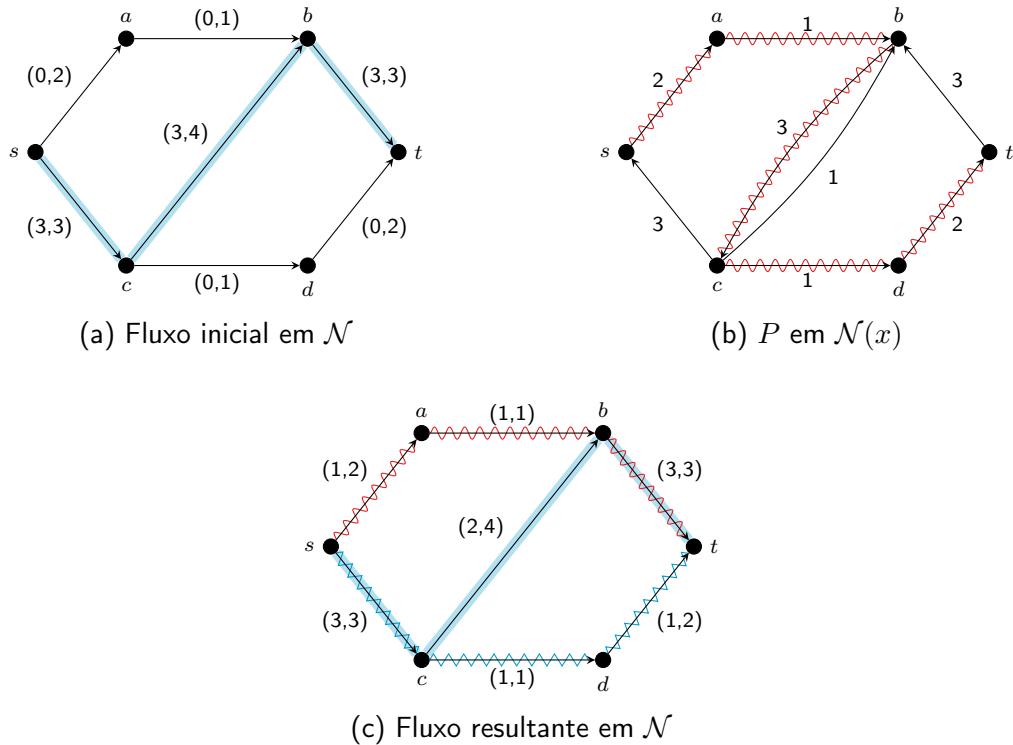


Fonte: Elaborada pelo autor.

Considere um (s, t) -fluxo x em uma rede \mathcal{N} e um caminho aumentante P em $\mathcal{N}(x)$. Seja x' o fluxo caminho ao longo de P ($|x'| = \delta(P)$). É possível incrementar $\delta(P)$ unidades ao valor de x , fazendo $x \oplus x'$. Como os arcos de P são conhecidos, é possível identificar facilmente os arcos em \mathcal{N} para os quais o valor do fluxo deverá ser alterado. Embora o valor do fluxo na rede seja aumentado, em alguns arcos ele pode ser decrementado para viabilizar o envio de mais unidades de fluxo de s a t . Observe na Figura 8 que o fluxo em $\mathcal{N}(x)$ é um caminho aumentante de capacidade residual 2.

Seja um (s, t) -fluxo x em uma rede \mathcal{N} . Note que, quando $|x| = 0$, $\mathcal{N}(x) = \mathcal{N}$. Nesse caso, um caminho aumentante em $\mathcal{N}(x)$ é necessariamente um caminho direcionado em \mathcal{N} . A partir do segundo caminho aumentante, isso pode não acontecer. Portanto, a adição de um caminho aumentante a um (s, t) -fluxo pode provocar, neste, o surgimento de dois (s, t) -fluxos caminhos. A Figura 9 ilustra isso. Ela traz, em (a), uma rede \mathcal{N} e um fluxo caminho x de valor 3; em (b), um caminho aumentante P em $\mathcal{N}(x)$, com $\delta(P) = 1$; e em (c) o fluxo resultante em \mathcal{N} após encontrar P em $\mathcal{N}(x)$ e aumentar o fluxo x , que passa a ter valor 4. Cada arco ij de \mathcal{N} está associado a uma dupla (x_{ij}, u_{ij}) e cada arco ij de $\mathcal{N}(x)$ está associado à sua capacidade residual r_{ij} . Note que P usa o arco de retorno bc , provocando a redução de uma unidade de fluxo no arco cb e redirecionando-a pelo arco cd . Com isso, é possível enviar mais uma unidade de fluxo em \mathcal{N} , pelo caminho $sabt$. Apesar de terem sido somados dois fluxos caminhos, isso gerou três fluxos caminhos na rede original, que são: $sabt$, $scbt$ e $scdt$, cujos valores são, respectivamente, 1, 2 e 1.

Figura 9 – Efeito causado no fluxo original por um caminho aumentante.



Fonte: Elaborada pelo autor.

2.3.3 Fluxo máximo e corte mínimo

Um (s, t) -corte em uma rede $\mathcal{N} = (D, u)$, onde $D = (V, A)$, é um conjunto de arcos da forma $(S, \bar{S}) = \{uv \in A \mid u \in S \text{ e } v \in \bar{S}\}$, com $s \in S$ e $t \in \bar{S}$, em que S e \bar{S} formam uma partição de V ($S \cap \bar{S} = \emptyset$ e $S \cup \bar{S} = V$). A capacidade de um (s, t) -corte, denotada por $u(S, \bar{S})$, é a soma das capacidades dos seus arcos.

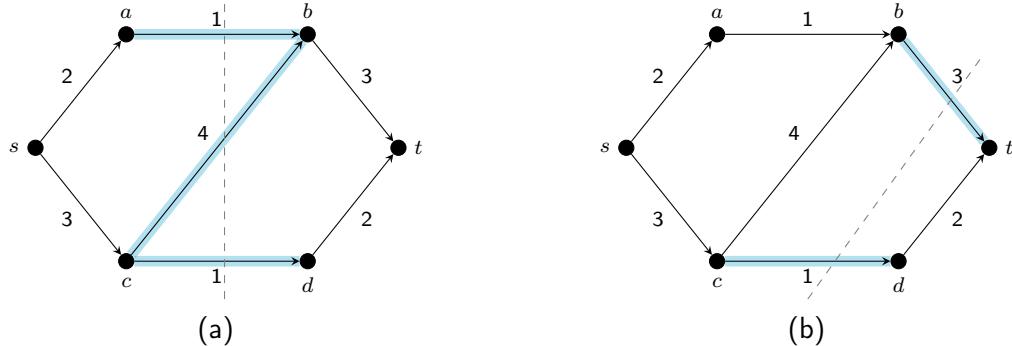
O *Problema do Corte Mínimo* consiste em encontrar um (s, t) -corte (S', \bar{S}') de capacidade mínima em uma rede. Trata-se da versão dual do *Problema do Fluxo Máximo*, que consiste em encontrar o maior valor de um (s, t) -fluxo em uma rede.

O Teorema 2.2 traz um resultado fundamental da Teoria de Fluxos em Redes, conhecido com o *Teorema do Fluxo Máximo e Corte Mínimo*.

Teorema 2.2 (Teorema do fluxo máximo e corte mínimo - Ford e Fulkerson (1956)). *Em qualquer rede \mathcal{N} , o valor do fluxo máximo é igual à capacidade do corte mínimo.*

Na Figura 10, os arcos destacados são exemplos de (s, t) -corte. O rótulo de cada arco indica a sua capacidade. Em (a), um corte de capacidade 6, e em (b) um corte de capacidade 4. Este é mínimo, logo corresponde ao valor de um (s, t) -fluxo máximo.

Figura 10 – Exemplos de (s, t) -corte.



Fonte: Elaborada pelo autor.

Considere uma rede $\mathcal{N} = (D, u)$, com $D = (V, A)$, e um (s, t) -fluxo x nesta. Se x é máximo, então não há mais caminhos aumentantes em $\mathcal{N}(x)$. A prova do Teorema 2.2 sugere um método simples para encontrar um (s, t) -fluxo máximo em \mathcal{N} , chamado de *Método de Ford e Fulkerson*. Para mais detalhes sobre esse método, bem como suas limitações, implementações e complexidades de cada uma, consultar o Apêndice B.

O Teorema 2.2 implica o *Teorema de Menger*, um outro resultado clássico da Teoria dos Grafos, enunciado a seguir:

Teorema 2.3 (Menger (1927)). *Sejam D um (multi)digrafo e $s, t \in V(D)$, com $s \neq t$. O número máximo de (s, t) -caminhos arco-disjuntos é igual ao número mínimo de arcos que, se removidos, desconectam os vértices s e t .*

Isso pode ser observado considerando uma rede de capacidade unitária, $\mathcal{N} = (D, u \equiv 1)$. A quantidade de (s, t) caminhos arco-disjuntos é dada pelo valor de um fluxo máximo x . Pelo Teorema 2.2, $|x|$ é igual à capacidade de um (s, t) -corte mínimo (S, \bar{S}) que, nesse caso, corresponde à quantidade de arcos deste.

3 FLUXOS K-DIVISÍVEIS

Em problemas clássicos de fluxos, o produto enviado da origem ao destino pode precisar passar por muitos caminhos na rede. Isso pode ser indesejado, ou até mesmo proibido, em algumas aplicações como nas redes de comunicação, ou no transporte de mercadorias, entre outras. Nesse contexto surge o conceito de fluxos divisíveis, que será formalizado neste capítulo, juntamente com algumas variações e resultados de complexidade relacionados a cada uma destas. Com exceção do Teorema 3.8, todos os resultados deste capítulo são de terceiros.

3.1 O Problema e suas variações

Segundo Kleinberg (1996), o Problema do Fluxo Máximo e Corte Mínimo baseia-se em um outro resultado fundamental da Teoria dos Grafos - o Teorema de Menger. Este, por sua vez, fornece uma boa caracterização para o seguinte problema:

Problema: Caminhos disjuntos com fonte única

Entrada: um grafo G com um vértice fonte s e vértices terminais t_1, \dots, t_k

Pergunta: existem caminhos disjuntos em arestas de s a t_i , para $1 \leq i \leq k$?

Considere um conjunto de mercadorias (em inglês, *commodity*) $C_i = (s_i, t_i, d_i)$, com $1 \leq i \leq k$, em que s_i , t_i e d_i representam, respectivamente, a origem, o destino e a demanda da mercadoria i . O problema de *fluxo de multicommodities* consiste em encontrar um fluxo que satisfaça todas as demandas de cada mercadoria entre uma origem e um destino, respeitando as capacidades dos arcos da rede. Em Kleinberg (1996), foi introduzido o problema de *fluxos indivisíveis*, como uma generalização para o problema acima, que é uma versão restrita do fluxo de *multicommodities* em que $s_1 = \dots = s_k = s$ e a demanda de cada mercadoria deve ser enviada por um único caminho.

Problema: Fluxo indivisível com fonte única

Entrada: um grafo G (direcionado ou não, em que cada aresta tem capacidade pelo menos 1) com uma fonte s e terminais t_i , cada um destes com uma demanda $d_i \leq 1$, para $1 \leq i \leq k$

Pergunta: é possível escolher um único (s, t_i) -caminho, para cada i , de modo que o conjunto resultante de caminhos respeite as restrições de capacidade?

Como observado pelo autor, trata-se de uma variante básica e \mathcal{NP} -Difícil do Problema do Fluxo Máximo. Ele complementa ainda que o problema compreende vários problemas \mathcal{NP} -Completos, como os de particionamento, escalonamento, empacotamento, roteamento em circuitos virtuais etc.

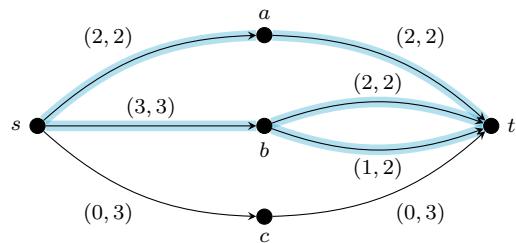
Em Baier, Köhler e Skutella (2005), foi proposta uma generalização para o problema de fluxos indivisíveis, que pode ser aplicada a uma ou mais mercadorias e a demanda de cada uma destas deve ser enviada por um número restrito e possivelmente diferente de caminhos. Trata-se do problema de *fluxos divisíveis*. Problemas desse tipo acontecem, por exemplo, em redes de comunicação, onde os clientes podem demandar conexões com determinadas capacidades entre dados pares de nós. Se essas capacidades são muito altas, fica inviável para o administrador da rede atendê-las de forma indivisível. Por outro lado, o cliente pode não ter interesse em lidar com muitas conexões, cada uma destas com pequenos valores de capacidade.

Segundo Baier, Köhler e Skutella (2005), um (s, t) -fluxo x em uma rede \mathcal{N} é k -divisível se puder ser representado por k pares $(P_1, f_1), \dots, (P_k, f_k)$, onde cada par representa um (s, t) -caminho P_i , com fluxo f_i ao longo deste; ou seja, se x puder ser decomposto em k fluxos caminhos $x^1 \dots x^k$, de modo que $\sum_{i=1}^k |x^i| = |x|$. Os caminhos dos fluxos $x^1 \dots x^k$ não precisam ser distintos. Assim, para valores reais, um fluxo k -divisível também é k' -divisível, para $k' > k$.

Como consequência do Teorema 2.1 (Teorema da Decomposição de Fluxos), todo (s, t) -fluxo pode ser decomposto, em tempo polinomial, em no máximo m fluxos caminhos de s a t e uma circulação. Com exceção desta, pode-se dizer que todo fluxo é m -divisível. Portanto, estamos interessados em valores de k menores que m . No contexto de fluxos k -divisíveis, podemos citar os seguintes problemas:

1. Dados uma rede \mathcal{N} e um inteiro positivo k , encontrar o maior valor de fluxo que seja k -divisível.
2. Dados uma rede \mathcal{N} e um (s, t) -fluxo x , encontrar o menor valor de k , tal que x é k -divisível.

Figura 11 – Exemplo de fluxo k -divisível.



Fonte: Elaborada pelo autor.

Na Figura 11, cada arco ij está associado a um par (x_{ij}, u_{ij}) representando seus valores de fluxo e de capacidade. Se desejarmos saber qual o valor de um fluxo máximo 2-divisível, a resposta é 5. Isso pode ser obtido enviando 2 unidades de fluxo ao longo do caminho sat , e 3 unidades ao longo do caminho sct . Por outro lado, se considerarmos o fluxo dado x , cujo valor é 5, e perguntarmos qual o menor valor de k tal que x k divisível, a resposta é 3. Esses problemas serão formalizados e analisados nas Seções 3.2 e 3.3.

3.2 Maximizando o fluxo enviado por um número dado de caminhos

Nesta seção, serão apresentados resultados de Baier, Köhler e Skutella (2005) sobre a complexidade do *Problema do Fluxo Máximo k-Divisível*. Segundo os autores, este problema corresponde ao caso de uma única mercadoria cuja demanda deve ser roteada por no máximo k caminhos de s a t . Este problema é fortemente \mathcal{NP} -Completo, pois contém o caso do fluxo indivisível com fonte única, de Kleinberg (1996), como um caso especial. Dada uma instância desse problema, um digrafo D com uma fonte s e sumidouros t_1, \dots, t_k e as respectivas demandas d_1, \dots, d_k , adiciona-se um novo sumidouro t , com arcos t_i, t com capacidade d_i , para $1 \leq i \leq n$. Assim, calcular o fluxo k -divisível de valor igual a $\sum_{i=1}^k d_i$ equivale a resolver o problema de fluxo indivisível de fonte única.

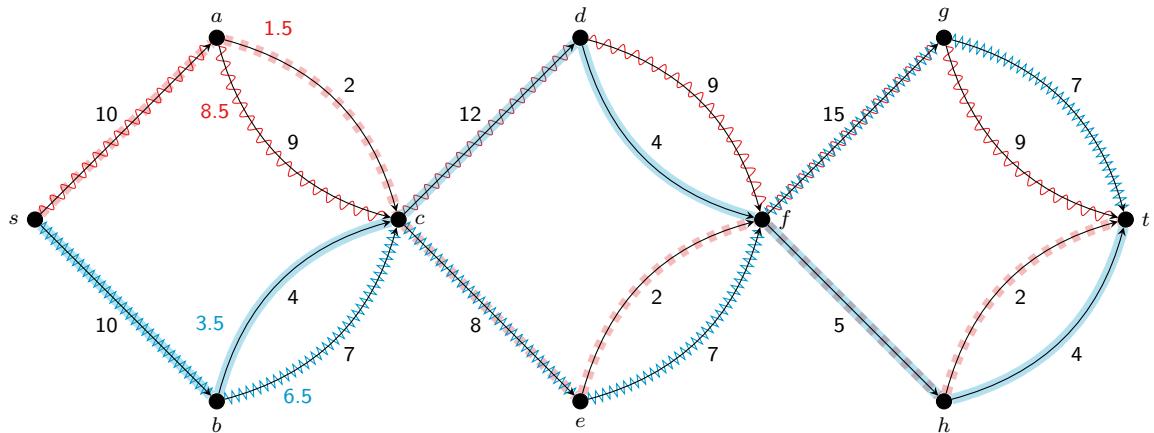
A versão de decisão do Problema do Fluxo Máximo k -Divisível pode ser formalizada da seguinte maneira:

Problema: FluxoMaxKDiv

Entrada: uma rede $\mathcal{N} = (D, u)$, um inteiro positivo k e um valor de fluxo f .

Pergunta: \mathcal{N} admite um fluxo k -divisível de valor pelo menos f ?

Figura 12 – Exemplo de fluxo máximo 4-divisível não inteiro.



Fonte: Adaptada de Baier, Köhler e Skutella (2005, p. 244)

Segundo Baier, Köhler e Skutella (2005), mesmo em redes com capacidades inteiras, o (s, t) -fluxo máximo k -divisível pode não ser obtido a partir de fluxos caminhos inteiros. Na rede da Figura 12, é possível obter um fluxo 4-divisível de valor 20, com fluxos caminhos de valores 1.5, 8.5, 3.5 e 6.5. Os autores mostram que o valor de pelo menos um fluxo caminho deve estar entre 1 e 2. Note que não existe um (s, t) -fluxo 3-divisível de valor 20. Suponha que a solução admita um fluxo caminho de valor $x < 1$. Isso forçaria um fluxo caminho de valor $10 - x > 9$, mas a capacidade residual máxima de um caminho na rede é 9. Agora, assuma que todos os fluxos caminhos da solução possuam valor $x > 2$. Com isso, seria possível remover todos os arcos cuja capacidade é no máximo 2, o que inviabilizaria a obtenção de um

(s, t) -fluxo de valor 20. Por fim, suponha que exista pelo menos um fluxo caminho de valor 1 (ou 2). Isso força um fluxo caminho de valor 9 (ou 8) pelo arco sa . Estes forçam um fluxo caminho de valor 3 (ou 4) pelo arco cd , o que fixaria o valor do quarto fluxo caminho em 7 (ou 6). No entanto, não é possível agrupar esses valores, em pares, para que sejam roteados pelos arcos fg e fh .

Seja $FluxoMax2Div$ o problema $FluxoMaxKDiv$ restrito a $k = 2$. Os autores de Baier, Köhler e Skutella (2005), inicialmente, apresentaram formas de obter um limite superior, bem como um bom fator de aproximação, para o problema $FluxoMax2Div$. Isso é feito, utilizando algoritmos de caminho aumentante para obtenção do fluxo máximo (ver Apêndice B). Estes resultados estão descritos a seguir.

Lema 3.1 (Baier, Köhler e Skutella (2005)). *Após duas iterações do algoritmo de caminho aumentante genérico, o fluxo resultante pode ser decomposto na soma de um fluxo 3-divisível e uma circulação.*

Isso decorre do fato de que o segundo caminho aumentante não é necessariamente um caminho direcionado na rede original. Como observado na Figura 9, página 26, ele pode provocar, na rede original, o surgimento de três fluxos caminhos. Além destes, pelo Teorema 2.1 (Teorema da Decomposição de Fluxo), pode surgir uma circulação.

Um limite superior para o valor do fluxo máximo 2-divisível é obtido tomando, sucessivamente, o primeiro e o segundo caminho aumentante de maior capacidade residual (ver Apêndice B - Algoritmo 3), o que pode ser visto no resultado a seguir.

Lema 3.2 (Baier, Köhler e Skutella (2005)). *Após duas iterações do algoritmo de caminho aumentante de capacidade máxima, o valor do fluxo resultante é um limite superior para o fluxo máximo 2-divisível.*

Demonstração. Dada uma rede $\mathcal{N} = (D, u)$, seja x^* um o fluxo máximo 2-divisível em \mathcal{N} que pode ser decomposto em $\{(P_1^*, f_1^*), (P_2^*, f_2^*)\}$. Além disso, sejam P_1 e P_2 os caminhos aumentantes de maior capacidade residual, com valores de fluxo, respectivamente, f_1 e f_2 . Temos que mostrar que $f_1 + f_2 \geq f_1^* + f_2^*$.

Seja $\mathcal{N}' = (D', u')$, onde $D' = [A(P_1) \cup A(P_1^*) \cup A(P_2^*)]$. Defina a capacidade u' de cada arco de D' como sendo o menor valor de modo que tanto a solução ótima $\{(P_1^*, f_1^*), (P_2^*, f_2^*)\}$ como o fluxo f_1 ao longo do caminho P_1 sejam viáveis. Seja x^1 o fluxo de valor f_1 ao longo de P_1 e considere $\mathcal{N}'(x^1)$. Vamos mostrar que existe um caminho aumentante de capacidade residual pelo menos $f_1^* + f_2^* - f_1$ nesta, produzindo um fluxo caminho de valor $f_2 \geq f_1^* + f_2^* - f_1$.

Se $f_1^* + f_2^* - f_1 \leq 0$, então não há nada a ser mostrado. Assim, podemos assumir que $f_1 < f_1^* + f_2^*$. Além disso, pela escolha de P_1 , sabemos que $f_1 \geq \max\{f_1^*, f_2^*\}$. Seja P um caminho aumentante de capacidade residual máxima em $\mathcal{N}'(x^1)$ e seja a um arco crítico de P . Note que $r_a > 0$, pois $f_1^* + f_2^* > f_1$. Distinguimos os três casos a seguir:

Caso 1: a é um arco de avanço de P_1 . Uma vez que $r_a > 0$, ambos P_1^* e P_2^* devem conter a , de modo que $r_a = f_1^* + f_2^* - f_1$.

Caso 2: a é um arco de retorno de P_1 . Nesse caso, $r_a = f_1 \geq f_1^* + f_2^* - f_1$.

Caso 3: $a \notin A(P_1)$. Nesse caso, $r_a \geq \min\{f_1^*, f_2^*\} \geq f_1^* + f_2^* - f_1$. \square

Os Lemas 3.1 e 3.2, juntos, produzem o seguinte resultado de aproximação para o problema do fluxo máximo 2-divisível:

Teorema 3.3 (Baier, Köhler e Skutella (2005)). *Existe um algoritmo 2/3-aproximativo para o problema do fluxo máximo 2-divisível, com tempo de execução $O(m \log m)$, onde m é o tamanho do digrafo sobre o qual a rede está definida.*

Demonstração. Executando as duas primeiras iterações do algoritmo de caminho aumentante de capacidade máxima, o fluxo resultante pode ser decomposto em 3 fluxos caminhos e uma circulação, pelo Lema 3.1. Removendo-se a circulação e o caminho com a menor capacidade residual, obtém-se um fluxo 2-divisível. O fator de aproximação 2/3 é uma consequência imediata do Lema 3.2. Finalmente, o tempo de execução é o tempo de encontrar os caminhos aumentantes, que é $O(m \log m)$, pelo Algoritmo de rotulação de Dijkstra modificado. \square

Isso pode ser generalizado para $k \geq 3$, como mostra o resultado a seguir:

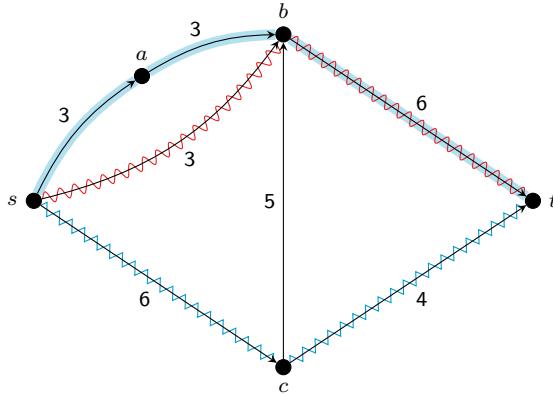
Corolário 3.4 (Baier, Köhler e Skutella (2005)). *Existe um algoritmo 2/k-aproximativo para o problema do fluxo máximo k-divisível, para $k \geq 3$, com tempo $O(m \log m)$, onde m é o tamanho do digrafo sobre o qual a rede está definida.*

Demonstração. Novamente, após as duas primeiras iterações do algoritmo de caminho aumentante de capacidade máxima, o fluxo resultante pode ser decomposto em 3 fluxos caminhos e uma circulação. Pelo Lema 3.2, o valor do fluxo 3-divisível resultante é pelo menos o valor do fluxo máximo em dois caminhos. Logo, o valor de um fluxo máximo 2-divisível é pelo menos uma fração 2/k do valor de um fluxo máximo k-divisível. \square

A abordagem de pegar os k primeiros caminhos aumentantes de maior capacidade residual, como forma de encontrar um limite superior para o fluxo máximo k -divisível, não funciona para $k \geq 3$. Isso pode ser comprovado pela rede da Figura 13. O primeiro caminho aumentante seria $scbt$, de capacidade residual 5. O segundo seria $sabct$ ou $sbct$, ambos de capacidade residual 3. Por fim, como as capacidades residuais dos arcos bt e ct são iguais a 1, esta seria naturalmente a capacidade residual do terceiro caminho aumentante. O valor do fluxo obtido seria 9, mas o fluxo máximo 3-divisível, representado pelos três caminhos destacados, tem valor 10.

Os autores mostraram que o fator de aproximação 2/3 (ver Teorema 3.3) para o *FluxoMax2Div* é justo, ou seja, não existe um algoritmo aproximativo para esse problema com um fator de aproximação $2/3 + \varepsilon$, para qualquer $\varepsilon > 0$, a menos que $\mathcal{P} = \mathcal{NP}$.

Figura 13 – Exemplo de fluxo 3-divisível.



Fonte: Adaptada de Baier, Köhler e Skutella (2005, p. 238)

Teorema 3.5 (Baier, Köhler e Skutella (2005)). *É N P-difícil aproximar instâncias do problema FluxoMax2Div com fator de aproximação melhor que 2/3.*

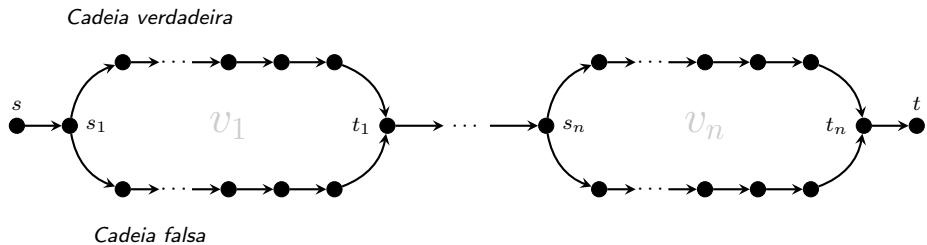
Demonstração. É dada uma redução a partir do Problema SAT. Dada uma instância qualquer deste - uma fórmula φ com n variáveis e m cláusulas - construímos um digrafo D , com uma fonte s e um sumidouro t , sobre o qual será definida uma rede \mathcal{N} . Em seguida mostramos que, se a instância φ do SAT for satisfatível, há um (s, t) -fluxo 2-divisível de valor 3. Caso contrário, o valor fluxo máximo 2-divisível que pode ser obtido na rede é igual a 2. Desse modo, um algoritmo approximativo com fator de aproximação melhor que 2/3 implicaria a existência de um algoritmo polinomial para decidir se φ é satisfatível.

Primeiro, será descrita a construção do subdigrafo para encontrar uma atribuição de valores verdade às variáveis v_1, \dots, v_n de φ que a tornam verdadeira. Este é acíclico, possui $n + 1$ pontes, uma fonte s e um sumidouro t . Para cada variável v_i , há dois vértices s_i e t_i , o primeiro é cabeça da ponte i e o segundo é cauda da ponte $i + 1$ ($s_1 = s$ e $t_{n+1} = t$). Adicionalmente, há um segmento construído por dois caminhos paralelos, entre pontes consecutivas. Um destes é chamado de *cadeia verdadeira* e o outro de *cadeia falsa* de v_i . Seja $o(\ell)$ o número de ocorrências do literal ℓ , a cadeia verdadeira (resp. falsa) possui tamanho $2o(\bar{v}_i)$ (resp. $2o(v_i)$). Todos os arcos descritos neste passo têm capacidade igual a 2. Na cadeia verdadeira (resp. falsa) de cada variável v_i , são reservados dois arcos consecutivos para cada cláusula em que ocorre o literal \bar{v}_i (resp. v_i). Chamamos o primeiro arco de cada um destes pares de *arco representativo* da cláusula correspondente. A ordem das cláusulas em cada cadeia é monotonicamente decrescente pelos seus índices. Se um (s, t) -caminho com fluxo de valor 2 usa a cadeia verdadeira (resp. falsa) de uma variável v_i , então isso será interpretado como a atribuição do valor 1 (resp. 0) à variável v_i . Essa construção está ilustrada na Figura 14.

Na segunda etapa, construímos um subdigrafo para codificar as cláusulas. A ideia é construir um segundo (s, t) -caminho usando os arcos representativos. Este caminho fica bloqueado se o assinalamento dado pelo primeiro não torna a fórmula verdadeira. Para cada

cláusula C_i , há dois vértices u_i e w_i , com arcos $w_{i-1}u_i$, para $1 \leq i \leq m+1$ ($w_0 = s$ e $u_{m+1} = t$). Para cada arco representativo ab de uma cláusula C_i , com $1 \leq i \leq m$, adicionam-se arcos u_ia e bw_i . Todos os arcos descritos nesta etapa possuem capacidade unitária. A construção completa está ilustrada no exemplo da Figura 15.

Figura 14 – Redução do SAT para o Fluxo Máximo 2-Divisível (primeira etapa).



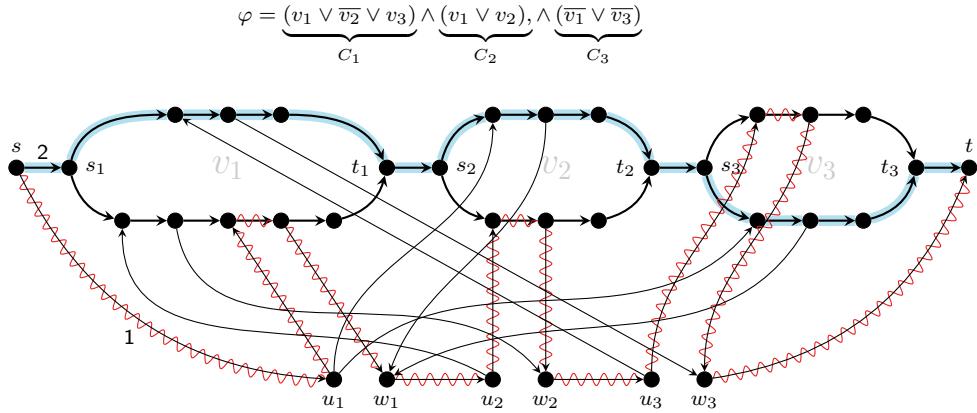
Fonte: Elaborada pelo autor.

Assuma que a fórmula φ , instância do SAT , seja satisfatível. Considere uma atribuição de valores verdade às suas variáveis de modo que ela seja verdadeira. Enviamos 2 unidades de fluxo de s a t , através do subdigrafo de variáveis. Para cada variável v_i , com $1 \leq i \leq n$, o caminho tomado deve usar a cadeia verdadeira de v_i , se o valor de v_i for 1 (verdadeiro), ou a cadeia falsa, caso contrário. Pela construção do digrafo, para cada cláusula C_j , com $1 \leq j \leq m$, há um caminho de u_j para w_j , via arco representativo de C_j nas cadeias verdadeira ou falsa de cada variável que a compõem. Este caminho é disjunto do que fora escolhido inicialmente, com base no valor verdade das variáveis. Desse modo, é possível obter um (s, t) -caminho $su_1 \dots w_1 u_2 \dots w_2 \dots u_n \dots w_n t$ de valor 1, que é disjunto caminho de valor 2, escolhido inicialmente. Consequentemente, a rede admite um (s, t) -fluxo 2-divisível de valor 3. Além disso, pela capacidade dos arcos com cauda s , este é maximal. Resta mostrar que um (s, t) -fluxo 2-divisível de valor maior que 2 produz uma atribuição de valores às variáveis de φ que a tornam verdadeira. Primeiro, note que os dois caminhos devem ser arco-disjuntos, pois, do contrário, o valor do fluxo máximo seria 2. Segundo, um dos caminhos deve passar completamente pelos arcos de capacidade 2. Caso contrário, o valor do fluxo em cada caminho seria no máximo 1.

O caminho P_1 , com valor de fluxo maior que 1, é usado para determinar os valores verdade das variáveis. Como mencionado, este caminho está completamente no subdigrafo das variáveis, construído na primeira etapa. Portanto, para cada variável v_i , P_1 deve usar todas as arestas de uma das cadeias (verdadeira ou falsa) de v_i . O caminho P_2 , por outro lado, deve deixar s pelo arco su_1 e pelos arcos representativos da cláusula C_i , em ordem crescente do índice destas, e pelos arcos $w_i u_{i+1}$, para $1 \leq i \leq m$ (considere $u_{i+1} = t$). Esse percurso consecutivo de arcos representativos seguindo a ordem crescente do índice das cláusulas é garantido pela ordenação decrescente dos arcos representativos ao longo de cada segmento de variável. Consequentemente, P_2 não pode pular qualquer par de vértices u_i e w_i e assim se verifica que o assinalamento satisfaz φ . \square

A Figura 15 traz um exemplo da redução apresentada no Teorema 3.5. Observe que, com os valores $v_1 = 1$, $v_2 = 1$ e $v_3 = 0$, a fórmula φ é satisfeita. Isso está representado pelos dois (s, t) -caminhos destacados, um de valor 2 e outro de valor 1.

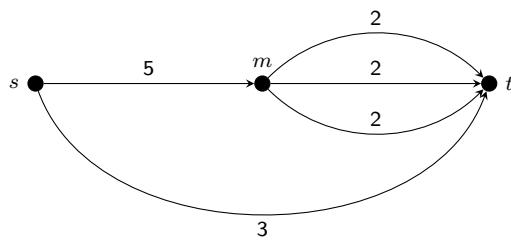
Figura 15 – Redução do *SAT* para o *FluxoMax2Div*.



Fonte: Elaborada pelo autor.

Uma variação do problema consiste em determinar o valor do maior fluxo que pode ser decomposto em k caminhos, todos de mesmo valor. Trata-se do Problema do *Fluxo MÁximo Uniformemente k -Divisível*. Note que o fluxo obtido não é necessariamente uniforme. Segundo Baier, Köhler e Skutella (2005), isso pode ser resolvido em tempo $O(km \log m)$. Mesmo considerando valores inteiros de capacidades, a solução para esse problema pode não ser inteira. No exemplo da Figura 16, o valor máximo de um fluxo uniformemente 4-divisível é $20/3$, que corresponde a 4 fluxos caminhos de valor $5/3$, cada.

Figura 16 – Rede com fluxo máximo 4-divisível uniforme não inteiro.



Fonte: Elaborada pelo autor.

De acordo com Baier, Köhler e Skutella (2005), o problema de encontrar o fluxo máximo 2-divisível permanece \mathcal{NP} -difícil, mesmo quando a diferença entre os valores de fluxo nos caminhos é pequena, ou seja, quando os fluxos caminhos possuem valores muito próximos e a decomposição é, portanto, quase uniforme.

Corolário 3.6 (Baier, Köhler e Skutella (2005)). *Para um ε arbitrário e positivo, é fortemente \mathcal{NP} -difícil calcular o (s, t) -fluxo máximo 2-divisível, mesmo que os valores dos fluxos nos caminhos difiram por um fator de no máximo $1 + \varepsilon$.*

Segundo Baier, Köhler e Skutella (2005), é fácil de encontrar uma solução aproximada para o problema *FluxoMax2Div* nas condições apresentadas no Corolário 3.6. Note que o fluxo máximo uniforme k -divisível é claramente uma solução $1/(1 + \varepsilon)$ -aproximada para o *FluxoMax2Div*.

Em Baier, Köhler e Skutella (2005), os autores mencionam ainda um problema relacionado ao *FluxoMaxKDiv*, no qual, em vez de buscar um (s, t) -fluxo de valor máximo, deseja-se saber se é possível enviar determinada quantidade de fluxo utilizando no máximo k caminhos. Como observado pelos autores, esse problema é fortemente \mathcal{NP} -Completo. Isso pode ser visto a partir da construção apresentada na demonstração do Teorema 3.5. Nesta, a fórmula φ do *SAT* é satisfatível se e somente se o menor número de caminhos necessários para enviar 3 unidades de fluxo na rede \mathcal{N} for igual a 2. Daí, segue o resultado:

Corolário 3.7 (Baier, Köhler e Skutella (2005)). *Calcular um (s, t) -fluxo de um dado valor usando um número mínimo de caminhos é um problema fortemente \mathcal{NP} -Difícil.*

3.3 Minimizando o número de caminhos de um dado fluxo

Muitos algoritmos de fluxo produzem como saída um conjunto de valores associados aos arcos da rede. No entanto, é mais natural e conveniente ver um fluxo como um conjunto de caminhos entre os vértices de origem e de destino. No transporte de mercadorias que devem ser armazenadas em contêiners, por exemplo, cada contêiner pode ser visto como um caminho. Minimizar a quantidade destes implica minimizar a quantidade de contêiners necessários para que o fluxo da mercadoria.

Formalmente, o problema corresponde a determinar o menor valor inteiro k de modo que um dado (s, t) -fluxo x em uma rede \mathcal{N} seja k -divisível. Esse problema é referenciado aqui como *FluxoMinDiv*, cuja versão de decisão está formalizada a seguir:

Problema: *FluxoKDiv*

Entrada: Uma rede $\mathcal{N} = (D, u)$, com $D = (V, A)$, um (s, t) -fluxo x e um $k \in \mathbb{Z}_+^*$.

Pergunta: O fluxo x é k -divisível?

Segundo Vatinlen et al. (2008), esse problema é \mathcal{NP} -Completo mesmo em DAGs. De forma independente, mostramos com o Teorema 3.8 um resultado equivalente, a partir do *Problema 3-Partição*, definido em Garey e Johnson (1979) da seguinte maneira:

Problema: *3-Partição*

Entrada: Um conjunto de objetos $A = \{a_1, a_2, \dots, a_{3r}\}$ e um inteiro positivo T , tal que o valor de cada objeto a_i , denotado por $v(a_i)$, é um inteiro positivo entre $T/4$ e $T/2$ e $\sum_{i=1}^{3r} v(a_i) = rT$.

Pergunta: É possível partitionar o conjunto A em r subconjuntos de 3 elementos, tal que a soma dos valores dos objetos de cada subconjunto seja T ?

Em Garey e Johnson (1979, p. 99) foi mostrado que o Problema 3-Partição é fortemente \mathcal{NP} -Completo.

Teorema 3.8. *O problema FluxoKDiv é fortemente \mathcal{NP} -Completo mesmo em DAGs.*

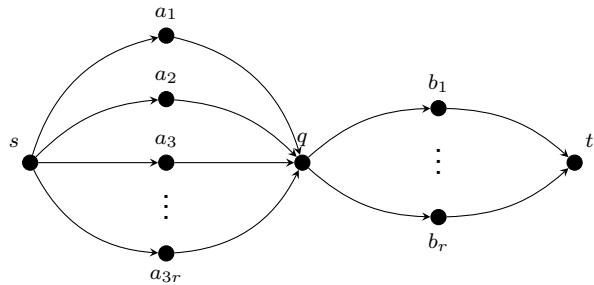
Demonstração. Claramente, esse problema está em \mathcal{NP} , pois dado um conjunto de k fluxos caminhos $x^1 \dots x^k$, é possível verificar em tempo $O(mk)$ se $\sum_{\ell=1}^k x_{ij}^\ell = x_{ij}$, para todo arco ij da rede. Mostraremos uma redução polinomial para esse problema, a partir do problema 3-Partição.

Dada uma instância $S = \{a_1, \dots, a_{3r}\}$, do problema 3-Partição, fazemos $k = 3r$ e construímos uma rede $\mathcal{N} = (D, u)$, onde $D = (V, A)$ é um DAG, com um (s, t) -fluxo x de valor rT da seguinte maneira:

- $V = \{a_1, \dots, a_k, b_1, \dots, b_r, q, s, t\}$;
- $A = \{sa_i, a_iq \mid 1 \leq i \leq k\} \cup \{qb_j, b_jt \mid 1 \leq j \leq r\}$;
- definimos $u_{sa_i} = u_{a_iq} = v(a_i)$, $u_{qb_j} = u_{b_jt} = T$;
- fazemos $x_{ij} = u_{ij}$ para todo $ij \in A$ (isso é possível, pois $\sum_{i=1}^{3r} v(a_i) = rT$);

Essa construção está ilustrada na Figura 17.

Figura 17 – Construção de rede a partir de uma instância do 3-Partição.



Fonte: Elaborada pelo autor.

Se a resposta para o problema 3-Partição for sim, o fluxo x é k -divisível. Seja $\{S_1, \dots, S_r\}$ uma solução para o primeiro problema. Para cada $a_i \in S_j$, com $1 \leq j \leq r$, existe um fluxo caminho sa_iqb_jt com valor $v(a_i)$ na rede.

Considere que o fluxo x é k -divisível. Cada vértice a_i , para $1 \leq i \leq k$, deve estar em exatamente um fluxo caminho, cujo valor é $v(a_i)$. Note que há exatamente três fluxos caminhos com um vértice b_j em comum, para $1 \leq j \leq r$, pela restrição para os valores de a_i ($T/4 < v(a_i) < T/2$). Sejam eles sa_xqb_jt , sa_yqb_jt e sa_zqb_jt , com valores, respectivamente, $v(a_x)$, $v(a_y)$ e $v(a_z)$. Para cada três caminhos desses, construímos um subconjunto $\{a_x, a_y, a_z\}$, com soma igual a T , para o problema 3-Partição. \square

Quando todos os valores de fluxo nos arcos da rede são múltiplos de um determinado inteiro positivo λ , é possível encontrar um limite superior para o número mínimo de caminhos do *FluxoMinDiv*, conforme observado por Hartman et al. (2012), no Lema 3.9.

Lema 3.9 (Hartman et al. (2012)). *Considere uma rede \mathcal{N} sobre um digrafo $D = (V, A)$ e um (s, t) -fluxo x nesta. Para todo arco ij com fluxo positivo, se x_{ij} for múltiplo de λ , para um inteiro positivo λ , então é possível decompor x em exatamente $|x|/\lambda$ caminhos, cada um destes com valor λ .*

Como consequência do Lema 3.9, o problema *FluxoMinDiv* pode ser resolvido em tempo polinomial para fluxos λ -uniforme. Nesse caso, o valor do (s, t) -fluxo é $k \cdot \lambda$, para algum inteiro k . Como todos os arcos com fluxo positivo possuem valor de fluxo λ , esse é o maior valor de um (s, t) -fluxo caminho da decomposição. Consequentemente, o menor número de caminhos desta é igual a k . Além disso, esses caminhos são arco-disjuntos; caso contrário, deveria existir pelo menos um arco com fluxo maior que λ . Caso haja uma circulação, esta pode ser descartada; pois o número de caminhos da decomposição, nesse caso, depende apenas do valor do fluxo.

Um outro caso em que o problema *FluxoMinDiv* pode ser resolvido em tempo polinomial é quando o valor de fluxo em cada arco é um de dois possíveis valores, a e b , e um deles é múltiplo do outro. Definimos a operação $Suporte(\mathcal{N}, x, a)$, que retorna uma rede a partir de \mathcal{N} , contendo apenas os arcos em que o valor do fluxo x é pelo menos a . Adicionalmente, a capacidade de cada arco na rede resultante é o valor do fluxo no arco correspondente na rede original.

De acordo com Hartman et al. (2012), dados uma rede \mathcal{N} e um (s, t) -fluxo x , tal que $x_{ij} \in \{a, b\}$ para todo arco ij da rede e b divide a ($b \mid a$), é possível encontrar a solução ótima para o problema, nessas condições, com o seguinte algoritmo:

- i. Construímos uma rede $\mathcal{N}_a = Suporte(\mathcal{N}, x, a)$ e calculamos um (s, t) -fluxo máximo x_a em \mathcal{N}_a , que é decomposto em p_1 caminhos de valor a ;
- ii. obtemos um fluxo x' em \mathcal{N} , decrementando o valor x_a de x (arco a arco);
- iii. construímos uma rede $\mathcal{N}_b = Suporte(\mathcal{N}, x', b)$ e calculamos um (s, t) -fluxo máximo x_b , que é decomposto em p_2 caminhos de valor b .

Na etapa (i), é possível obter p_1 pelo Lema 3.9. Na etapa (iii), como todos os valores de fluxo x_b são múltiplos de b , é possível obter p_2 também pelo Lema 3.9.

Teorema 3.10 (Hartman et al. (2012)). *Considere uma rede \mathcal{N} e um (s, t) -fluxo x nesta, de modo que há apenas dois valores distintos a e b de fluxo nos arcos e $b \mid a$ (b divide a). A solução produzida pelo algoritmo acima é ótima.*

Demonstração. Seja p^* o número mínimo de caminhos de uma decomposição ótima de x . Seja x' a parte de x que é roteada pela solução ótima através de fluxos caminhos cujos valores são maiores que b . Note que cada caminho desses envia no máximo a unidades de fluxo. Consequentemente, a porção $x - x'$ do fluxo é roteada por fluxos caminhos de valores menores que a . Então, segue que

$$p^* \geq \frac{|x'|}{a} + \frac{|x| - |x'|}{b}.$$

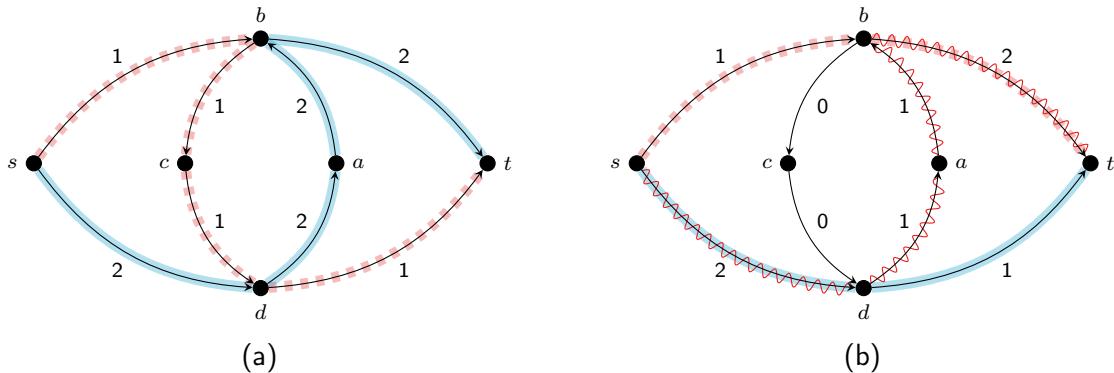
Para minimizar o lado direito dessa inequação, é preciso que $|x'|$ seja o maior possível. No entanto, $|x'|$ é no máximo $|x_a|$, já que x_a foi obtido pelo fluxo máximo da rede suporte, considerando apenas arcos de capacidade a . Pelo algoritmo, o valor desse fluxo máximo é dado por $p_1 \cdot a$. Substituindo na inequação acima, temos:

$$p^* \geq \frac{p_1 \cdot a}{a} + \frac{|x| - p_1 \cdot a}{b} = p_1 + p_2.$$

□

Como foi visto na Seção 2.3.1, uma circulação pode ser removida sem afetar o valor de um (s, t) -fluxo. No entanto, diferentemente do que foi observado na decomposição de fluxos λ -uniformes, a remoção de uma circulação pode afetar o número mínimo de caminhos necessários na decomposição do fluxo, mesmo quando há apenas dois valores distintos de fluxo nos arcos da rede. Um exemplo disso pode ser observado na Figura 18. Em (a), o fluxo pode ser decomposto em 2 fluxos caminhos, um de valor 1 e um de valor 2. Em (b), após a remoção da circulação de valor 1 (ao longo do ciclo $abcta$), são necessários três fluxos caminhos para decompor o fluxo.

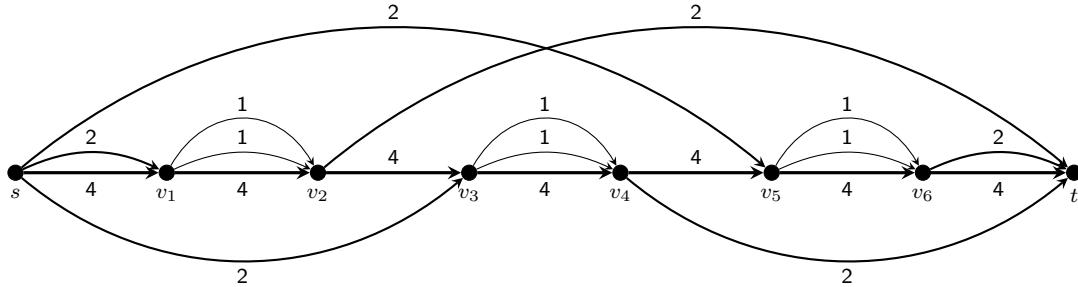
Figura 18 – Influência da circulação na decomposição de um fluxo.



Fonte: Adaptada de Vatinlen et al. (2008, p. 1392)

A estratégia descrita no algoritmo do Teorema 3.10, de pegar os caminhos mais promissores a cada iteração, não funciona quando há mais de 2 valores distintos de fluxo nos arcos da rede. Isso pode ser observado no exemplo da Figura 19. Se for retirado 1 fluxo caminho de valor 4, restarão 6 fluxos caminhos de valor 1, totalizando 7. No entanto, é possível extraír inicialmente 1 fluxo caminho de valor 2 (do fluxo caminho de valor 4); três fluxos caminhos de valor 2, iniciando pelos arcos partindo de s com esse valor; e mais 2 fluxos caminhos de valor 1, totalizando 6. Esse número é mínimo. Para verificar isso, observe que, para evitar os 6 fluxos caminhos de valor 1, é necessário passar 3 fluxos caminhos (dois de valor 1 e um de valor 2) por pelo menos um arco que tem valor de fluxo 4, especialmente pelo arco de v_2 para v_3 ou pelo arco de v_4 para v_5 , e o grau de saída de s é 4. Logo, há pelo menos 4 fluxos caminhos de s a t e um desses deve ser dividido em 3 fluxos caminhos.

Figura 19 – Exemplo ruim para a decomposição de fluxo de forma gulosa.



Fonte: Elaborada pelo autor.

Essa diferença entre as soluções ótima e gulosa pode ser arbitrariamente grande. Para ver isso, basta generalizar a rede da Figura 19 para uma rede com $2n + 2$ vértices $(s, t, v_1, \dots, v_{2n})$. Com a estratégia gulosa, seriam necessários $2n + 1$ (s, t) -fluxos caminhos. Extraindo-se os caminhos de forma semelhante ao sugerido como solução ótima para a rede acima, obtém-se uma decomposição em $n + 3$ (s, t) -fluxos caminhos.

Considerando o caso em que há exatamente três valores distintos de fluxo nos arcos da rede, em Hartman et al. (2012) foi mostrado que o problema *FluxoKDiv* é \mathcal{NP} -Completo mesmo quando os valores de fluxo são potências de 2. Os autores mostraram uma redução polinomial a partir do *3SAT*, um dos problemas \mathcal{NP} -Completos mostrados por Karp (1972), formalizado a seguir:

Problema: 3SAT

Entrada: Uma fórmula φ na 3FNC, com m cláusulas e n variáveis.

Pergunta: φ é satisfatível?

Teorema 3.11 (Hartman et al. (2012)). *FluxoKDiv* é \mathcal{NP} -Completo, mesmo quando a rede possui apenas três valores distintos de fluxo em seus arcos, sendo estes 1, 2 ou 4.

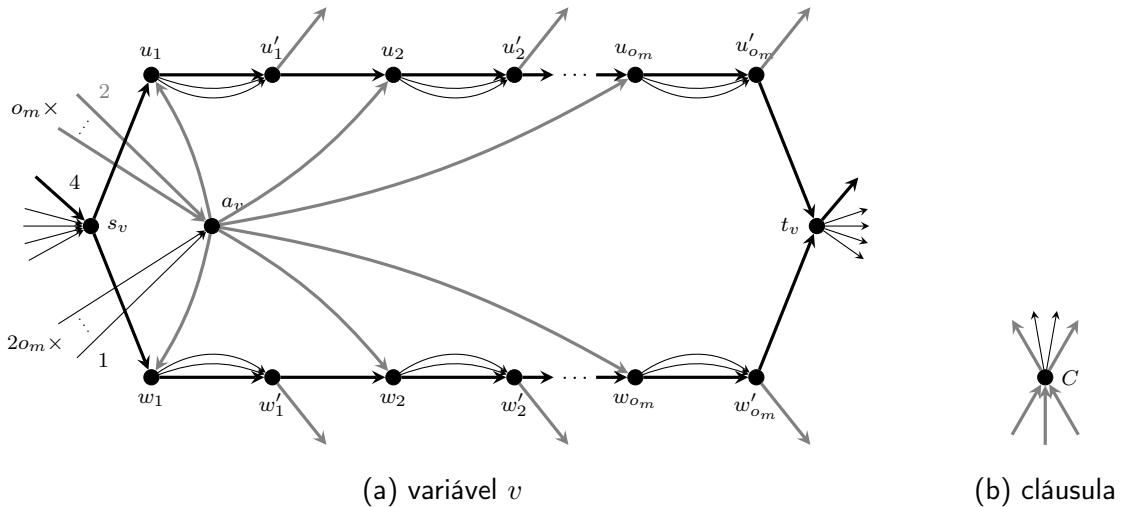
Demonstração. Para uma variável v qualquer, sejam $o(v)$ o número de ocorrências do literal v e $o'(v)$ o número de ocorrências do literal \bar{v} . Seja $o_m(v) = \max\{o(v), o'(v)\}$. Dada uma instância φ do *3SAT* com um conjunto Z de n variáveis e um conjunto C de m cláusulas, é possível construir uma rede \mathcal{N} , com $m+2+\sum_{v \in Z}(3+4o_m(v))$ vértices e $4m+\sum_{v \in Z}(15o_m(v)+12)$ arcos e um fluxo x nesta, de modo que há uma decomposição de x em $\sum_{v \in Z}(5+3o_m(v))$ fluxos caminhos se e somente se φ é satisfatível.

Para cada variável v , é construído um dispositivo mostrado na Figura 20a. Há dois vértices, $s(v)$ e $t(v)$, que são conectados com dois caminhos paralelos com $o_m(v)$ pares de vértices - $u_i(v)$ e $u'_i(v)$, em um; e $w_i(v)$ e $w'_i(v)$ em outro, para $1 \leq i \leq o_m(v)$. Há um arco de $s(v)$ para $u_1(v)$, de $u_i(v)$ para $u'_i(v)$ e deste para $u_{i+1}(v)$, para $1 \leq i < o_m(v)$, e de $u'_{o_m(v)}(v)$ para $t(v)$. Similarmente, Há um arco de $s(v)$ para $w_1(v)$, de $w_i(v)$ para $w'_i(v)$ e deste para $w_{i+1}(v)$, para $1 \leq i < o_m(v)$, e de $w'_{o_m(v)}(v)$ para $t(v)$.

A fonte s é conectada a todos os vértices $s(v_j)$ e todos os vértices $t(v_j)$ são conectados ao sumidouro t , para $1 \leq j \leq n$. O fluxo nos arcos especificados até aqui é 4 e estes são os únicos arcos com esse valor de fluxo em x . Adicionalmente, há quatro arcos com valor de fluxo 1, de s para $s(v_j)$ e de $t(v_j)$ para t , com $1 \leq j \leq n$. Há também um par de arcos com valor de fluxo 1 de $u_i(v)$ para $u'_i(v)$ e de $w_i(v)$ para $w'_i(v)$, para $1 \leq i \leq o_m(v)$. Para cada variável v , há um vértice $a(v)$. O vértice s é conectado ao $a(v)$ com $o_m(v)$ arcos com valor de fluxo 2 e $2o_m(v)$ arcos com valor de fluxo 1. Além disso, $a(v)$ é conectado a $u_i(v)$ e a $w_i(v)$, para $1 \leq i \leq o_m(v)$, com um arco de fluxo 2.

Para cada cláusula c , há um vértice c conectado a t por quatro arcos, dois com fluxo 1 e dois com fluxo 2; e três arcos com fluxo 2, cada um vindo da estrutura de uma variável correspondente a um literal de c , como ilustrado na Figura 20b.

Figura 20 – Redução do 3SAT para o FluxoKDiv com 3 valores de fluxo



Fonte: Adaptada de Hartman et al. (2012)

Por fim, é preciso conectar os vértices dos dispositivos de variáveis aos vértices de cláusulas. Cada vértice $u'_i(v)$ é conectado a um vértice de cláusula correspondente a uma cláusula em φ que contém v , para $1 \leq i \leq o(v)$. Essas conexões são feitas de tal modo que cada um desses vértices de cláusula esteja conectado a um único vértice $u'_i(v)$. Analogamente, cada vértice $w'_i(v)$ é conectado a um vértice de cláusula correspondente a uma cláusula em φ que contém \bar{v} . Se $o(v) < o_m(v)$, cada $u'_i(v)$, para $o(v) + 1 \leq i \leq o_m(v)$, é conectado a t . Se $o'(v) < o_m(v)$, cada arco $w'_i(v)$, para $o'(v) + 1 \leq i \leq o_m(v)$, é conectado a t . Todos os arcos descritos neste parágrafo têm valor de fluxo 2.

Agora será mostrado que φ é satisfatível se e somente se é possível decompor x em $\sum_{v \in Z}(5 + 3o_m(v))$ caminhos. Esse número $5 + 3o_m(v)$ corresponde à quantidade de arcos de s para cada estrutura de variável. Assuma que φ é satisfatível e considere uma atribuição de valores às suas variáveis que a tornam verdadeira. Cada caminho da decomposição satura um arco de s . Para uma variável v cujo valor é 1 (verdadeiro), há um par de caminhos de

valor 1 para cada vértice de cláusula que contém o literal v e um caminho de valor 2 para cada vértice de cláusula contendo o literal \bar{v} . Para uma variável v cujo valor é 0 (*falso*), há um par de caminhos de valor 1 para cada vértice de cláusula que contém o literal \bar{v} e um caminho de valor 2 para cada vértice de cláusula contendo o literal v . Cada cláusula tem pelo menos um literal com valor 1 e, portanto, pelo menos um par de caminhos de valor 1 entrando no vértice da cláusula. É possível rotear esses caminhos através dos vértices das cláusulas sem dividi-los.

Para cada variável v , será mostrado inicialmente como devem ser tomados os 5 (s, t) -caminhos que passam pelo vértice $s(v)$, e, em seguida, os $3o_m(v)$ (s, t) -caminhos que passam pelo vértice $a(v)$.

Se $v = 1$, tomar um único fluxo caminho de valor 4 a partir de s , passando por $s(v)$, por $u_i(v)$ e $u'_i(v)$, para $1 \leq i \leq o_m(v)$, até $t(v)$; e 4 fluxos caminhos de valor 1 a partir de s , passando por $s(v)$, por $w_i(v)$ e $w'_i(v)$, para $1 \leq i \leq o_m(v)$, até $t(v)$. Dois desses caminhos usam arcos com valor de fluxo 1, de $w_i(v)$ para $w'_i(v)$, e os outros dois usam os arcos de fluxo 4. Se $v = 0$, tomar um único fluxo caminho de valor 4 a partir de s , passando por $s(v)$, por $w_i(v)$ e $w'_i(v)$, para $1 \leq i \leq o_m(v)$, até $t(v)$; e 4 fluxos caminhos de valor 1 a partir de s , passando por $s(v)$, por $u_i(v)$ e $u'_i(v)$, para $1 \leq i \leq o_m(v)$, até $t(v)$. Dois desses caminhos usam arcos com valor de fluxo 1, de $u_i(v)$ para $u'_i(v)$, e os outros dois usam os arcos de valor de fluxo 4.

Se $v = 1$, tomam-se $2o(v)$ fluxos caminhos de valor 1, de s , passando por $a(v)$, depois por $u_i(v)$ e $u'_i(v)$ e deste para um vértice de cláusula contendo o literal v , para $1 \leq i \leq o(v)$. Se $o(v) < o_m(v)$, deve-se tomar $2(o_m(v) - o(v))$ fluxos caminhos de valor 1 passando por $a(v)$, por $u_i(v)$ e $u'_i(v)$ e deste para t , com $o(v) < i \leq o_m(v)$. Pegam-se também $o'(v)$ fluxos caminhos de valor 2 a partir de s , passando por $a(v)$, por $w_i(v)$ e $w'_i(v)$ e deste para um vértice de cláusula contendo o literal \bar{v} , para $1 \leq i \leq o(v)$. Se $o'(v) < o_m(v)$, deve-se tomar $o_m(v) - o'(v)$ fluxos caminhos de valor 2 passando por $a(v)$, $w_i(v)$ e $w'_i(v)$ e deste para t , com $o'(v) < i \leq o_m(v)$. Cada um desses fluxos caminhos usa um arco de fluxo 4 de $w_i(v)$ para $w'_i(v)$; e, se o caminho passa por um vértice de cláusula (que tem o literal \bar{v}), ele usa um arco com fluxo 2 saindo desse vértice.

Simetricamente, se $v = 0$, tomam-se $o(v)$ fluxos caminhos de valor 2 a partir de s , passando por $a(v)$, por $u_i(v)$ e $u'_i(v)$ e deste para um vértice de cláusula contendo o literal v , para $1 \leq i \leq o(v)$. Se $o(v) < o_m(v)$, deve-se tomar $o_m(v) - o(v)$ fluxos caminhos de valor 2 a partir de s , passando por $a(v)$, $u_i(v)$ e $u'_i(v)$ e deste para t , com $o(v) < i \leq o_m(v)$. Cada um desses caminhos usa um arco de fluxo 4 de $u_i(v)$ para $u'_i(v)$; e, se o caminho passa por um vértice de cláusula (que tem o literal v), ele usa um arco de valor de fluxo 2 saindo desse vértice. Deve-se pegar também $2o'(v)$ fluxos caminhos de valor 1 a partir de s , passando por $a(v)$, depois por $w_i(v)$ e $w'_i(v)$ e em seguida para um vértice de cláusula contendo \bar{v} , para $1 \leq i \leq o'(v)$. Se $o'(v) < o_m(v)$, deve-se pegar $2(o_m(v) - o'(v))$ fluxos caminhos de valor 1, a partir de s , passando por $a(v)$ e por $w_i(v)$ e $w'_i(v)$ e destes para t , com $o'(v) < i \leq o_m(v)$.

Para mostrar que esta decomposição é bem definida, pode-se argumentar que cada fluxo caminho de valor 2 chegando a um vértice de cláusula pode usar um arco de valor de fluxo 2 saindo deste. Uma vez que a atribuição satisfaz φ , toda cláusula tem pelo menos um literal cujo valor é 1, que corresponde a dois fluxos caminhos de valor 1 chegando ao vértice desta. Logo, cada vértice de cláusula tem no máximo dois fluxos caminhos de valor 2 chegando a ele e saindo por um dos arcos com valor de fluxo 2.

Suponha que haja uma decomposição de x em $\sum_{v \in Z} (5 + 3o_m(v))$ caminhos. Note que há exatamente $5 + 3o_m(v)$ arcos saindo de s para a estrutura de cada variável v (um com fluxo 4, $o_m(v)$ com fluxo 2 e $2o_m(v) + 4$ com fluxo 1). Então, nenhum fluxo caminho de s a t sofre divisão. Isso significa que há dois fluxos caminhos de valor 1 saindo das estruturas de cada variável v , passando pelas estruturas de cláusulas que contém um de seus literais (ou v ou \bar{v}). Para cada variável v de φ , observa-se a estrutura correspondente a ela na rede. Se, nesta, o caminho de valor 4 passa pelos vértices u_i , então há dois caminhos de valor 1 passando pelos vértices $u'_i(v)$, para $1 \leq i \leq o(v)$. Nesse caso, atribui-se o valor 1 à variável v . Caso contrário, atribui-se o valor 0 a v . \square

Em relação ao problema de encontrar o menor número de caminhos necessários para decompor um dado fluxo, foi visto que a solução ótima pode ser obtida em tempo polinomial quando todos os arcos possuem o mesmo valor de fluxo, ou quando há apenas dois valores possíveis para o fluxo em cada arco e o menor divide o maior deles. Resta analisar o caso em que há dois valores de fluxo nos arcos da rede suporte, mas o menor não divide o maior deles. Isso será tratado na Seção 4.3.1, ao abordarmos a decomposição de fluxos em redes arco-coloridas monocromáticas.

4 FLUXOS EM REDES ARCO-COLORIDAS

De acordo com Granata et al. (2013), digrafos arco-coloridos são usados para modelar situações nas quais é crucial representar aspectos qualitativos (em vez de quantitativos) em diferentes regiões do próprio digrafo. Cada cor pode representar uma propriedade a ser modelada. Problemas desse tipo têm aplicações em diversas áreas como redes de comunicação, redes de transporte multimodal, entre outras.

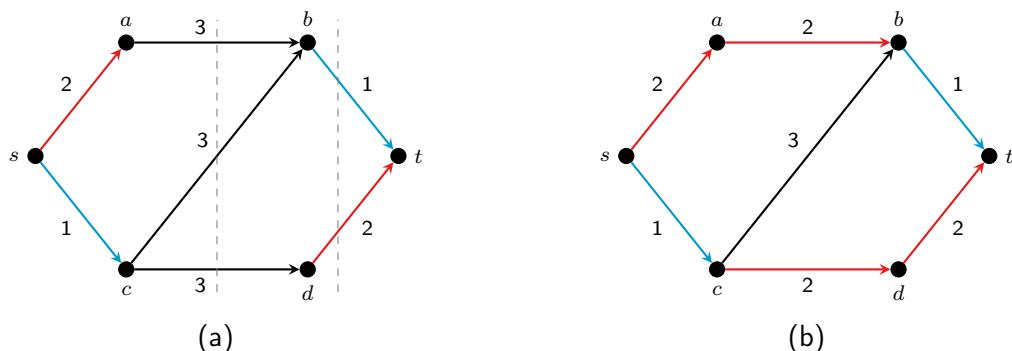
Neste capítulo, apresentaremos o conceito de fluxos em redes arco-coloridas com os problemas relacionados e suas variações e um estudo da complexidade destas. Nossas contribuições são dadas na Seção 4.3.

4.1 O Problema e suas variações

Denotamos por $D^c = (V, A, c)$ um (multi)digrafo $D = (V, A)$ com uma coloração de arcos $c : A \rightarrow \{1, 2, \dots, p\}$ não necessariamente própria, ou seja, é possível usar uma mesma cor para arcos adjacentes. Uma rede \mathcal{N} definida sobre D^c é dita arco-colorida. Denotamos por $\text{cores}(D^c)$ o conjunto imagem de c , e por $n_c(D^c)$ a sua cardinalidade. Para uma cor i , o $\text{span}(i, D^c)$ indica o número de componentes conexas do subdigrafo induzido de D^c pelos arcos de cor i . A Figura 21 traz duas redes arco-coloridas. Em ambas, o span das cores 1 e 2 é igual a 2 e o da cor 3 é igual a 1.

Segundo Coudert et al. (2007), em algumas situações é necessário levar em conta as correlações entre arcos, especialmente no conceito de sobrevivência de um Grupo de Recursos de Risco Compartilhado (do inglês, *Shared Risk Resource Group - SRRG*). Um SRRG é naturalmente modelado associando a cada risco uma cor. Em um sistema desse tipo, um caminho “mais seguro” é o que tem o menor número de cores distintas. Dois caminhos são disjuntos em cores se não compartilham cores. SRRGs são relevantes em inúmeros contextos práticos. Em particular, eles foram introduzidos para modelar grupos de recursos em uma rede de telecomunicações que podem falhar simultaneamente.

Figura 21 – Exemplos de (s, t) -corte em redes arco-coloridas.



Fonte: Elaborada pelo autor.

De acordo com Coudert et al. (2007), problemas em grafos coloridos diferem daqueles em grafos tradicionais, não apenas em termos de complexidade, mas também nas relações que mantém entre eles. Um caso particular é o número de (s, t) -caminhos disjuntos em cores e o *MC-st-Cut* (de *Minimum Colour (s, t)-Cut*) que, diferentemente das versões tradicionais (ver Teorema de Menger), não estão intimamente ligados.

Segundo Coudert et al. (2007), o objetivo do *MC-st-Cut* é encontrar um conjunto de arcos com o menor número de cores que separa os vértices s e t . Os autores mostraram que esse problema é \mathcal{NP} -Difícil, ao passo que a versão tradicional pode ser resolvida em tempo polinomial pelos algoritmos de fluxo máximo. Veja os exemplos de rede da Figura 21. Em (a), as linhas pontilhadas representam (s, t) -cortes - um com uma cor e capacidade 3λ e outro com duas cores e capacidade 2λ . Em (b), qualquer (s, t) -corte tem pelo menos duas cores. No entanto, não há 2 (s, t) -caminhos disjuntos em cores.

Neste trabalho, são abordados os seguintes problemas:

1. Dada uma rede $\mathcal{N} = (D^c, u)$, encontrar um (s, t) -fluxo máximo que usa o menor número de cores.
2. Dados uma rede \mathcal{N} sobre um digrafo colorido D^c e um (s, t) -fluxo x nesta, encontrar uma decomposição de x em (s, t) -fluxos caminhos que minimize a soma do número de cores em cada um destes.

Esses dois problemas serão melhor detalhados nas Seções 4.2 e 4.3 a seguir.

4.2 Encontrando o fluxo máximo com o menor número de cores

Como foi dito no início deste capítulo, as cores são utilizadas para representar características de cada região por onde o fluxo irá passar. Por exemplo, em uma rede de distribuição de água, as cores podem ser utilizadas para representar substâncias com as quais a água entrará em contato em determinadas regiões. Ao chegar ao destino, é preciso tratar a água para cada substância com a qual houve contato. Assim, minimizar o número de cores do fluxo implica minimizar o custo com o tratamento final da água.

O problema consiste em, dada uma rede $\mathcal{N} = (D^c, u)$, encontrar um (s, t) -fluxo máximo nesta com o menor número de cores distintas nos arcos com fluxo positivo.

Inicialmente, abordaremos o problema de encontrar uma árvore geradora em um grafo com cores nas arestas, de modo que o número de cores distintas da árvore seja mínimo, chamado de *MinColSpanningTree*. Esse problema é \mathcal{NP} -Difícil, pois, segundo Broersma e Li (1997), a sua versão de decisão, aqui referenciada como *KColSpanningTree*, é um problema \mathcal{NP} -Completo (ver Teorema 4.1).

Problema: <i>KColSpanningTree</i>
Entrada: Um grafo $G = (V, E)$, uma coloração $c : E \rightarrow \mathbb{N}$ e um inteiro positivo k .
Pergunta: Existe uma árvore geradora T de G , com no máximo k cores?

Broersma e Li (1997) propuseram uma redução para o *KColSpanningTree* a partir da versão de decisão do *Problema Conjunto Dominante*, formalizado a seguir, que segundo Haynes, Hedetniemi e Slater (1998) é \mathcal{NP} -Completo.

Problema: *Conjunto Dominante*

Entrada: Um grafo $G = (V, E)$ e um inteiro positivo k .

Pergunta: Existe um conjunto $S \subseteq V$, com $|S| \leq k$, de modo que todo vértice em V esteja em S ou seja adjacente a algum vértice em S ?

Teorema 4.1 (Broersma e Li (1997)). *KColSpanningTree* é \mathcal{NP} -Completo.

Demonstração. É fácil ver que o problema está em \mathcal{NP} . Basta verificar se T é conexo, se tem $n - 1$ arestas ($n = |V|$), e se $n_c(T) \leq k$.

Será apresentada uma redução para esse problema, a partir do *Problema do Conjunto Dominante*. Dados um grafo $G = (V, E)$ e um inteiro $k \leq n$, vamos construir, em tempo polinomial, um grafo $G' = (V', E')$ e uma coloração $c : E' \rightarrow \{1, \dots, n+1\}$, de modo que G' tem uma árvore geradora T com $n_c(T) \leq k + 1$ se e somente se G tem um conjunto dominante S de tamanho no máximo k .

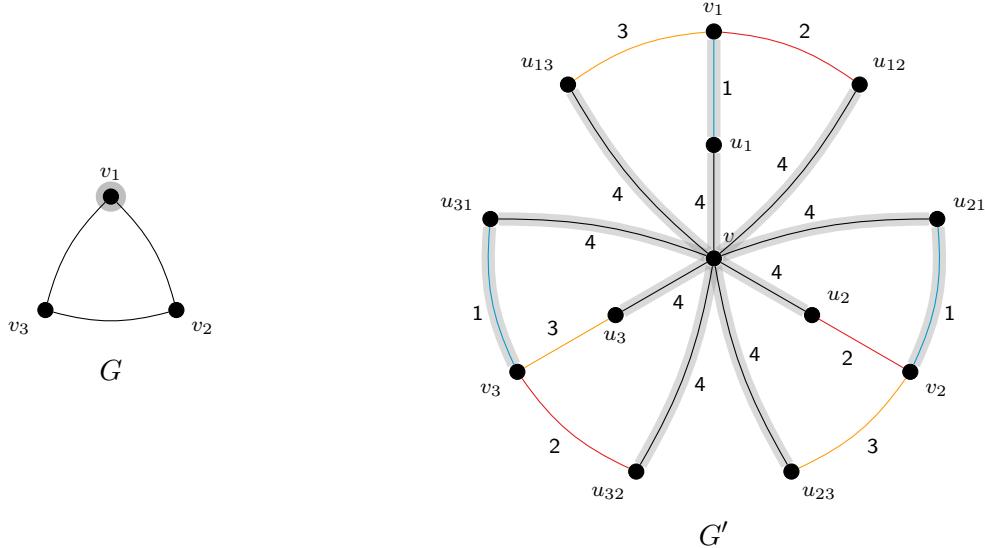
Seja $V = \{v_1, \dots, v_n\}$. Agora, construímos G' como segue. Comece com o conjunto de vértices V e, para cada $v_i \in V$, adicione um novo vértice u_i e uma aresta $v_i u_i$ com cor i . Além disso, para cada aresta $v_i v_j \in E$, adicione dois vértices u_{ij} e u_{ji} , uma aresta $v_i u_{ij}$ com cor j e uma aresta $v_j u_{ji}$ com cor i . Por fim, adicione um vértice v , com arestas de cor $n + 1$ de v para cada vértice u . Essa construção, que está ilustrada na Figura 22, é feita em tempo polinomial.

Primeiro, assuma que G tem um conjunto dominante S , com $|S| \leq k$. Para cada vértice $v_i \in V$, escolha um vértice $v_j \in S$ tal que ou $v_i = v_j$ ou $v_i v_j \in E$. Em G' , escolha a aresta correspondente $v_i u_i$ com cor $i = j$, se $v_i = v_j$; ou a aresta $v_i u_{ij}$ com cor j , se $v_i v_j \in E$. Daí, em G' , foram escolhidas precisamente n arestas com no máximo $|S| \leq k$ cores distintas. Adicionando todas as arestas com a cor $n + 1$, obtemos um conjunto de arestas de uma árvore geradora de G' com no máximo $k + 1$ cores distintas.

Agora, assuma que G' tem uma árvore geradora T com $n_c(T) \leq k + 1$. Claramente, T contém uma aresta com a cor $n + 1$, pois toda aresta incidente em v tem essa cor. É possível obter uma árvore geradora T' a partir de T em que cada vértice v_i tem grau 1. Note que T contém precisamente um caminho de tamanho 2 de v a v_i . Escolha uma aresta incidente a v_i em T e substitua quaisquer outras arestas incidentes com v_i em T por arestas adjacentes a estas e que incidam em v . Então, a nova árvore geradora T' de G' tem $n_c(T') \leq k + 1$. É fácil ver que as cores das arestas de T' não incidentes em v correspondem aos índices dos vértices de um conjunto dominante S de G , com $|S| = n_c(T') - 1 \leq k$. Se a aresta em T' tem extremidade v_i e cor j , com $i \neq j$, então o vértice v_i é dominado pelo vértice v_j em G . Isso completa a prova. \square

A Figura 22 traz um exemplo da redução apresentada no Teorema 4.1. Em G' , o rótulo de cada arco indica a sua cor. Observe que $S = \{v_1\}$ é um conjunto dominante em G e que G' admite uma árvore geradora T com $|S| + 1 = 2$ cores (1 e 4).

Figura 22 – Exemplo de redução do *Conjunto Dominante* para *KColSpanningTree*.



Fonte: Elaborada pelo autor.

Observe que se a função c for injetiva, ou seja, se não existem duas arestas com a mesma cor, então qualquer árvore geradora terá exatamente $n - 1$ cores distintas.

Considere agora o problema *FluxoMaxKCores*, que consiste em determinar se uma rede arco-colorida admite um (s, t) -fluxo máximo com um número limitado de cores.

Problema: FluxoMaxKCores

Entrada: Uma rede $\mathcal{N} = (D^c, u)$ e um inteiro positivo k .

Pergunta: \mathcal{N} admite um (s, t) -fluxo máximo usando arcos de no máximo k cores?

Em Granata et al. (2013), os autores mostraram que o problema acima é \mathcal{NP} -Completo. Para isso eles propuseram uma redução polinomial para ele a partir do *Problema KColSpanningTree*,

Teorema 4.2 (Granata et al. (2013)). *FluxoMaxKCores* é \mathcal{NP} -Completo.

Demonstração. O problema está em \mathcal{NP} , pois é possível verificar em tempo polinomial se um fluxo x é viável e máximo, e se o seu digrafo suporte tem no máximo k cores distintas.

Dado um grafo $G = (V, E)$, com $V = \{v_1, \dots, v_n\}$, e uma coloração $c : E \rightarrow \{1, \dots, p\}$, será construído um digrafo arco-colorido $D^c = (V', A', c')$, com $c' : A' \rightarrow \{1, \dots, p+1\}$; e será definida uma rede \mathcal{N} sobre este, de modo que existe uma árvore geradora de G com no máximo k cores se e somente se \mathcal{N} admite um (s, t) -fluxo máximo e o digrafo suporte de x tem no máximo $k' = k + 1$ cores distintas.

Inicialmente, seja D^c o digrafo associado de G , onde cada aresta é substituída por dois arcos anti-paralelos. Procedemos com a construção de D^c da seguinte maneira:

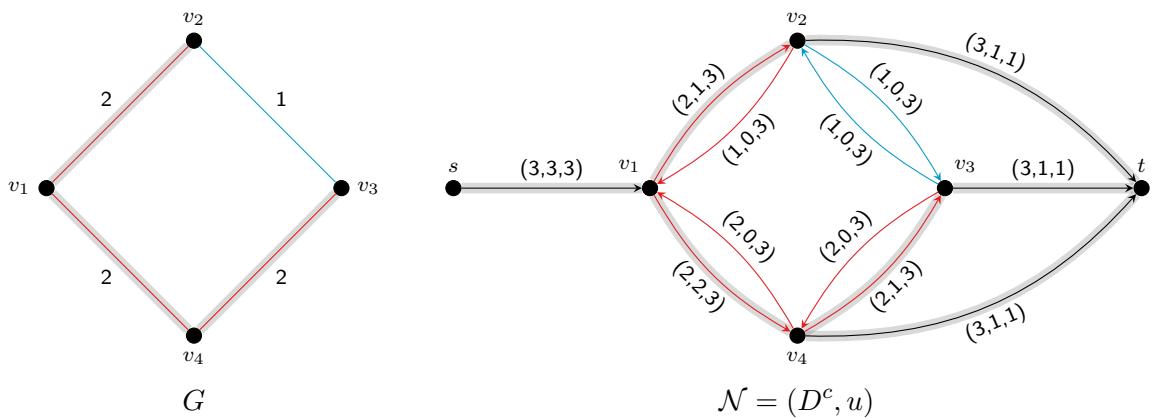
- adicionar dois novos vértices a V' , a fonte s e o sumidouro t ($V' = V \cup \{s, t\}$);
- adicionar a A' um arco sv_1 , com capacidade $n - 1$;
- adicionar a A' um arco $v_i t$, com capacidade 1, para $1 < i \leq n$;
- a capacidade dos demais arcos de A' é $n - 1$;
- defina a coloração c' da seguinte forma: $c'(ij) = c'(ji) = c(ij)$, para todo $ij \in E$, e $c'(ij) = p + 1$ para os demais arcos em A' ;

O digrafo D^c possui $n + 2$ vértices, $2m + n$ arcos e $p + 1$ cores. Isso é feito em tempo polinomial. Essa construção está ilustrada na Figura 23. Note que o valor de um fluxo máximo na rede $\mathcal{N} = (D^c, u)$ é $n - 1$. Além disso, todos os arcos com cor $p + 1$ devem ter fluxo positivo em qualquer solução de um fluxo máximo, pois todo arco com cauda s ou cabeça t tem essa cor.

Considere agora uma árvore geradora T de G com no máximo k cores. Um (s, t) -fluxo máximo em \mathcal{N} pode ser definido enviando $n - 1$ unidades de fluxo de s a v_1 , uma unidade de fluxo de v_i a t , e adicionando uma unidade de fluxo aos arcos de cada caminho de v_1 a v_i , com $1 < i \leq n$.

Considere agora um (s, t) -fluxo máximo x em \mathcal{N} com $k' = k + 1$ cores. Pela construção acima, $|x| = n - 1$ e há $n - 1$ (s, t) -fluxos caminhos de valor 1, cada um destes finalizando em um arco $v_i t$, para $1 < i \leq n$. Seja D_x^c o digrafo suporte de x e $G' = G(D_x^c)$ o seu grafo subjacente, considerando que a cor de cada aresta ij é a mesma dos arcos ij e ji em D^c . Apenas os arcos com extremidade em s ou t possuem a cor $p + 1$. Assim, $G'' = G'[\{v_1, \dots, v_n\}]$ é conexo e é um subgrafo gerador de G com no máximo k cores e qualquer árvore geradora T de G'' também é uma árvore geradora de G . \square

Figura 23 – Exemplo de Redução do *MCSpaningTree* para *MCMaxFlow*.



Fonte: Elaborada pelo autor.

A Figura 23 traz um exemplo da redução proposta no Teorema 4.2. O grafo G possui 2 cores (representadas pelos rótulos nas arestas) e o digrafo D^c da rede \mathcal{N} possui 3 cores. Em \mathcal{N} , a cada arco ij está associado um rótulo (c_{ij}, x_{ij}, u_{ij}) , indicando, respectivamente, a sua cor, o valor do fluxo nele e a sua capacidade. Os arcos destacados representam, em G , uma árvore geradora; e em \mathcal{N} , o digrafo suporte de x .

4.3 Minimizando o custo da decomposição de um fluxo colorido

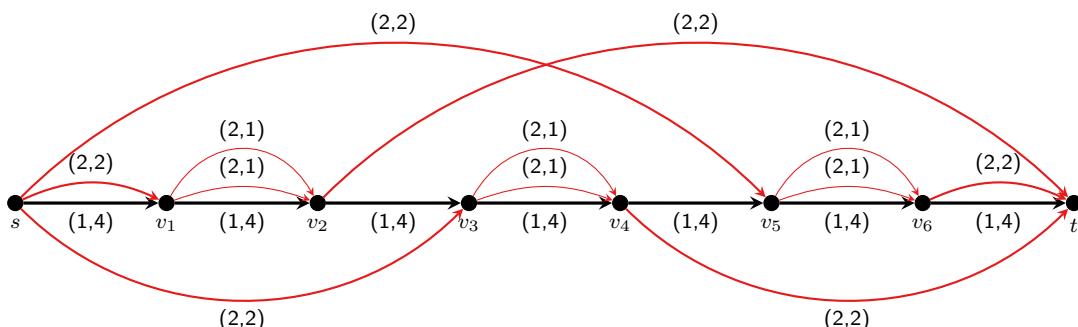
Como foi mencionado na Seção 3.3, um fluxo em uma rede pode ser visto como um conjunto de fluxos caminhos da origem ao destino. No transporte de mercadorias, cada fluxo caminho pode ser visto como um contêiner que será usado no transporte de parte da demanda ao longo de um caminho da origem ao destino. As cores podem ser usadas para modelar fatores externos aos quais a mercadoria estará sujeita, ou tipos de transporte utilizados em cada trecho. Minimizar o número de cores de um caminho implica minimizar o tratamento inicial que deve ser dado à carga que seguirá por ele.

O problema que estamos propondo consiste em: dados uma rede $\mathcal{N} = (D^c, u)$ e um (s, t) -fluxo viável x sobre ela; desejamos encontrar uma decomposição de x em ℓ (s, t) -fluxos caminhos, x^1, \dots, x^ℓ , onde cada x^i é enviado ao longo de um caminho direcionado P_i , com o objetivo de minimizar o custo da solução, dado por:

$$n_c(P_1, \dots, P_\ell) = \sum_{i=1}^{\ell} n_c(P_i) \quad (4.1)$$

Esse problema está referenciado neste trabalho por *DFColCustoMin* (*Decomposição de Fluxo Colorido com Custo Mínimo*). Apesar de se assemelhar ao problema de decomposição de um fluxo em um número mínimo de caminhos, a solução ótima para este nem sempre minimiza o custo do primeiro. Isso pode ser visto na Figura 24, em que é dado um (s, t) -fluxo de valor 10 em uma rede sobre um digrafo arco-colorido. Nesta, a cada arco está associado um rótulo (c, x) indicando a cor e o valor de fluxo dele.

Figura 24 – Exemplo de fluxo em rede sobre um digrafo arco-colorido.



Fonte: Elaborada pelo autor.

Usando uma decomposição que minimiza o número de (s, t) -fluxos caminhos, é possível obter 6 destes. Observe que essa é uma versão colorida da rede da Figura 19 (página 40). Foi visto que qualquer decomposição que minimiza a quantidade de fluxos caminhos deve usar um arco com valor de fluxo 4 em três fluxos caminhos, um de valor 2 e dois de valor 1. Os dois fluxos caminhos de valor 1 são, portanto, bicromáticos. Restam quatro fluxos caminhos de valor 2, dos quais três são necessariamente bicromáticos e são extraídos da seguinte forma: cada fluxo caminho i usa o arco de cor 1 de v_{2i-1} a v_{2i} e o arco de cor 2 de v_{2i} a t , com $1 \leq i \leq 3$. Dependendo de como for extraído o primeiro fluxo caminho na etapa anterior (se for usado o arco de cor 1 ou o de cor 2, de s a v_1) o último fluxo caminho poderá ter uma ou duas cores. Logo, o custo de uma decomposição que minimiza o número de fluxos caminhos pode ser 11 ou 12. No entanto, é possível decompor o (s, t) -fluxo da Figura 24 em 7 (s, t) -fluxos caminhos monocromáticos, e portanto a um custo 7, da seguinte maneira: 1 fluxo caminho de cor 1 e valor 4; 3 pares de fluxos caminhos de cor 2 e valor 1, cada par saindo de s para v_{2i-1} , de v_{2i-1} para v_{2i} , e de v_{2i} a t , com $1 \leq i \leq 3$.

O problema de encontrar um (s, t) -caminho com o menor número de cores em um digrafo arco-colorido, chamado de *MinColourPath*, é \mathcal{NP} -Difícil. Consideremos a sua versão de decisão, o problema *KColourPath*, definido como:

Problema: *KColourPath*

Entrada: Um digrafo arco-colorido D^c , dois vértices s e t e um inteiro positivo k .

Pergunta: Existe um (s, t) -caminho P em D^c tal que $n_c(P) \leq k$?

Em Yuan, Varma e Jue (2005), foi mostrada uma redução para o *KColourPath*, a partir do *Problema de Cobertura de Conjuntos* (*SetCover*), um dos 21 problemas \mathcal{NP} -Completos de Karp (1972), definido da seguinte maneira:

Problema: *SetCover*

Entrada: Dois conjuntos, A e $C \subseteq \mathcal{P}(A)$, tal que $\bigcup C = A$, e um inteiro positivo k .

Pergunta: Existe $C' \subseteq C$, com $|C'| \leq k$, de modo que $\bigcup C' = A$?

Teorema 4.3. (Yuan, Varma e Jue (2005)) *KColourPath* é \mathcal{NP} -Completo.

Demonstração. Este problema está em \mathcal{NP} , pois dada uma sequência de vértices e arcos P , é possível verificar em tempo polinomial se P é um (s, t) -caminho e se $n_c(P) \leq k$. Será feita uma redução do *SetCover* para esse problema. Dados dois conjuntos $A = \{a_1, \dots, a_\ell\}$ e $C \subseteq \mathcal{P}(A) = \{A_1, \dots, A_m\}$, com $\bigcup C = A$ e um inteiro $k \leq m$, construimos um digrafo $D^c = (V', A', c)$, com $c : A \rightarrow \{1, \dots, m\}$, da seguinte maneira:

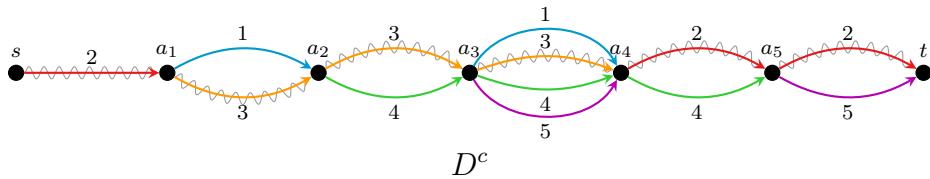
- $V' = \{a_0, a_1, \dots, a_\ell\}$, onde $a_0 = s$ e $a_\ell = t$;
- para cada elemento a_i , e para cada conjunto A_j tal que $a_i \in A_j$, com $1 \leq i \leq \ell$ e $1 \leq j \leq m$, criar um arco de a_{i-1} para a_i com a cor j .

Note que se há um (s, t) -caminho com até k cores, então as cores neste podem ser mapeadas diretamente para um conjunto $C' \subseteq C$, com $|C'| \leq k$, que cobre todos os elementos de A , ou seja, $\bigcup C = A$. Por outro lado, se há um conjuntos $C' \subseteq C$, com $|C'| \leq k$, tal que $\bigcup C' = A$, então é possível encontrar um caminho iniciando em s , passando por cada vértice a_i , selecionando um arco com a cor j , representando um conjunto $C_j \in C'$ tal que $a_i \in C_j$, para $1 \leq i \leq \ell$, que terá no máximo k cores. \square

Figura 25 – Exemplo de redução do *SetCover* para o *KColourPath*.

$$A = \{a_1, a_2, a_3, a_4, a_5, a_6\}$$

$$C = \{\underbrace{\{a_2, a_4\}}_{A_1}, \underbrace{\{a_1, a_5, a_6\}}_{A_2}, \underbrace{\{a_2, a_3, a_4\}}_{A_3}, \underbrace{\{a_3, a_4, a_5\}}_{A_4}, \underbrace{\{a_4, a_6\}}_{A_5}\} \text{ e } k = 2$$



Fonte: Elaborada pelo autor.

A Figura 25 traz um exemplo da redução apresentada no Teorema 4.3. Considere os conjuntos A e C e um inteiro $k = 2$. No digrafo D^c , o rótulo de cada arco indica a sua cor. O (s, t) -caminho destacado possui apenas as cores 2 e 3, o que corresponde a dizer que $C' = \{A_2, A_3\}$ é uma cobertura de conjuntos de A .

Motivados pelo resultado acima, decidimos investigar a complexidade do problema *DFColCustoMin*. Em Carvalho et al. (2023a), mostramos que a versão de decisão do problema, o *DFColCustoK* (enunciado a seguir), é \mathcal{NP} -Completo.

Problema: *DFColCustoK*

Entrada: Uma rede $\mathcal{N} = (D^c, u)$, com $D^c = (V, A, c)$, um (s, t) -fluxo x e um $k \in \mathbb{Z}_+^*$.

Pergunta: Existe uma decomposição de x em (s, t) -fluxos caminhos com custo no máximo k ?

Teorema 4.4. *DFColCustoK* é \mathcal{NP} -Completo.

Demonstração. Esse problema está em \mathcal{NP} , pois dado um (s, t) -fluxo x e uma sequência de (s, t) -fluxos caminhos x^1, \dots, x^ℓ , verifica-se em tempo polinomial se $\sum_{i=1}^\ell x_a^i = x_a$, para todo arco a , onde cada x^i é enviado ao por um caminho P_i e se $\sum_{i=1}^\ell n_c(P_i) \leq k$.

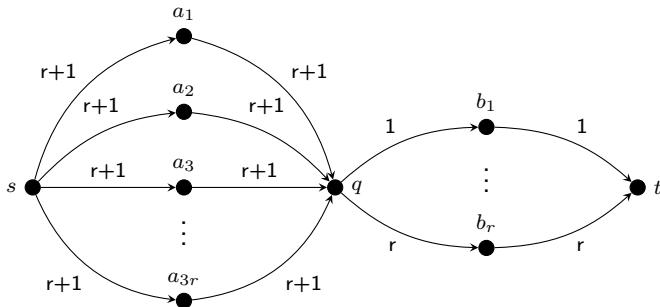
Seja $S = \{a_1, \dots, a_{3r}\}$, com $\sum_{i=1}^{3r} v(a_i) = rT$ e $T/4 < v(a_i) < T/2$, uma instância do problema 3-Partição¹, definimos $k = 6r$ e construímos uma rede \mathcal{N} com um (s, t) -fluxo x nesta, de valor rT , da seguinte maneira:

¹ Problema definido na página 36

- $V = \{a_1, \dots, a_{3r}, b_1, \dots, b_r, q, s, t\}$;
- $A = \{sa_i, a_iq \mid 1 \leq i \leq 3r\} \cup \{qb_j, b_jt \mid 1 \leq j \leq r\}$;
- definimos $u_{sa_i} = u_{a_iq} = v(a_i)$, e $u_{qb_j} = u_{b_jt} = T$;
- fazemos $x_{ij} = u_{ij}$ para todo $ij \in A$ (isso é possível, pois $\sum_{i=1}^{3r} v(a_i) = rT$);
- definimos a coloração $c_{sa_i} = c_{a_iq} = r+1$ e $c_{qb_j} = c_{b_jt} = j$.

A rede obtida possui $4r+3$ vértices e $8r$ arcos. Portanto, a construção é feita em tempo polinomial no tamanho da entrada. Isso está ilustrado na Figura 26. Nesta, o rótulo de cada arco indica a sua cor.

Figura 26 – Construção de rede a partir de uma instância do 3-Partição.



Fonte: Elaborada pelo autor.

Se a resposta para o problema 3-Partição for sim, o fluxo pode ser decomposto em $3r$ fluxos caminhos de custo 2, cada, da seguinte forma: seja $\{S_1, \dots, S_r\}$ uma solução para o primeiro problema. Para cada $a_i \in S_j$, com $1 \leq j \leq r$, existe um fluxo caminho sa_iqb_jt com valor $v(a_i)$, usando duas cores (j e $r+1$), na rede. Logo, existe uma decomposição do fluxo com custo $6r$.

Considere que o (s, t) -fluxo x definido na rede \mathcal{N} acima pode ser decomposto em fluxos caminhos a um custo total no máximo $k = 6r$. Na verdade, esse custo não pode ser inferior a $6r$, pois há exatamente $3r$ caminhos arco-disjuntos de s a q e todo caminho de s a t possui duas cores. São necessários, portanto, $3r$ (s, t) -fluxos caminhos bicromáticos. Cada vértice a_i , para $1 \leq i \leq 3r$, deve estar em exatamente um fluxo caminho, cujo valor é $v(a_i)$. Note que há exatamente três fluxos caminhos com um vértice b_j em comum, para $1 \leq j \leq r$, pela restrição para os valores de a_i ($T/4 < v(a_i) < T/2$). Sejam eles sa_xqb_jt , sa_yqb_jt e sa_zqb_jt . Para cada três fluxos caminhos desses, construímos um subconjunto $S_j = \{a_x, a_y, a_z\}$, com $v(a_x) + v(a_y) + v(a_z) = T$. \square

Na redução proposta no Teorema 4.4, a rede possui um número grande de cores, que é aproximadamente $1/8$ do seu número de arcos, e o fluxo dado não é uniforme. Uma linha de raciocínio natural, ao se deparar com um problema \mathcal{NP} -Completo, é impor restrições à entrada, para tentar obter soluções eficientes (polinomiais). Separamos os casos considerando o número de cores e, em seguida, o número de valores distintos de fluxo nos arcos da rede.

4.3.1 Redes com apenas uma cor

Para redes com apenas uma cor (monocromáticas), minimizar o custo da decomposição corresponde a minimizar o número de caminhos desta (ver Seção 3.3).

Para fluxos λ -uniformes, o problema pode ser resolvido em tempo polinomial, como consequência do Lema 3.9.

Quando há apenas dois valores distintos de fluxo nos arcos da rede, e o menor divide o maior, o problema pode ser resolvido em tempo polinomial (ver Teorema 3.10).

Estudamos o caso em que um dos valores não é múltiplo do outro. Para isso, propusemos o Algoritmo 1. Este recebe como parâmetro uma rede \mathcal{N} , vértices s e t , um (s, t) -fluxo x e dois valores inteiros, a e b (valores de fluxo nos arcos de \mathcal{N} , sendo $a \geq b$), e retorna um vetor P com a quantidade de caminhos de cada iteração da decomposição de x . Isso é feito com base no *Algoritmo de Euclides* para cálculo do MDC de dois valores inteiros positivos. Ele usa a operação $Suporte(\mathcal{N}, x, a)$, definida na página 38.

Observe que, após cada iteração do laço *enquanto* do Algoritmo 1, o valor de a é reduzido a no máximo $a/2$. Há duas situações:

Caso 1: se $b \leq a/2$, então $a \bmod b < b \leq a/2$;

Caso 2: se $b > a/2$, então $a \bmod b = a - b < a/2$.

Portanto, os valores dos parâmetros a e b são reduzidos, alternadamente, a no máximo metade do valor anterior. Consequentemente, o número de iterações realizadas pelo Algoritmo 1 é $O(\log a + \log b) = O(\log a)$, pois $a \geq b$. Seja $f(m, n)$ a função de complexidade de tempo do algoritmo usado para calcular fluxo máximo em uma rede com m arcos e n vértices (ver algoritmos de fluxo máximo no Apêndice B). A complexidade de tempo do Algoritmo 1 é, portanto, $O(f(m, n) \cdot \log a)$.

Algoritmo 1: DecomporFluxo2V($\mathcal{N}, s, t, x, a, b$)

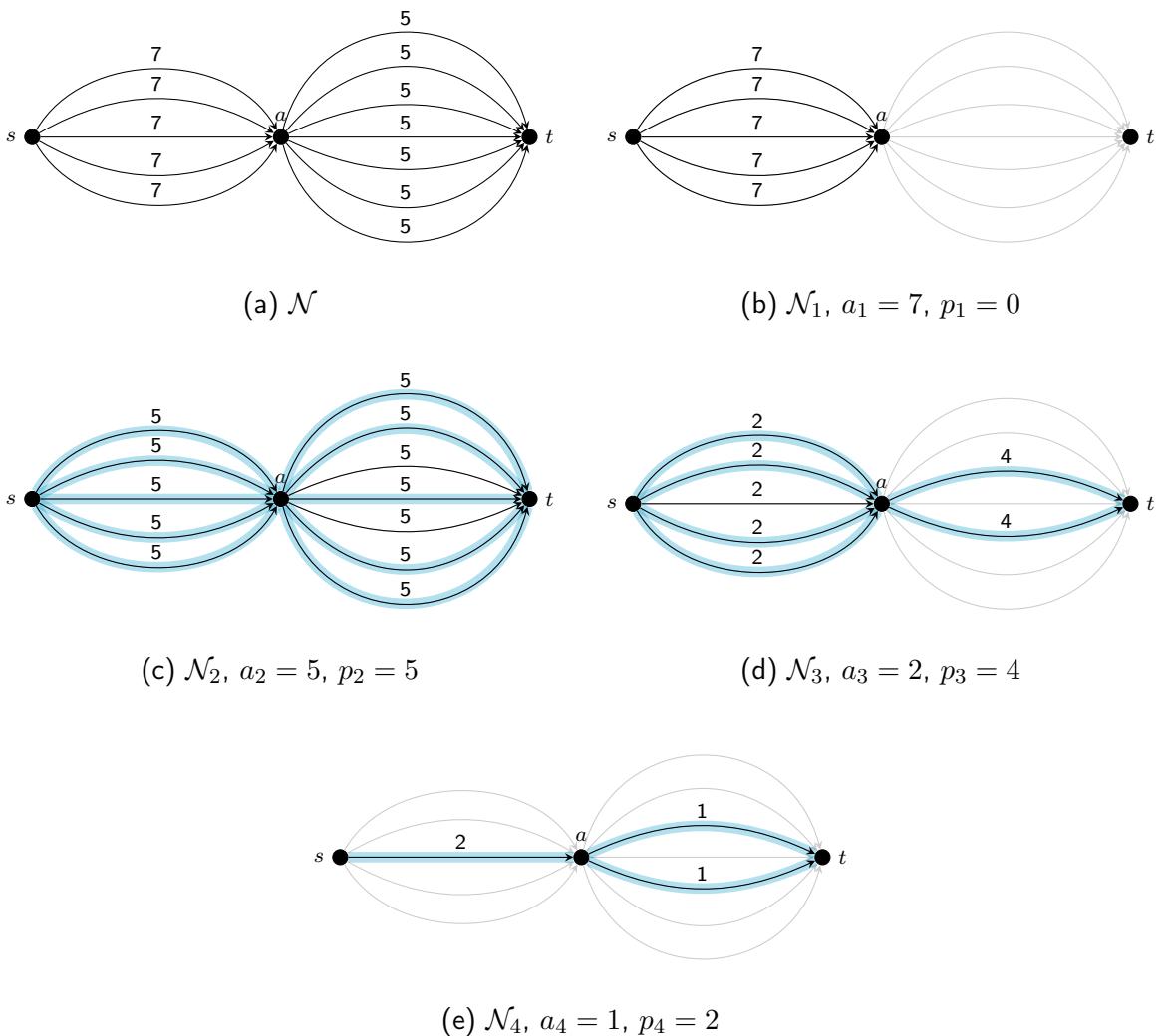
```

1 início
2    $i \leftarrow 1$ ;
3   enquanto ( $b > 0$ ) faça
4      $\mathcal{N}_i \leftarrow Suporte(\mathcal{N}, x, a)$ ;
5     Atualizar a capacidade de cada arco  $ij$  de  $\mathcal{N}_i$  para  $\lfloor x_{ij}/a \rfloor \cdot a$ ;
6     Calcular um  $(s, t)$ -fluxo máximo  $x'$  em  $\mathcal{N}_i$ ;
7      $P[i] \leftarrow |x'|/a$ ;
8      $x \leftarrow x - x'$ ; // Atualizar o fluxo  $x$  nos arcos em que  $x'$  é positivo
9      $r \leftarrow a \bmod b$ ;
10     $a \leftarrow b$ ;
11     $b \leftarrow r$ ;
12     $i \leftarrow i + 1$ ;
13  fim
14   $\mathcal{N}_i \leftarrow Suporte(\mathcal{N}, x, a)$ ;
15  Calcular um  $(s, t)$ -fluxo máximo  $x'$  em  $\mathcal{N}_i$ ;
16   $P[i] \leftarrow |x'|/a$ ;
17  retorna  $P$ ;
18 fim
```

Sejam $P = \langle p_1, \dots, p_m \rangle$ o vetor retornado pelo Algoritmo 1 e a_i o valor de a utilizado a cada iteração i . Note que $a_1 > \dots > a_m$ e que cada p_i é obtido a partir do fluxo máximo em uma rede com capacidades pelo menos a_i . Como $a_m | a_{m-1}$, o fluxo x pode ser decomposto em $\sum_{i=1}^m p_i$ fluxos caminhos.

No cálculo de p_1 , todos os arcos da rede possuem capacidade $a_1 = a$. Para $i > 1$, os valores de capacidade nos arcos da rede estão em $\{a_i, \lfloor a_{i-1}/a_i \rfloor \cdot a_i\}$, ou seja, no máximo dois valores e o menor destes divide o maior e não há fluxos caminhos de valor maior que a_i . É possível obter p_i , pelo Lema 3.9. Além disso, pela Condição de Conservação do Fluxo (Equação (2.4)), após o cálculo de p_i , todos os arcos com capacidade maior que a_i devem ter sido utilizados por (s, t) -fluxos caminhos de valor a_i .

Figura 27 – Exemplo da decomposição produzida pelo Algoritmo 1.



Fonte: Elaborada pelo autor.

A decomposição produzida pelo Algoritmo 1 está ilustrada na Figura 27. O rótulo em cada arco representa sua cor. Nós temos em (a) a rede original e um fluxo de valor 35. Em (b), a rede N_1 com arcos cuja capacidade é pelo menos $a_1 = 7$. Em (c), a rede N_2 com arcos cuja capacidade é pelo menos $a_2 = 5$. Note que os arcos com capacidade 7 em

N_1 tiveram sua capacidade ajustada para $\lfloor 7/5 \rfloor \cdot 5 = 5$. Em (d), a rede N_3 com arcos cuja capacidade é pelo menos $a_3 = 2$. Similarmente ao caso anterior, os arcos com capacidade 5 em N_2 tiveram sua capacidade ajustada para $\lfloor 5/2 \rfloor \cdot 2 = 4$. Por fim, em (e), temos a rede N_4 com arcos cuja capacidade é pelo menos 1. O número de caminhos nessa decomposição é dado por $p_1 + p_2 + p_3 + p_4 = 0 + 5 + 4 + 2 = 11$.

O Algoritmo também funciona quando $a = b$. Nesse caso, o (s, t) -fluxo x é a -uniforme e o vetor retornado é $P = \langle p_1 \rangle$, onde $p_1 = |x|/a$. Se $a \neq b$ e o menor valor dos dois divide o maior (suponha que $b|a$), o resultado produzido pelo algoritmo é ótimo. Isso corresponde ao resultado do Teorema 3.10. Os resultados a seguir se referem ao caso em que $a > b$, $b \nmid a$ e o digrafo sobre o qual a rede está definida é acíclico.

Lema 4.5. *Seja uma rede \mathcal{N} definida sobre um DAG e a e b os valores de fluxo nos arcos, com $a > b$ e $b \nmid a$. Seja $P = \langle p_1, \dots, p_m \rangle$ o vetor retornado pelo Algoritmo 1. Após o cálculo de cada p_i , há no máximo 2 valores distintos de fluxo nos arcos de \mathcal{N} .*

Demonstração. Seja a_i o valor de a usado para o cálculo de p_i em cada iteração i . Cada p_i é obtido do fluxo máximo em uma rede com capacidade pelo menos a_i . Como $a > b$, note que $a_1 = a$, $a_2 = b$ e $a_i = a_{i-2} \bmod a_{i-1}$, para $3 \leq i \leq m$, e que $a_m \mid a_{m-1}$.

Vamos proceder com indução no tamanho de P . No início, os valores de fluxo nos arcos são a_1 e a_2 . Após o cálculo de p_1 , o valor de fluxo em cada arco está em $\{a_1, a_2\}$. Da mesma forma, após o cálculo de p_2 , o valor de fluxo em cada arco está em $\{a_2, a_3\}$. Seja $k < m$ o último valor para o qual isso foi observado, ou seja, após o cálculo de p_k , há no máximo dois valores de fluxo nos arcos da rede. Vamos mostrar que isso também vale para o cálculo de p_{k+1} .

Se $k+1 < m$, então, após o cálculo de p_{k+1} , o valor de fluxo em cada arco deve estar em $\{a_{k+1}, a_{k+2}\}$. Suponha que existe pelo menos um arco uv com valor de fluxo a_k e seja P' um (s', t') -fluxo caminho maximal de valor a_k que contém uv . Note que $s' \neq s$ ou $t' \neq t$, pois o valor de p_k é máximo. Se $s' \neq s$, então, como $b(s') = 0$, há pelo menos a_k unidades de fluxo entrando em s' por (s, s') -caminhos de valor a_{k+1} . Seja P'_s um desses caminhos. Caso contrário, se $s' = s$ então P'_s é vazio. De forma análoga, Se $t' \neq t$, então, como $b(t') = 0$, há pelo menos a_k unidades de fluxo saindo de t' por (t', t) -caminhos de valor a_{k+1} . Seja P'_t um desses caminhos. Caso contrário, se $t' = t$ então P'_t é vazio. É possível enviar a_{k+1} unidades de fluxo pelo (s, t) -caminho $P'_s P' P'_t$. Isso é uma contradição, pois o valor de p_{k+1} é máximo.

Se $k+1 = m$, então $a_{k+1} \mid a_k$. Todos os fluxos caminhos terão valor a_{k+1} e, após o cálculo de p_{k+1} , o valor de fluxo em cada arco é zero e o resultado segue. \square

Pela demonstração do Lema 4.5, quando $a = a_i$, para $i > 1$, não há mais caminhos aumentantes de capacidade a_{i-1} .

Teorema 4.6. *Se a rede \mathcal{N} for acíclica, a solução produzida pelo Algoritmo 1 é ótima.*

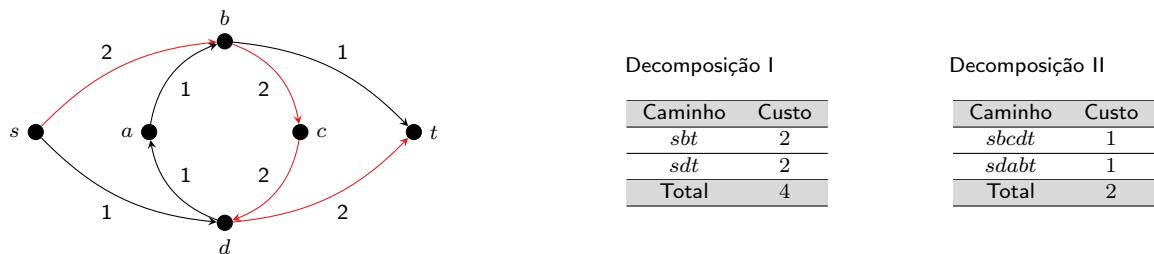
Demonstração. Sejam $P = \langle p_1, \dots, p_m \rangle$ e a_i o valor de a no cálculo de p_i , com $1 \leq i \leq m$. Note que, para todo $1 \leq i < m$, $a_i > a_{i+1}$. Pelo Lema 4.5, após cada iteração i , há no máximo dois valores distintos de fluxo nos arcos da rede, os quais devem estar em $\{a_i, a_{i+1}\}$, e não há caminhos aumentantes de valor a_i . Portanto, não é possível aumentar o valor de algum p_i reduzindo o valor de um ou mais p_j , com $i < j \leq m$. Por outro lado, a redução de uma unidade em algum p_i implica o acréscimo de $\lceil a_i/a_{i+1} \rceil \geq 2$ fluxos caminhos nas etapas seguintes. Logo, o número de fluxos caminhos da decomposição de x realizada pelo Algoritmo 1, que é dado por $p_1 + \dots + p_m$, é mínimo. \square

O problema permanece em aberto quando há apenas dois valores distintos de fluxo nos arcos e a rede não é acíclica. Para redes em que há três valores distintos de fluxo nos arcos, mesmo que esses sejam 1, 2 e 4, o problema é \mathcal{NP} -Completo (ver Teorema 3.11).

4.3.2 Redes com apenas duas cores

Iniciamos considerando o caso de fluxos uniformes em redes bicromáticas. Diferentemente do que acontece na versão monocromática, uma circulação pode influenciar no custo da decomposição do fluxo. Isso pode ser visto na Figura 28. Nesta, cada arco possui fluxo igual a λ e cor representada por seu rótulo.

Figura 28 – Influência de ciclos na decomposição de fluxo uniforme em redes coloridas.



Fonte: Elaborada pelo autor.

Com o resultado a seguir, mostramos que o Problema *DFColCustoMin*, que é a versão de otimização do problema *DFColCustoK*, pode ser resolvido em tempo polinomial para fluxos λ -uniformes e redes bicromáticas.

Teorema 4.7. *DFColCustoMin pode ser resolvido em tempo polinomial, se a rede possuir exatamente duas cores e o (s, t) -fluxo x for λ -uniforme.*

Demonstração. Como x é λ -uniforme, ele pode ser decomposto em $p = |x|/\lambda$ (s, t) -fluxos caminhos de valor λ . Esses caminhos são arco-disjuntos, e portanto p é mínimo. Tomemos uma rede $\mathcal{N}^k = (D^k, u^k)$, com $D^k = (V, A^k)$, a partir de \mathcal{N} e de x , apenas com os arcos de cor k , fazendo $u_a^k = x_a$, $\forall a \in A^k$, para $k \in \{1, 2\}$; calculamos o fluxo máximo x^k , que é λ -uniforme, e temos a quantidade de fluxos caminhos de valor λ neste, que é dada por $p_k = |x^k|/\lambda$. Logo, o número de fluxos caminhos bicromáticos é $p_{12} = p - p_1 - p_2$. O custo dessa solução é $p_1 + p_2 + 2p_{12}$, que é mínimo, pois p_1 e p_2 são máximos. \square

Esse resultado pode ser generalizado para o caso em que a rede possui duas cores e dois valores de fluxo distintos nos arcos, onde o menor destes divide o maior, e cada cor está associada a um valor de fluxo.

Teorema 4.8. *DFColCustoMin pode ser resolvido em tempo polinomial, se houver exatamente duas cores distintas (1 e 2) e dois valores de fluxo distintos (v_1 e v_2) nos arcos da rede, com $v_1 > v_2 \mid v_1$, e se o fluxo em todo arco de cor i for igual v_i , com $i \in \{1, 2\}$.*

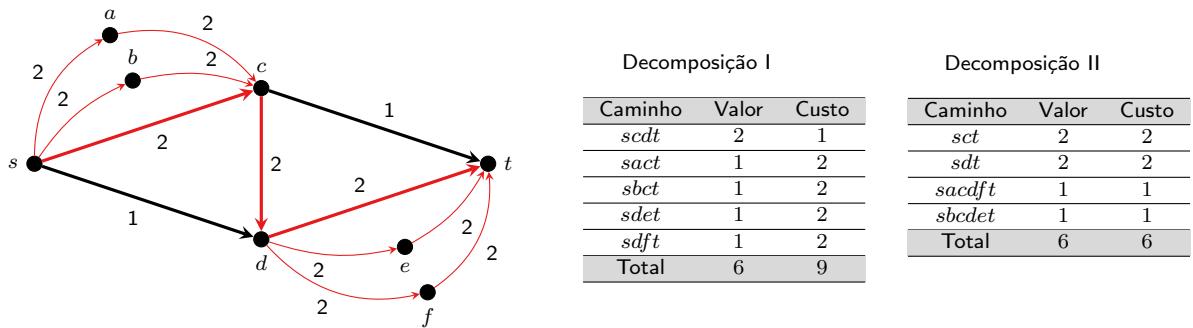
Demonstração. Sejam \mathcal{N} uma rede com as características apresentadas e um (s, t) -fluxo x nesta. Note que $v_1 = k \cdot v_2$ para algum $k \in \mathbb{Z}_+^*$. Substituindo cada arco de cor 1 por k arcos de cor 1 e com valor de fluxo v_2 , com as mesmas extremidades, obtemos um (s, t) -fluxo x' v_2 -uniforme, com $|x'| = |x|$. Cada (s, t) -fluxo caminho de cor 1 de x corresponde a k (s, t) -fluxos caminhos de cor 1 e valor v_2 de x' . Aplicando a decomposição do Teorema 4.7, obtemos o custo mínimo $p_1 + p_2 + 2 \cdot p_{12}$, que corresponde ao custo mínimo $p_1/k + p_2 + 2 \cdot p_{12}$ de uma decomposição de x . \square

Com o resultado acima, resolvemos um caso bem específico de redes com duas cores (1 e 2) e com valores de fluxo v_1 e v_2 . Ficam em aberto os casos a seguir:

1. $v_2 \mid v_1$ e não há correspondência entre cores e valores de fluxo;
2. $v_2 \nmid v_1$ e todo arco de cor i tem valor de fluxo v_i , com $i \in \{1, 2\}$;
3. $v_2 \nmid v_1$ e não há correspondência entre cores e valores de fluxo.

Considerando o caso 1, mesmo em um DAG, a solução ótima não consiste em maximizar os caminhos monocromáticos e depois os bicromáticos de maior valor de fluxo; seguidos dos monocromáticos e bicromáticos de menor valor de fluxo. Isso pode ser observado na rede da Figura 29. Nesta, os arcos finos têm valor de fluxo 1, os arcos espessos têm valor de fluxo 2 e o rótulo de cada arco indica a sua cor. A decomposição I, que maximiza os caminhos monocromáticos de maior valor, tem custo 9; já a decomposição II, que é ótima, tem custo 6.

Figura 29 – Rede bicromática e acíclica com dois valores de fluxo nos arcos.



Fonte: Elaborada pelo autor.

Em redes com mais de uma cor e três valores de fluxo distintos nos arcos, mesmo que sejam potências de 2, o problema *DFColCustoMin* é \mathcal{NP} -Difícil. Considerando uma rede nessas condições, com o resultado a seguir, será mostrada uma redução para o *DFColCustoK* a partir do *FluxoKDiv*, que é \mathcal{NP} -Completo (ver Teorema 3.11).

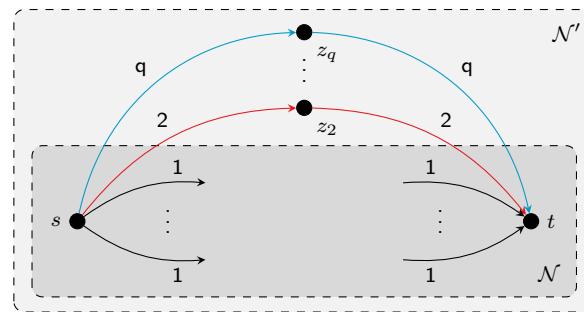
Teorema 4.9. *DFColCustoK* é \mathcal{NP} -Completo em redes com duas ou mais cores e o conjunto de valores de fluxo nos arcos é $\{1, 2, 4\}$.

Demonstração. Será mostrada uma redução polinomial do problema *FluxoKDiv* para o *DFColCustoK*. Dada uma instância $\langle \mathcal{N} = (D, u), x, k \rangle$ do *FluxoKDiv*, em que o valor de fluxo em cada arco está em $\{1, 2, 4\}$, construímos uma instância $\langle \mathcal{N}' = (D^c, u'), x', k' \rangle$ do *DFColCustoK*, com $q \geq 2$ cores, da seguinte maneira:

- Seja $D = (V, A)$, com dois vértices especiais $s, t \in V$, então $D^c = (V', A', c)$, com $V' = V \cup \{z_i \mid 2 \leq i \leq q\}$ e $A' = A \cup \{sz_i, z_it \mid 1 \leq i \leq q\}$; e uma coloração $c : A' \rightarrow \{1, \dots, q\}$, onde $c(a) = 1$, se $a \in A$; ou $c(a) = i$, se z_i é uma das extremidades de a ;
- definimos $x'(a) = x(a)$, se $a \in A$; ou $x'(a) = 1$, caso contrário;
- definimos $u'(a) = x'(a)$, para todo $a \in A'$;
- fazemos $k' = k + q - 1$

Essa construção está ilustrada na Figura 30. Nesta, os rótulos próximos aos arcos indicam as cores destes.

Figura 30 – Redução do Problema *FluxoKDiv* para o *DFColCustoK*.



Fonte: Elaborada pelo autor.

O (s, t) -fluxo x em \mathcal{N} pode ser decomposto em k (s, t) -fluxos caminhos se e somente se x' em \mathcal{N}' puder ser decomposto em (s, t) -fluxos caminhos a um custo k' .

Seja $x^1 \dots x^k$ uma decomposição de x em (s, t) -fluxos caminhos. Todo fluxo caminho em \mathcal{N} é um fluxo caminho de custo 1 em \mathcal{N}' . Os caminhos sz_it , para $2 \leq i \leq q$, são monocromáticos e, portanto, os fluxos caminhos ao longo deles possuem custo 1, cada. Assim, x' admite uma decomposição em q fluxos caminhos, $x^1, \dots, x^k, \dots, x^{q-1}$ monocromáticos e o custo total desta é $k + q - 1$.

Agora, considere uma decomposição de x' em r (s, t) -fluxos caminhos a um custo total de $k + q - 1$. Note que, em \mathcal{N}' , os caminhos sz_{it} , para $2 \leq i \leq q$, são arco-disjuntos e monocromáticos. Logo, eles contribuem com um custo $q - 1$ para o custo total da solução. Os outros fluxos caminhos, também monocromáticos, possuem um custo total $k = r - (q - 1)$. Como estes também são fluxos caminhos em \mathcal{N} , eles correspondem a uma decomposição de x em k fluxos caminhos. \square

4.3.3 Redes com pelo menos três cores

Para redes com três cores, mesmo com um (s, t) -fluxo uniforme, mostramos com o Teorema 4.10 que o problema $DFColCustoMin$ permanece \mathcal{NP} -Difícil. Fazemos isso com uma redução a partir do Problema *Weak 2-Linkage* (descrito a seguir), que segundo Fortune, Hopcroft e Wyllie (1980) é \mathcal{NP} -Completo.

Problema: Weak 2-Linkage

Entrada: Um digrafo $D = (V, A)$ e 4 vértices u_1, u_2, v_1 e $v_2 \in V$.

Pergunta: Há 2 caminhos arco-disjuntos, um de u_1 a v_1 e outro de u_2 a v_2 , em D ?

Teorema 4.10. O $DFColCustoK$ é \mathcal{NP} -Completo mesmo quando o fluxo é uniforme e há apenas três cores distintas nos arcos da rede.

Demonstração. Como já foi visto no Teorema 4.4, o $DFColCustoK$ pertence à classe \mathcal{NP} . Será mostrada uma redução polinomial do *Weak 2-Linkage* para o $DFColCustoK$. Dada uma instância $\langle D = (V, A), u_1, u_2, v_1, v_2 \rangle$, com $u_1, u_2, v_1, v_2 \in V$, do *Weak 2-Linkage*, será construída uma rede $\mathcal{N} = (D^c, u \equiv \lambda)$, com 3 cores, e um (s, t) -fluxo λ -uniforme x , instância do $DFColCustoK$. Sejam $n = |V|$ e $m = |A|$. Note que $n \geq 4$ e $m \geq 2$. Caso contrário, teríamos uma instância *não* do problema *Weak 2-Linkage*.

Inicialmente, será construído o multidigrafo colorido $D^c = (V', A', c)$, com $c : A' \rightarrow \{1, 2, 3\}$, sobre o qual a rede \mathcal{N} será definida, da seguinte maneira:

- $V' = V \cup \{s, t, s_1, s_2, t_1, t_2\}$;
- Os arcos em A' serão adicionados em etapas, descritas a seguir:
 - Todo arco de A passa a ser também um arco em A' , com a cor 3;
 - Adicionam-se 4 arcos - de s para u_1 , de s para u_2 , de v_1 para t , e de v_2 para t . Cada arco desses, com extremidade u_i ou v_i , tem cor i , para $i \in \{1, 2\}$;
 - Adicionam-se $m - 2$ arcos de cor 1 de s para s_1 ; $m - 2$ arcos de cor 3, de s_1 para s_2 e de t_1 para t_2 ; e $m - 2$ arcos de cor 2, de t_2 para t ;
 - Adicionam-se $d_D^+(u_i) - 1$ arcos de cor 2 de s_2 para u_i , e $d_D^-(v_i) - 1$ arcos de cor 1 de v_i para t_1 , para $i \in \{1, 2\}$;
 - Para todo $v \in V \setminus \{u_1, u_2\}$, adicionam-se $d_D^+(v)$ arcos de cor 2 de s_2 para v ;
 - Para todo $v \in V \setminus \{v_1, v_2\}$, adicionam-se $d_D^-(v)$ arcos de cor 1 de v para t_1 ;

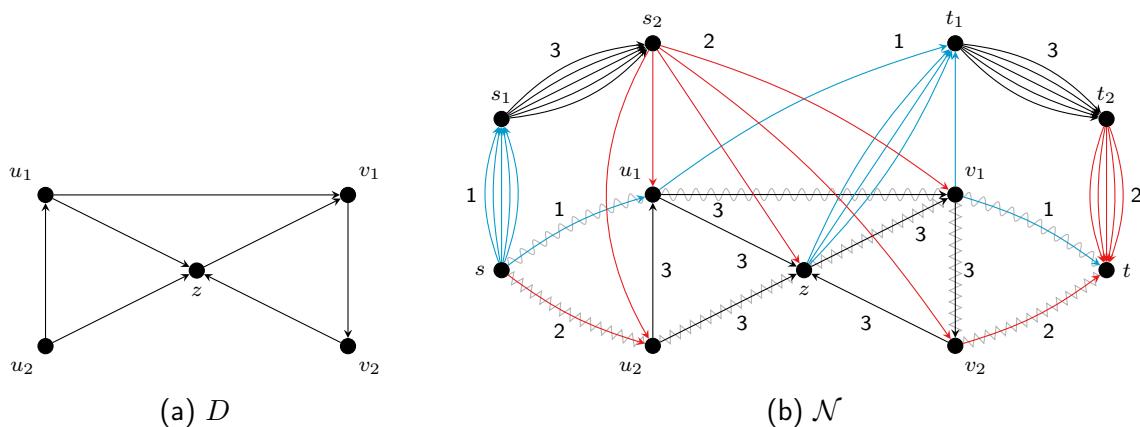
Para completar a definição de \mathcal{N} , fazemos $u(a) = x(a) = \lambda$, para todo $a \in A'$. Observe que $b_x(v) = 0$ para todo $v \in V' \setminus \{s, t\}$, e que $b_x(s) = -b_x(t) = m \cdot \lambda$, que é o valor do (s, t) -fluxo x . Este pode ser decomposto em m fluxos caminhos, cada um destes com valor λ . O multidigrafo D^c obtido possui $n + 6$ vértices e $6(m - 2) + m + 4 = 7m - 8$ arcos. A construção, ilustrada na Figura 31, é portanto polinomial no tamanho da entrada.

Observe que, para cada arco uv de A , existe um caminho com três cores de s a u ou de v a t em D^c . Assim, o custo de qualquer decomposição do (s, t) -fluxo x em m (s, t) -fluxos caminhos é no máximo $3m$. Será mostrado que D possui dois caminhos arco-disjuntos, um de u_1 a v_1 e um de u_2 a v_2 , se e somente se o (s, t) -fluxo x admitir uma decomposição em m (s, t) -fluxos caminhos com custo $3m - 2$.

Assuma que D possui os dois caminhos arco-disjuntos. Esses também são caminhos em D^c . Adicionando os arcos de s a u_i e de v_i a t , ambos com a cor i , para $i \in \{1, 2\}$, temos dois fluxos caminhos bicromáticos. Os outros $m - 2$ fluxos caminhos possuem três cores. Logo, o custo dessa decomposição é $3(m - 2) + 2 \cdot 2 = 3m - 2$.

Agora, assuma que o (s, t) -fluxo x pode ser decomposto em m fluxos caminhos, x^1, \dots, x^m , com custo $3m - 2$. Como x é λ -uniforme, m é mínimo e os fluxos caminhos da decomposição são arco-disjuntos dois a dois. Cada fluxo caminho possui de duas a três cores. Como há no máximo dois fluxos caminhos com duas cores, há pelo menos $m - 2$ fluxos caminhos com três cores. Na verdade, há exatamente $m - 2$ fluxos caminhos com 3 cores. Caso contrário, o custo desta seria $3(m - 2 + k) + 2(2 - k) = 3m - 2 + k > 3m - 2$, para $k \in \{1, 2\}$. Os arcos de cor 3 ou estão em A , ou são de s_1 para s_2 ou de t_1 para t_2 . Estes dois últimos tipos devem estar necessariamente em fluxos caminhos com três cores. Os fluxos caminhos com duas cores possuem a cor 3 e uma segunda cor (ou 1 ou 2). Nestes, os únicos arcos com cor diferente de 3 são o primeiro, de s para u_i , e o último, de v_i para t , com $i \in \{1, 2\}$. Removendo estes arcos inicial e final, obtemos dois caminhos, de u_1 para v_1 e de u_2 para v_2 , que são arco-disjuntos em D . \square

Figura 31 – Exemplo de redução do *Weak 2-Linkage* para o *DFColCustoK*.



Fonte: Elaborada pelo autor.

A Figura 31 traz um exemplo da redução apresentada no Teorema 4.10. Em (a) temos uma entrada do *Weak 2-Linkage*; e em (b), uma entrada para o *DFColCustoK*, em que todos os arcos possuem capacidade e fluxo iguais a λ . Os rótulos nos arcos indicam a cor destes. O fluxo pode ser decomposto em $m = 7$ fluxos caminhos arco disjuntos (2 bicromáticos e 5 tricromáticos), com custo 19.

O resultado do Teorema 4.10 pode ser generalizado para o caso em que há mais de um valor de fluxo no conjunto de arcos da rede. Para isso, basta adicionar à rede da redução um arco com cor 3 de s a t para cada novo valor de fluxo. Daí, temos:

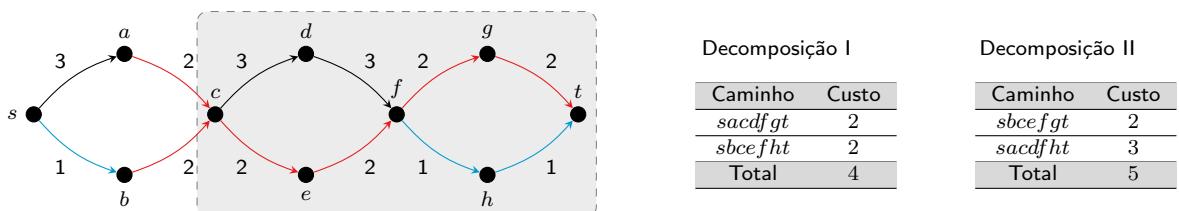
Corolário 4.11. *DFColCustoK* é \mathcal{NP} -Completo para redes com três cores.

De acordo com Bang-Jensen, Havet e Maia (2015), o problema *Weak 2-Linkage* é \mathcal{NP} -Completo mesmo em digrafos de grau no máximo 3. A rede da redução proposta no Teorema 4.10 pode ser modificada de maneira que, com exceção de s e de t , todo vértice tenha grau no máximo 6. Para isso, basta tomar como base uma instância do *Weak 2-Linkage* em que todo vértice tem grau no máximo 3 e transformar cada multicaminho ss_1s_2 e t_1t_2t em $m - 2$ multicaminhos do tipo $ss_1^i s_2^i$ e $m - 2$ multicaminhos do tipo $t_1^i t_2^i t$, com $1 \leq i \leq m - 2$. Com exceção de s e de t , cada vértice desses caminhos tem grau 2 e cada vértice do digrafo original terá seu grau dobrado. Daí, temos o seguinte resultado:

Corolário 4.12. *O Problema DFColCustoK é \mathcal{NP} -Completo, mesmo quando restrito a redes com 3 cores em que, com exceção de s e t , todo vértice tem grau no máximo 6.*

Segundo Fortune, Hopcroft e Wyllie (1980), o Problema *Weak 2-Linkage* pode ser resolvido em tempo polinomial em *DAGs*. Decidimos analisar também o problema *DFColCustoK* para *DAGs* com 3 cores. Tomemos como exemplo a rede da Figura 32. Nesta, o rótulo de cada arco representa a sua cor. Observe que cada (s, t) -fluxo caminho possui pelo menos duas cores. Dependendo de como o primeiro caminho é selecionado, o caminho restante poderá ter 2 ou 3 cores, conforme decomposições I e II. Selecionando parte da primeira decomposição, que é ótima, considerando os trechos dos fluxos caminhos a partir do vértice c , note que não se obtém uma decomposição ótima para a rede da área destacada, pois há dois caminhos com duas cores, cada; ao passo que a solução ótima para esse trecho consiste em um caminho monocromático e um bicromático.

Figura 32 – Exemplo de rede sobre um *DAG* com 3 cores e fluxo uniforme.



Fonte: Elaborada pelo autor.

Como pode ser observado no exemplo anterior, mesmo em redes acíclicas com 3 cores e fluxo uniforme, o problema não possui a propriedade da *subestrutura ótima*, em que parte da solução ótima é solução ótima para parte do problema. De acordo com Cormen et al. (2022), essa propriedade é característica de problemas que podem ser solucionados de forma eficiente por algoritmos gulosos ou de programação dinâmica.

Mostramos que o *DFColCustoK* é \mathcal{NP} -Completo, mesmo para fluxos uniformes, em DAGs com pelo menos 5 cores. Isso foi feito com uma redução a partir do *Problema 1-em-3SAT* (apresentado a seguir), que segundo Schaefer (1978) é \mathcal{NP} -Completo.

Problema: 1-em-3SAT

Entrada: Uma fórmula φ na 3FNC, com m cláusulas e n variáveis.

Pergunta: Existe uma atribuição de valores verdade às variáveis de φ , de modo que cada cláusula possua apenas um literal verdadeiro?

Teorema 4.13. O *DFColCustoK* é \mathcal{NP} -Completo, mesmo com uma rede \mathcal{N} sobre um digrafo acíclico arco-colorido D^c com 5 ou mais cores e um (s, t) -fluxo uniforme x .

Demonstração. Note que o problema pertence à classe \mathcal{NP} , pois dada uma coleção de (s, t) -fluxos caminhos x^1, \dots, x^r , é possível verificar em tempo polinomial se isso é uma decomposição de x e se a soma das quantidades de cores dos caminhos é no máximo k . Será mostrada uma redução polinomial para esse problema, a partir do 1-em-3SAT. Dada uma instância φ deste, uma fórmula booleana com m cláusulas e n variáveis, será construída uma rede $\mathcal{N} = (D^c, u \equiv \lambda)$, com $n + 2$ cores, e um (s, t) -fluxo λ -uniforme x .

Inicialmente, criam-se dois vértices especiais s e t , e um vértice v_i para cada variável v_i , com $1 \leq i \leq n$, com dois arcos de s para ele, um de cor 1 e um de cor 2. Para cada cláusula C_j em φ , com $1 \leq j \leq m$, cria-se uma estrutura de cláusulas com vértices c_j^s , c_j^t e três arcos do primeiro para o segundo, um de cor 1 e dois de cor 2.

Agora é preciso construir os caminhos de s a t . Para cada variável v_i , serão construídos 2 desses caminhos. Se o literal v_i estiver em alguma cláusula, seja C_j, \dots, C_k , com $1 \leq j \leq k \leq m$, a sequência de cláusulas que possuem o literal v_i . Adicionar um arco do vértice v_i para o c_j^s . Se $j < k$, adicionar um arco de c_x^t para c_{x+1}^s , para $j \leq x < k$. Em seguida, adicionar um arco de c_k^t para t . Se não houver nenhuma cláusula com o literal v_i , criar um arco de v_i para t . Proceder de forma análoga para o literal \bar{v}_i . Todos os arcos mencionados aqui possuem a cor $i + 2$. Como a instância do 1-em-3SAT possui pelo menos 3 variáveis, o número de cores da rede aqui construída é pelo menos 5.

Pela construção descrita, ilustrada na Figura 33; além de s e t , são criados n vértices, um para cada variável, e $2m$ vértices, 2 para cada cláusula. Para cada variável são criados 4 arcos, e para cada cláusula são criados 6 arcos. Assim, a rede obtida possui exatamente $n + 2m + 2$ vértices e $4n + 6m$ arcos. A construção é, portanto, polinomial no tamanho da entrada φ .

Por fim, define-se $u(a) = x(a) = \lambda$, para todo arco a da rede. Dessa forma, o (s, t) -fluxo x , de valor $2n\lambda$, é λ -uniforme. Ele pode ser decomposto em $2n$ fluxos caminhos de s a t de valor λ , cada. Uma vez que os fluxos caminhos são arco-disjuntos, o número $2n$ é mínimo. Cada caminho possui pelo menos duas cores (a cor 1 ou 2 saindo de s para um vértice v_i e a cor $i + 2$ saindo deste). Assim, o custo qualquer decomposição de x em fluxos caminhos é pelo menos $4n$. Será mostrado que a fórmula φ no 1-em-3SAT é satisfatível se e somente se x puder ser decomposto em fluxos caminhos a um custo $4n$.

Observe que o número de fluxos caminhos de valor λ que passam pela estrutura correspondente a uma cláusula é exatamente 3. Destes, um passa pelo arco de cor 1 e cada um dos outros dois passa por um arco de cor 2.

Considere uma atribuição de valores às variáveis de φ , de modo que apenas um literal por cláusula seja igual a 1 e os demais iguais a 0. Para cada variável v_i de φ devem ser tomados dois (s, t) -fluxos caminhos, ambos com as cores $i + 2$ e uma segunda cor (1 para um caminho e 2 para o outro). Para cada variável v_i , pelo menos um de seus literais (v_i ou \bar{v}_i) deve estar em alguma cláusula de φ . Consequentemente, pelo menos um dos (s, t) -fluxos caminhos que passam por v_i deve passar por pelo menos uma das estruturas de cláusula. A seguir, será descrito como deve ser tomado cada um desses fluxos caminhos.

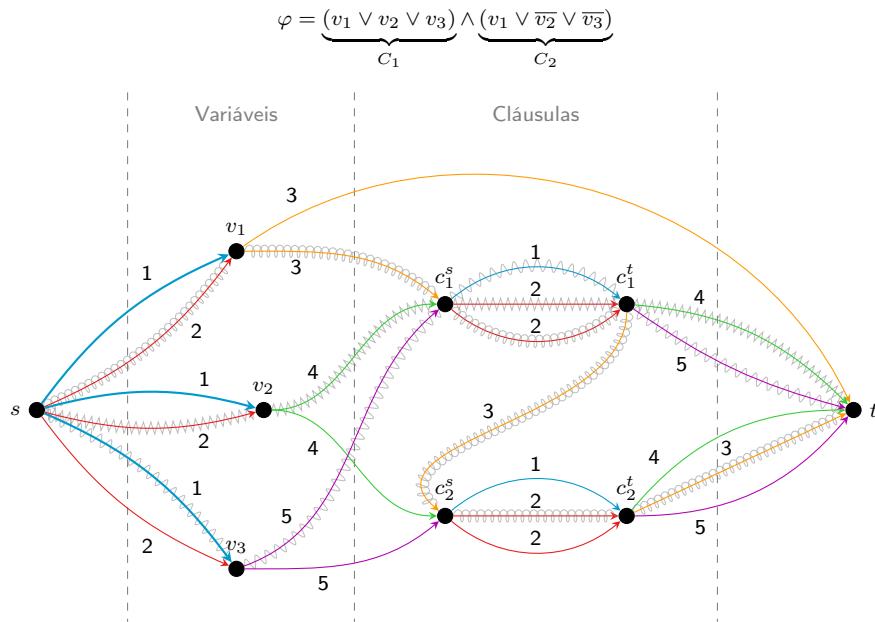
Se $v_i = 0$ (resp. $v_i = 1$), o primeiro (s, t) -fluxo caminho, com as cores 1 e $i + 2$, deve passar pelas estruturas correspondentes às cláusulas que possuem o literal \bar{v}_i (resp. v_i), caso haja alguma. Caso contrário, ele deve passar pelo arco de v_i a t . O segundo (s, t) -fluxo caminho, com as cores 2 e $i + 2$, deve passar pelas estruturas correspondentes às cláusulas que possuem o literal v_i (resp. \bar{v}_i), caso haja alguma. Caso contrário, ele deve passar pelo arco de v_i a t .

Considere agora que o (s, t) -fluxo x admite uma decomposição em (s, t) -fluxos caminhos com custo igual a $4n$. Como o menor número de fluxos caminhos de uma decomposição de x é $2n$ e o custo de cada um desses é pelo menos 2, há exatamente $2n$ (s, t) -fluxos caminhos, cada um com 2 cores. Note que cada (s, t) -fluxo caminho inicia com duas cores (ou 1 ou 2) de s para um vértice v_i e a cor $i + 2$ a partir deste, com $1 \leq i \leq n$. Logo, se um (s, t) -fluxo caminho tem as cores 1 e 2, então ele tem pelo menos três cores. Para cada vértice v_i , há pelo menos um e no máximo dois arcos dele para uma estrutura de cláusula. Selecione (arbitrariamente, caso haja mais de uma opção) um (s, t) -fluxo caminho que tem um arco de v_i para c_j^s , com $1 \leq j \leq m$. Se a cláusula C_j possui o literal v_i e o (s, t) -fluxo caminho selecionado tem a cor 1 (resp. 2), atribua 1 (resp. 0) à variável v_i em φ . De forma análoga, se a cláusula C_j possui o literal \bar{v}_i e o (s, t) -fluxo caminho selecionado tem a cor 1 (resp. 2), atribua 0 (resp. 1) à variável v_i em φ . Como cada estrutura de cláusula tem apenas um arco de cor 1, apenas um literal da cláusula correspondente em φ terá valor 1, condição para que a fórmula φ seja satisfeita no problema 1-em-3SAT. \square

A Figura 33 traz um exemplo de redução do 1-em-3SAT para o *DFCosCustoK*, com a rede obtida a partir de uma fórmula φ . Nesta, o rótulo em cada arco indica a sua cor.

Os (s, t) -fluxos caminhos destacados são os que passam por cada vértice v_i , com $1 \leq i \leq n$, e em seguida por um vértice de cláusula (no exemplo, o vértice c_1^s). A partir desses caminhos, pela redução do Teorema 4.13, é obtida a atribuição de valores que torna φ satisfatível no 1-em-3SAT. Observe que os caminhos que passam pelos vértices v_1 e v_2 têm a cor 2 e os literais v_1 e v_2 estão em C_1 . Deve-se atribuir o valor 0 a essas duas variáveis. O caminho que passa por v_3 tem a cor 1 e o literal v_3 está em C_1 . Portanto, deve-se atribuir o valor 1 à variável v_3 .

Figura 33 – Exemplo de Redução do 1-em-3SAT para o *DFColCustoK*.



Fonte: Elaborada pelo autor.

Pela redução proposta no Teorema 4.13, observa-se que, com exceção dos vértices s e t , todos os demais vértices possuem grau 4 ou 6. Daí, temos:

Corolário 4.14. *O DFColCustoK é NP-Completo, mesmo restrito a redes acíclicas com 5 cores ou mais em que, com exceção de s e t , todo vértice tem grau 4 ou 6.*

Para um (s, t) -fluxo x λ -uniforme em uma rede arco-colorida acíclica, se a função de coloração c for injetiva, o custo mínimo da decomposição é m . Isso porque a decomposição que minimiza o custo é feita em $|x|/\lambda$ fluxos caminhos arco-disjuntos. Nesse caso, cada arco da rede aparece em apenas um fluxo caminho da decomposição.

Fica em aberto o problema *DFColCustoK* para fluxos uniformes em redes acíclicas com 3 e com 4 cores. O Quadro 1 traz um resumo dos resultados vistos para o problema *DFColCustoK* neste capítulo, fazendo a relação entre quantidade de cores e a quantidade de valores distintos de fluxo nos arcos da rede. Com exceção do Lema 3.9 e dos Teoremas 3.10 e 3.11, de Hartman et al. (2012), os demais resultados são nossos.

Quadro 1 – Resultados de complexidade para o problema $DFColCustoK$.

Valores Cores \	1	2	≥ 3
1	\mathcal{P} Lema 3.9	\mathcal{P}^* Teoremas 3.10 e 4.6	\mathcal{NPC} Teorema 3.11
2	\mathcal{P} Teorema 4.7	\mathcal{P}^{**} Teorema 4.8	\mathcal{NPC} Teorema 4.9
≥ 3	\mathcal{NPC} Teorema 4.10	\mathcal{NPC} Corolário 4.11	\mathcal{NPC} Teorema 4.9

* Permanece em aberto para redes com ciclos e dois valores quaisquer

** Apenas se cada cor está associada a um valor de fluxo e o menor valor divide o maior

Fonte: Elaborado pelo autor

5 FLUXOS RAMIFICADOS ARCO-DISJUNTOS

Neste capítulo, serão apresentados os conceitos de ramificações e fluxos ramificados e os resultados fundamentais para o nosso estudo sobre a caracterização de redes que admitem fluxos s -ramificados arco-disjuntos.

5.1 Ramificações

Uma s -ramificação em um digrafo D é uma s -arborescência (uma arborescência enraizada em s) geradora de D , ou seja, é um subgrafo de D em que todo vértice $v \in V \setminus s$ tem grau de entrada igual a 1 e o vértice s tem grau de entrada 0. Em outras palavras, se há exatamente um caminho direcionado de s para cada um dos outros vértices de D .

Há muitos estudos referentes a ramificações (por exemplo, em Bang-Jensen e Gutin (2008, Cap. 9)), dada a relevância desse tema, tanto do ponto de vista teórico como prático, por conta das diversas aplicações relacionadas. Um resultado clássico de Edmonds (1973) caracteriza os digrafos que contêm k s -ramificações arco-disjuntas.

Teorema 5.1 (Edmonds (1973)). *Um digrafo $D = (V, A)$ possui k s -ramificações arco-disjuntas se e somente se $d_D^-(X) \geq k$ para todo $X \subseteq V(D) \setminus \{s\}$, com $X \neq \emptyset$.*

Uma outra demonstração para esse mesmo resultado, apresentada por Lovász (1976), mostra um algoritmo polinomial para encontrar tais ramificações, se elas existirem. Uma versão mais forte desse resultado, permitindo que cada ramificação tenha uma raiz possivelmente diferente, pode ser vista em Schrijver (2003, Cap. 53). Entre as aplicações para o *Teorema de Edmonds*, destacamos que ele pode ser usado para provar o Teorema de Menger (1927), para caracterizar a conectividade em arcos em Shiloach (1979) e para caracterizar cobertura por ramificação em Frank (1979). Em Campos et al. (2020), foi estudado o problema de encontrar ramificações disjuntas no contexto de grafos temporais.

5.2 Fluxos s -ramificados

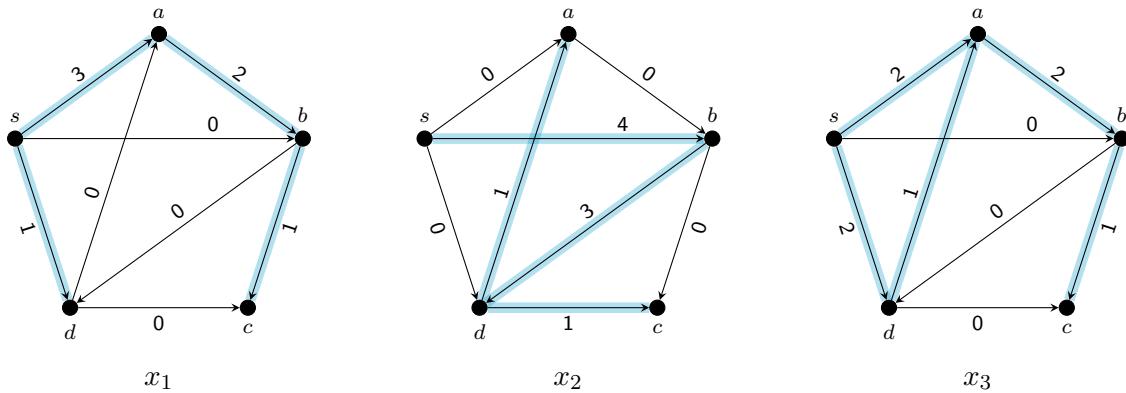
Seja uma rede $\mathcal{N} = (D, l \equiv 0, u)$ com um vértice especial s . Um fluxo s -ramificado nesta é um fluxo x tal que:

$$b_x(v) = \begin{cases} n - 1 & , \text{ se } v = s \\ -1 & , \text{ caso contrário} \end{cases} \quad (5.1)$$

Esse tipo de fluxo é útil para modelar situações em que se deseja enviar uma unidade de um dado produto, a partir de um ponto, para cada um dos outros pontos da rede. Cada ponto recebe uma certa quantidade do produto em questão, retém uma unidade e repassa a quantidade excedente, caso haja, adiante.

Na Figura 34 temos três exemplos de fluxos s -ramificados em uma rede \mathcal{N} em que todos os arcos possuem capacidade 4. Cada arco está associado a um rótulo que indica o valor de fluxo nele. Os arcos destacados representam o digrafo suporte de cada fluxo. Observe que os fluxos x_1 e x_2 são arco-disjuntos. Além disso, o digrafo suporte de cada um desses fluxos é uma s -ramificação. Isso não é observado para o fluxo x_3 .

Figura 34 – Exemplos de fluxos s -ramificados em uma rede $\mathcal{N} = (D, u \equiv 4)$.



Fonte: Elaborada pelo autor.

Um fluxo em uma rede pode ser definido em termos do seu vetor de balanço e o objetivo de um problema de fluxo pode ser visto como sendo encontrar um fluxo x cujo vetor de balanço respeite um dado conjunto de propriedades. Por exemplo, a versão de decisão do Problema de Fluxo Máximo pode ser vista como:

Problema: Fluxo Máximo

Entrada: Uma rede $\mathcal{N} = (D, u, b)$, com $s, t \in V(D)$, $b(s) = -b(t) = k$ e $b(v) = 0$ para todo vértice em $V \setminus \{s, t\}$, e um inteiro positivo k .

Pergunta: \mathcal{N} admite um fluxo x de valor k tal que $b_x = b$?

Como já foi visto, esse problema pode ser resolvido em tempo polinomial. Portanto, pelo Lema 5.2, com uma redução simples para o problema de fluxo máximo, o problema de decidir se uma rede admite um fluxo com um dado vetor de balanço prescrito pode ser resolvido em tempo polinomial.

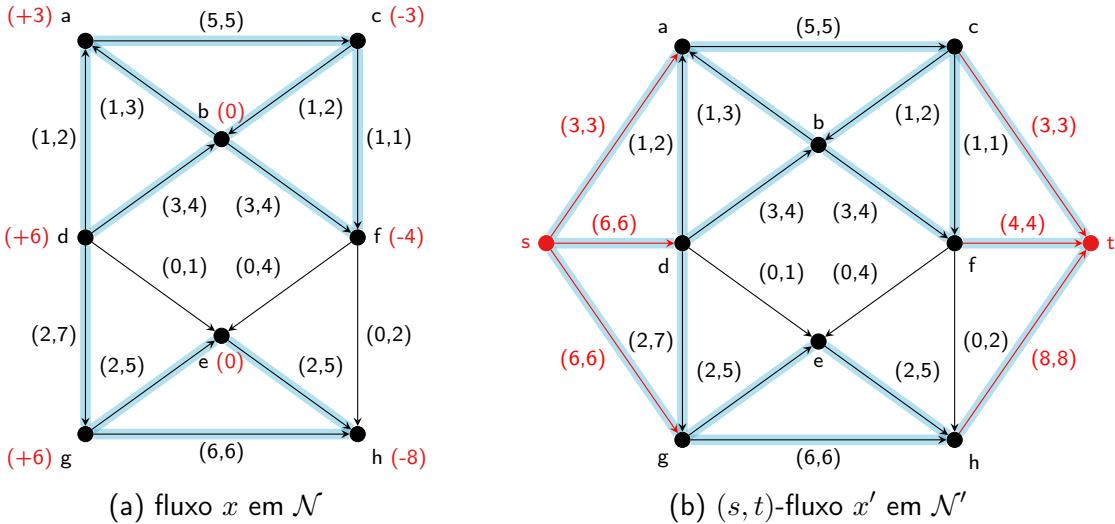
Lema 5.2 (Adaptado de Bang-Jensen e Gutin (2008)). *Considere uma rede $\mathcal{N} = (D, u)$ e um vetor de balanço prescrito b . Sejam $M = \sum_{v|b(v)>0} b(v)$ e $\mathcal{N}' = (D', u')$ uma rede definida como segue:*

- $V(D') = V(D) \cup \{s', t'\}$;
- $A(D') = A(D) \cup \{s'u : u \in V(D), b(u) > 0\} \cup \{vt' : v \in V(D), b(v) < 0\}$;
- $u'(a) = u(a), \forall a \in A(D)$, $u'(s'u) = b(u)$ se $b(u) > 0$, e $u'(vt') = -b(v)$ se $b(v) < 0$.

Então, \mathcal{N} admite um fluxo x com vetor de balanço b se e somente se \mathcal{N}' admite um (s', t') -fluxo x' de valor M .

Um exemplo da construção apresentada no Lema 5.2 pode ser observado na Figura 35. Ela traz, em (a), uma rede \mathcal{N} e um fluxo x (considere que o vetor de balanço de x é o vetor de balanço prescrito da rede) em que a soma dos balanços positivos é igual a 15; e em (b) uma rede \mathcal{N}' , obtida a partir de \mathcal{N} , e um (s, t) -fluxo de valor 15. Em ambas as redes, cada arco está associado a um rótulo (u_{ij}, x_{ij}) , que indica a sua capacidade e o seu valor de fluxo e os arcos destacados correspondem aos que possuem fluxo positivo.

Figura 35 – Exemplo de rede \mathcal{N} e sua equivalente \mathcal{N}' com fonte e sumidouro únicos.



Fonte: Adaptada de Bang-Jensen e Gutin (2008, p. 133)

A possibilidade de considerar a coexistência de fluxos respeitando algumas propriedades em uma rede dá ainda mais poder de modelagem a essa ferramenta. Em Bang-Jensen e Bessy (2014), o problema de encontrar fluxos arco-disjuntos foi introduzido e estudado. Eles mostraram que esse problema generaliza problemas \mathcal{NP} -Completos bem conhecidos como o problema WEAK-2-LINKAGE e problemas de PARTIÇÃO. Entre outros resultados, eles mostraram que os seguintes problemas são \mathcal{NP} -Completos:

- decidir se uma rede $\mathcal{N} = (D, u \equiv 1)$ admite dois fluxos arco-disjuntos;
- decidir se uma rede $\mathcal{N} = (D, u, b)$, em que todo arco tem capacidade 1 ou 2 e pelo menos um arco tem capacidade 2, admite dois fluxos arco-disjuntos com um vetor de balanço b .

Pelo Lema 5.2, encontrar um fluxo s -ramificado em uma rede \mathcal{N} é um problema fácil, pois é possível reduzir esse problema para o de encontrar um (s, t) -fluxo de valor $n - 1$. Em Bang-Jensen e Bessy (2014), foi mostrado que é possível encontrar, em tempo polinomial, k fluxos s -ramificados arco-disjuntos em uma rede $\mathcal{N} = (D, u \equiv 1)$.

Teorema 5.3 (Bang-Jensen e Bessy (2014)). *Dados uma rede $\mathcal{N} = (D, u \equiv 1, b)$ e um inteiro k . \mathcal{N} admite k fluxos arco-disjuntos, cada um com vetor de balanço b , se e somente se \mathcal{N} admite um fluxo x com vetor de balanço $b_x \equiv kb$. É possível decidir a existência desses fluxos em tempo polinomial.*

Em Bang-Jensen e Bessy (2014), os autores forneceram, com o Teorema 5.4, uma caracterização de redes que admitem k fluxos s -ramificados arco-disjuntos, como uma extensão da caracterização de Edmonds (1973) para s -ramificações.

Teorema 5.4 (Bang-Jensen e Bessy (2014)). *Uma rede $\mathcal{N} = (D, u \equiv n - 1)$, com $s \in V(D)$, admite k fluxos s -ramificados arco-disjuntos se e somente se existem k (s, v) -caminhos arco-disjuntos em D , para todo $v \in V(D) \setminus \{s\}$. Além disso, existe um algoritmo polinomial para construir tais fluxos, caso eles existam.*

Pelo Teorema 5.4, uma rede $\mathcal{N} = (D, u \equiv n - 1)$ admite k fluxos s -ramificados arco-disjuntos se e somente se o digrafo D contém k s -ramificações arco-disjuntas.

Agora, discutimos como a tratabilidade desse problema, de um modo geral, depende da escolha da função de capacidade.

Em Bang-Jensen e Bessy (2014), foi mostrado que o problema de decidir se uma rede admite k fluxos s -ramificados arco-disjuntos é \mathcal{NP} -Completo se todo arco tem capacidade no máximo 2. Em Bang-Jensen, Haved e Yeo (2016), esse resultado foi estendido para redes com capacidade no máximo ℓ , para todo $\ell \geq 2$ fixo. Para a maioria das escolhas de grandes capacidades, o problema permanece difícil.

Segundo Bang-Jensen e Bessy (2014), a menos que a *Hipótese Tempo Exponencial* (ver Apêndice A) falhe, não existe algoritmo polinomial para o problema de encontrar k fluxos s -ramificados arco-disjuntos em uma rede $\mathcal{N} = (D, u \equiv \lambda)$, para $n/2 \leq \lambda \leq n - (\log n)^{1+\varepsilon}$, $\varepsilon > 0$, mesmo se D for acíclico. Em Costa et al. (2019), os autores adaptaram esse último resultado para mostrar que o mesmo vale para $(\log n)^{1+\varepsilon} \leq \lambda \leq n/2$. De acordo com Bang-Jensen e Bessy (2014), o problema pode ser resolvido em tempo polinomial quando $\lambda = n - c$, para valores fixos de k e de $c \geq 1$. Eles mostraram que o problema é XP para k e c fixos, quando o objetivo é encontrar k fluxos s -ramificados arco-disjuntos. Isso foi melhorado para um algoritmo FPT em Bessy et al. (2021).

Em suma, os resultados apresentados no parágrafo anterior dizem grosso modo que o problema de decidir se uma dada rede $\mathcal{N} = (D, u \equiv n - c)$ admite k fluxos s -ramificados arco-disjuntos pode ser resolvido em tempo polinomial para valores fixos de $c \geq 1$ e é improvável que tais algoritmos existam para a maioria das outras escolhas de u . Assim, qualquer caracterização de redes que admitam tais fluxos deve ser difícil de verificar. A seguir, estendemos uma propriedade proposta em Costa et al. (2019).

Seja um digrafo $D = (V, A)$, com $s \in V$, e dois inteiros não negativos k e λ . Dizemos que D , e também uma rede $\mathcal{N} = (D, u \equiv \lambda)$, é (k, λ, s) -suficiente se, para todo $X \subseteq V(D) \setminus \{s\}$, com $X \neq \emptyset$, vale a desigualdade:

$$d_D^-(X) \geq k \cdot \left\lceil \frac{|X|}{\lambda} \right\rceil \quad (5.2)$$

Não é difícil ver o relacionamento entre essa propriedade e a do enunciado do Teorema 5.1. Se $\lambda = n - 1$, então todo arco de D tem capacidade suficiente para enviar tantas

unidades de fluxo quantas forem necessárias para alcançar todo conjunto $X \subseteq V(D) \setminus \{s\}$. Nesse caso, as capacidades não são um problema e a desigualdade (5.2) diz que $d_D^-(X) \geq k$, como no Teorema 5.1, uma vez que $|X| \leq n - 1$. Para outras escolhas de λ , note que todo fluxo s -ramificado em \mathcal{N} deve alcançar um conjunto de vértices X com pelo menos $|X|$ unidades. Assim, cada fluxo deve usar pelo menos $\lceil |X|/\lambda \rceil$ arcos para cobrir X . Em Costa et al. (2019), foi mostrado que a condição da Conjectura 5.6 é necessária para a existência de k fluxos s -ramificados arco-disjuntos em uma rede \mathcal{N} .

Proposição 5.5 (Costa et al. (2019)). *Seja $\mathcal{N} = (D, u \equiv \lambda)$ uma rede e $s \in V(D)$. Se \mathcal{N} admite k fluxos s -ramificados arco-disjuntos, então D é (k, λ, s) -suficiente.*

Em Costa et al. (2019), os autores apresentaram a seguinte conjectura:

Conjectura 5.6 (Costa et al. (2019)). *Seja $\mathcal{N} = (D, u \equiv \lambda)$ uma rede com n vértices e um vértice especial s . Então, para todo $1 \leq \lambda \leq n - 1$, \mathcal{N} admite k fluxos s -ramificados arco-disjuntos se e somente se \mathcal{N} é (k, λ, s) -suficiente.*

Como visto na Seção 2.2, um digrafo $D = (V, A)$ é um multicaminho se D é um caminho quando ignoramos seus arcos múltiplos (com mesma cabeça e mesma cauda). Em Costa et al. (2019), os autores mostraram que a Conjectura 5.6 é válida para qualquer valor de λ , se o digrafo D sobre o qual a rede está definida for um multicaminho.

Teorema 5.7 (Costa et al. (2019)). *Seja $\mathcal{N} = (D, u \equiv \lambda)$ um rede (k, λ, s) -suficiente, em que $D = (V, A)$ é um multicaminho. Então, para $1 \leq \lambda \leq n - 1$, \mathcal{N} admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Considere uma ordenação dos elementos de $V = \langle v_1, v_2, \dots, v_n \rangle$, com $s = v_1$, de modo que todo arco de A seja de v_i para v_{i+1} , com $1 \leq i < n$. Vamos mostrar como construir os k fluxos s -ramificados x^1, \dots, x^k em \mathcal{N} . Com exceção de v_n , todo vértice de D possui exatamente um vizinho de saída. Logo, para todo $2 \leq i \leq n - 1$, v_i deve receber $n - i + 1$ unidades de fluxo de v_{i-1} e deve enviar $n - i$ unidades para v_{i+1} .

Seja $\mu(v_i, v_{i+1})$ a quantidade de arcos de v_i para v_{i+1} . Observe que, para todo $1 \leq i < n$, temos $\mu(v_i, v_{i+1}) \geq k\lceil(n - i)/\lambda\rceil$. Desse modo, para $1 \leq i < n$, sepamos $\lceil(n - i)/\lambda\rceil$ arcos distintos de v_i a v_{i+1} para cada fluxo x^j , com $1 \leq j \leq k$, enviamos λ unidades de fluxo em $\lfloor(n - 1)/\lambda\rfloor$ desses arcos e, se for o caso, enviamos $(n - 1) \bmod \lambda$ unidades pelo arco sobressalente. \square

No próximo capítulo, será apresentada a nossa contribuição em relação a esse assunto, com um estudo mais detalhado sobre a Conjectura 5.6.

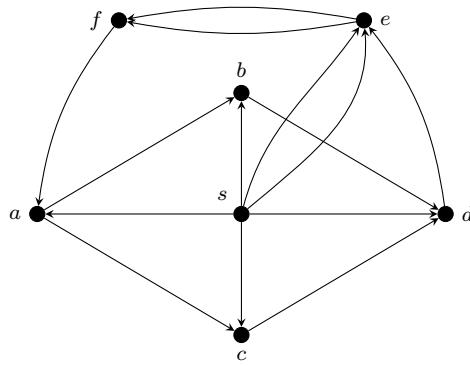
6 FLUXOS RAMIFICADOS ARCO-DISJUNTOS EM REDES (k, λ, s) -SUFICIENTES

Neste capítulo, apresentamos os nossos resultados referentes à Conjectura 5.6. Nós mostramos nas Seções 6.1 e 6.2 que redes $\mathcal{N} = (D, u)$ que são (k, λ, s) -suficientes podem ou não admitir k fluxos s -ramificados arco-disjuntos. Na Seção 6.3 mostramos que é difícil decidir se uma rede (k, λ, s) -suficiente admite tais fluxos.

6.1 Casos Negativos

Em Carvalho et al. (2020), mostramos um contraexemplo para a Conjectura 5.6, ilustrado na rede da Figura 36, considerando $k = \lambda = 2$.

Figura 36 – Contraexemplo para a Conjectura 5.6, para $k = \lambda = 2$.



Fonte: Elaborada pelo autor.

Teorema 6.1 (Carvalho et al. (2020)). *Considere a rede $\mathcal{N} = (D, u \equiv 2)$ da Figura 36. Ela é $(2, 2, s)$ -suficiente, no entanto não admite 2 fluxos s -ramificados arco-disjuntos.*

Demonstração. Inicialmente, vamos mostrar que \mathcal{N} é $(2, 2, s)$ -suficiente, ou seja, para todo $X \subseteq V(D) \setminus \{s\}$, $d_D^-(X) \geq 2\lceil |X|/2 \rceil$. Nesse caso, $d_D^-(X) \leq |X| + 1$. Observe que $d_D^+(v) \geq 1$, $\forall v \in V(D)$, $N_D^+(s) = \{a, b, c, d, e\}$ e $d_D^-(s) = 0$. Denotamos por $e(s, X)$ o número de arcos de s para X . Para $X = V(D) \setminus \{s\}$, temos $d_D^-(X) = 6 \geq 2\lceil |x|/2 \rceil$. Considere que $X \subsetneq V(D) \setminus \{s\}$. Devemos examinar três casos:

- (i). Se $f \notin X$, então, como $d_D^+(f) = 1$ e $e(s, X) \geq |X|$, temos $d_D^-(X) \geq |X| + 1$.
- (ii). Se $f \in X$ e $e \in X$, então, como há dois arcos de s para e e $\{a, b, c, d\} \setminus X \neq \emptyset$, temos $d_D^-(X) \geq |X| + 1$.
- (iii). Se $f \in X$ e $e \notin X$, então, como há dois arcos de e para f , temos $d_D^-(X) \geq e(s, X) + 2 \geq |X| - 1 + 2 = |X| + 1$.

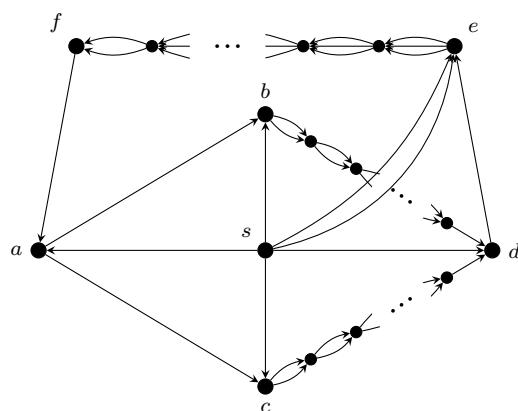
Agora, mostraremos que \mathcal{N} não admite dois fluxos s -ramificados arco-disjuntos x_1 e x_2 . Por contradição, assuma que \mathcal{N} admite tais fluxos. Uma vez que há somente dois arcos entrando em a , um deles deve ser usado pelo fluxo x_1 e o outro pelo fluxo x_2 . Isso também vale para os vértices b, c e f . Sem perda de generalidade, assuma que o arco fa é usado por x_1 , e que o arco sa é usado por x_2 . Se $x_1(fa) = 1$, então, o fluxo x_1 deve usar os arcos sb e sc . Consequentemente, ou $x_2^-(b) = 0$ ou $x_2^-(c) = 0$, o que é uma contradição, pois cada fluxo deve alcançar todos os vértices em $V(D) \setminus \{s\}$. Logo, concluímos que $x_1(fa) = 2$. Isso implica que $x_1^-(f) = x_1^+(f) + 1 = 3$. Como cada arco tem capacidade 2, os dois arcos de e para f devem ser usados por x_1 , o que é uma contradição com o fato de que cada fluxo deve usar um dos arcos que entra em f . \square

A partir da rede da Figura 36, nós mostramos em Carvalho et al. (2023b) como construir uma família de redes (k, λ, s) -suficientes, para qualquer constante $\lambda \geq 2$ e qualquer inteiro par $k \geq 2$, que não admitem k fluxos s -ramificados arco-disjuntos.

Teorema 6.2 (Carvalho et al. (2023b)). *Para toda constante $\lambda \geq 2$ e para todo inteiro par $k \geq 2$, existe uma rede $\mathcal{N} = (D, u \equiv \lambda)$ que é (k, λ, s) -suficiente e não admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Dados um inteiro par $k \geq 2$ e uma constante $\lambda \geq 2$, iniciamos a construção da rede \mathcal{N} adicionando os vértices s, a, b, c e d , com um arco de s para cada um dos outros vértices, e arcos ab, ac, bd e cd . Em seguida, subdividimos os arcos bd e cd $\lambda - 2$ vezes e seja B (resp. C) a união de $\{b\}$ (resp. $\{c\}$) e o conjunto de vértices obtidos pela subdivisão do arco bd (resp. cd) (note que se $\lambda = 2$, nada muda nesse passo, pois $B = \{b\}$ e $C = \{c\}$), e em seguida sobre os arcos entre os vértices de B (resp. C). Seja P um (e, f) -caminho com λ vértices. Triplicamos os primeiros $\lambda - 2$ arcos de P e duplicamos o último (com cabeça f). Feito isso, adicione P a D juntamente com os arcos de, fa e dois arcos paralelos de s para e . Para completar a construção, substitua cada arco por $k/2$ cópias do próprio e defina $u \equiv \lambda$. A rede, que tem $3\lambda + 1$ vértices, está ilustrada na Figura 37.

Figura 37 – Contraexemplo para $k \geq 2$ par e uma constante $\lambda \geq 2$.



Fonte: Carvalho et al. (2023b)

Aqui, dizemos que um conjunto $X \subseteq V(D) \setminus \{s\}$ é *bom* se $d_D^-(X) \geq k \lceil |X|/\lambda \rceil$. Para mostrar que D é (k, λ, s) -suficiente, vamos considerar o caso quando $k = 2$, pois para valores maiores de k nós multiplicamos o grau de entrada de cada subconjunto de $V(D) \setminus \{s\}$ por $k/2$. Observe que, com exceção de s , todo vértice de D tem grau de entrada pelo menos 2 e que, para todo $X \subseteq V(D) \setminus \{s\}$, cada vértice de $\{a, b, c, d, e\} \cap X$ soma pelo menos uma unidade do seu grau de entrada (o vértice e soma pelo menos duas).

Seja $X \subseteq V(D) \setminus \{s\}$. Se $D[X]$ tem um ciclo, então $d_D^-(X) \geq 5$, uma vez que X contém necessariamente $a, d, V(P)$ e B ou C , e $|X| \geq 2\lambda + 1$, o que significa que deve haver pelo menos um arco extra entrando em X para que ele seja um conjunto bom. Para verificar a existência desse arco extra, assuma, sem perda de generalidade, que $B \subset X$ (o caso em que $C \subset X$ é simétrico) e tome o (z, d) -caminho mais longo em $D[X]$, para $z \in C$. Se tal caminho não existe, então há pelo um arco saindo do conjunto C para d que não foi contado antes. Caso contrário, se o caminho existe, então ou $z = c$ ou os dois arcos de y para z entram em X , onde y é o vizinho de entrada de z . Pela escolha do vértice z , garantimos que y não está em X .

Agora, considere que $D[X]$ é acíclico. Note que cada fonte em $D[X]$ contribui com pelo menos duas unidades de fluxo no grau de entrada de X . Uma vez que cada componente de $D[X]$ tem pelo menos uma fonte, então todo conjunto X com cardinalidade no máximo λ é bom. Similarmente, o mesmo vale quando $\lambda < |X| \leq 2\lambda$ (o grau de entrada de X deve ser pelo menos 4) se $D[X]$ tem mais de uma componente ou apenas uma componente com alguns vértices do tipo fonte. Nesse caso, suponha que $D[X]$ tem somente uma componente W com uma única fonte w . Devemos considerar 5 possibilidades:

- (i). $w \in B$. Nesse caso, $\{d, e\} \subset X$, uma vez que $|X| \geq \lambda + 1$ e $|B| = \lambda - 1$, o que significa que $d_D^-(X) \geq 5$.
- (ii). $w \in C$. Situação análoga à anterior.
- (iii). $w = a$. Nesse caso, $|\{b, c, d\} \cap X| \geq 2$. Logo, $d_D^-(X) \geq 4$.
- (iv). $w = d$. Nesse caso, $e \in X$ e, portanto, $d_D^-(X) \geq 5$.
- (v). $w \in V(P)$. Com exceção de f , todo vértice de P tem grau de entrada 3 e, como a deve estar em X , temos que $d_D^-(X) \geq 4$. Quando $w = f$, além de a , X deve conter pelo menos um vértice de $\{b, c\}$, e com isso temos novamente que $d_D^-(X) \geq 4$.

Quando $2\lambda + 1 \leq |X| \leq 3\lambda$, o grau de entrada de X deve ser pelo menos 6. Iniciamos com o caso em que $D[X]$ tem apenas uma componente. Se w uma fonte em $D[X]$, então devemos considerar novamente 5 possibilidades:

- (i). $w \in B$. Nesse caso, como $|B \cup \{d\}| = \lambda$, ou há um vértice em C que também é fonte em $D[X]$ ou $a \in X$. Em ambos os casos, $d, e \in X$ pois $|B \cup C| < 2\lambda$. Assim, $d_D^-(X) \geq 6$.

(ii). $w \in C$. Situação análoga à anterior.

(iii). $w = a$. Como $|\{a, d\} \cup B \cup C| = 2\lambda$, se a é uma fonte, então há um caminho em $D[X]$ que inicia em a , passa por B ou por C , em seguida por d e termina em algum vértice de P . O grau de entrada de a e de e contribui com duas unidades, cada, para o grau de entrada de X . O grau de entrada de d contribui com pelo menos uma unidade. Como $\{b, c\} \cap X \neq \emptyset$, temos que $d_D^-(X) \geq 6$.

(iv). $w = d$. Nesse caso, temos que $\{a, b, c, d, e\} \subset X$. Logo, $d_D^-(X) \geq 6$.

(v). $w \in V(P)$. Aqui, no sentido de evitar ciclos, sabemos que se $w = f$, então w contribui com duas unidades para o grau de entrada de X e $\{a, b, c, d\} \subset X$; e se $w \neq f$, então w contribui com três unidades para o grau de entrada de X e $\{a, b, c\} \subset X$. Assim, $d_D^-(X) \geq 6$.

Se $D[X]$ possui mais de duas componentes, sabemos que $d_D^-(X) \geq 6$. Se $D[X]$ tem duas componentes Y, Z , então uma delas, digamos Y , tem pelo menos $\lambda + 1$ vértices e, portanto, como nos casos anteriores, $d_D^-(Y) \geq 4$. Logo, $d_D^-(X) \geq 6$.

Note que se há k fluxos s -ramificados arco-disjuntos x^1, x^2, \dots, x^k em \mathcal{N} , como $d_D^-(B) = d_D^-(C) = k$, cada um desses k arcos deve ser usado por um fluxo diferente e levar pelo menos $\lambda - 1$ unidades de fluxo, ou seja, uma unidade de fluxo para cada vértice de B e de C . Agora, observe que $k/2$ dos arcos que entram em B vêm de a e o mesmo acontece com os que entram em C . Portanto, a deve receber λ unidades de fluxo em cada um dos k que entram nele, para que possa enviar a quantidade apropriada de fluxo para B e para C . Assuma, sem perda de generalidade, que $k/2$ cópias do arco sa estão sendo usadas pelos fluxos $x^1, x^2, \dots, x^{k/2}$ e que $k/2$ cópias do arco fa são destinadas aos fluxos $x^{k/2+1}, x^{k/2+2}, \dots, x^k$. Como o grau de entrada de f também é igual a k , ele pode receber no máximo λ unidades de fluxo de cada x^1, x^2, \dots, x^k . Daí, ele pode enviar no máximo $\lambda - 1$ unidades de qualquer um desses fluxos para a . Portanto, \mathcal{N} não admite os k fluxos s -ramificados arco-disjuntos. \square

Em Carvalho et al. (2023b) mostramos que, para toda constante inteira $c > 2$, é possível construir uma rede \mathcal{N} que é $(2, n - c, s)$ -suficiente e, no entanto, não admite 2 fluxos s -ramificados arco-disjuntos (Teorema 6.3). Assim, o melhor que se pode esperar é que toda rede $(2, n - 2, s)$ -suficiente admita tais fluxos. Na Seção 6.2, será mostrado (conforme Teorema 6.10) que isso é verdade para redes em geral.

Teorema 6.3 (Carvalho et al. (2023b)). *Para toda constante inteira $c > 2$, existe uma rede \mathcal{N} que é $(2, n - c, s)$ -suficiente e não admite 2 fluxos s -ramificados arco-disjuntos.*

Neste trabalho, generalizamos o resultado anterior para quaisquer constantes $k > 3$ e $c > 2$, o que pode ser visto com o Teorema 6.4.

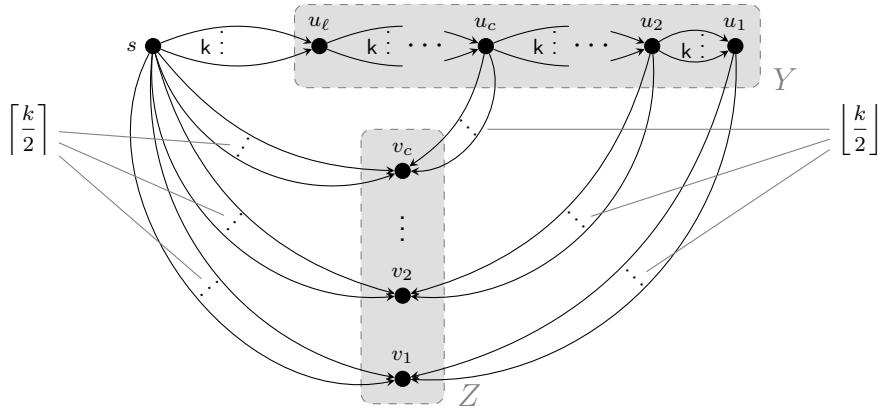
Teorema 6.4. *Para quaisquer constantes inteiras $c > 2$ e $k > 3$, existe uma rede \mathcal{N} que é $(k, n - c, s)$ -suficiente e não admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Dados dois inteiros $k > 3$ e $c > 2$, iniciamos com a construção da rede $\mathcal{N} = (D, u)$ definindo o digrafo $D = (V, A)$ da seguinte maneira:

- $V = \{s, u_1, u_2, \dots, u_c, \dots, u_\ell, v_1, v_2, \dots, v_c\}$;
- adicionamos k arcos paralelos de s a u_ℓ , e de u_i a u_{i-1} , para $\ell \geq i > 1$;
- adicionamos $\lceil k/2 \rceil$ arcos de s para v_i , com $1 \leq i \leq c$;
- adicionamos $\lfloor k/2 \rfloor$ arcos de u_i para v_i , com $1 \leq i \leq c$.

Para completar a definição de \mathcal{N} , definimos $u(a) = n - c$, para todo $a \in A$. Considere os conjuntos $Y = \{u_1, \dots, u_\ell\}$ e $Z = \{v_1, \dots, v_c\}$. Note que $|Y| = n - c - 1$, $|Z| = c$ e $d_D^-(v) = k$, $\forall v \in V(D) \setminus \{s\}$. Essa construção está ilustrada na Figura 38.

Figura 38 – Contraexemplo para $k > 3$, $c > 2$ e $\lambda = n - c$.



Fonte: Elaborada pelo autor.

Vamos mostrar que \mathcal{N} é $(k, n - c, s)$ -suficiente. Para qualquer conjunto não vazio $X \subseteq V(D) \setminus \{s\}$, seja $y = |X \cap Y|$ e $z = |X \cap Z|$. Devemos considerar dois casos:

- (i). Se $|X| \leq n - c$, então devemos mostrar que $d_D^-(X) \geq k$. Se $y > 0$, então há k arcos entrando em $X \cap Y$, e temos $d_D^-(X) = k + z \cdot \lceil k/2 \rceil \geq k$. Caso contrário, temos que $y > 0$ e, novamente, $d_D^-(X) = z(\lceil k/2 \rceil + \lfloor k/2 \rfloor) = z \cdot k \geq k$.
- (ii). Se $|X| > n - c$, então devemos mostrar que $d_D^-(X) \geq 2k$. Nesse caso, $y > 0$ e $z \geq 2$, pois $|Y| = n - c - 1$. Logo, $d_D^-(X) = k + z \cdot \lceil k/2 \rceil \geq 2k$.

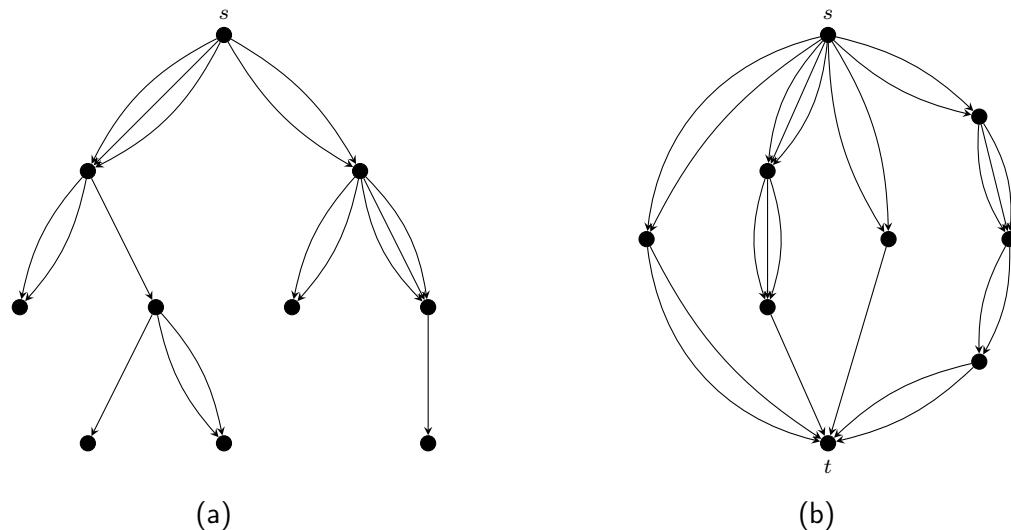
Agora, vamos mostrar que \mathcal{N} não admite k fluxos s -ramificados arco-disjuntos. Note que são necessários $k \cdot c$ arcos com fluxo positivo chegando a Z , pois cada um dos k fluxos deve alcançar cada um dos c vértices em Z . Assuma que há essa quantidade de arcos com fluxo positivo. Observe que cada fluxo pode usar no máximo um arco de Y para Z , pelo tamanho de Y e pela capacidade dos arcos. Assim, há no máximo k arcos com fluxo positivo de Y para Z . Veja também que há no máximo $c \cdot \lceil k/2 \rceil$ arcos com fluxo positivo de s para Z . Como $k > 3$ e $c > 2$, o total de arcos com fluxo positivo é $c \cdot \lceil k/2 \rceil + k < k \cdot c$, uma contradição com o número mínimo de arcos necessários. \square

6.2 Casos Positivos

Em Costa et al. (2019), os autores mostraram que uma rede $\mathcal{N} = (D, u \equiv \lambda)$, em que D é um multicaminho (k, λ, s) -suficiente, admite k fluxos s -ramificados arco-disjuntos. Nesta seção, nós mostramos outros resultados positivos considerando redes (k, λ, s) -suficientes. Destacamos que as provas construtivas destes resultados podem naturalmente ser adaptadas para obter algoritmos de tempo polinomial para encontrar os fluxos desejados.

Como foi visto ao final da Seção 2.2, um digrafo $D = (V, A)$ é uma multiarborescência se D é uma arborescência quando ignoramos seus arcos múltiplos. Para $\ell \geq 1$, denotamos por \mathcal{D}_ℓ a classe de *multispindles*, isto é, a classe de digrafos com um vértice fonte s , um sumidouro t , e ℓ multicaminhos de s a t internamente disjuntos em vértices P_1, \dots, P_ℓ , cada um deles com $p_i \geq 1$ vértices internos, para $1 \leq i \leq \ell$, respectivamente. A Figura 39 traz exemplos de multiarborescência, em (a), e de *multispindle*, em (b).

Figura 39 – Exemplos de multiarborescência e multispindle.



Fonte: Carvalho et al. (2023b)

O primeiro caso positivo foi mostrado por Costa et al. (2019), quando a rede é definida sobre um multicaminho. A seguir, apresentamos uma generalização desse resultado, referente a redes definidas sobre multiarborescências. O Lema 6.5 é útil para demonstrar este resultado.

Lema 6.5 (Carvalho et al. (2023b)). *Considere um digrafo D qualquer que seja (k, λ, s) -suficiente e um subconjunto W não vazio de $V(D) \setminus \{s\}$ tal modo que $N_D^+(V(D) \setminus W) = \{w\}$, $w \in W$. O digrafo $D[W]$ é (k, λ, w) -suficiente.*

Demonstração. Como w é o único vértice de W que tem vizinhos de entrada em $V(D) \setminus W$, temos que $d_{D[W]}^-(v) = d_D^-(v)$ para todo $v \in W \setminus \{w\}$. Logo, para todo $X \subseteq W \setminus \{w\}$, temos que $d_{D[W]}^-(X) = d_D^-(X)$. \square

Teorema 6.6 (Carvalho et al. (2023b)). *Seja $\mathcal{N} = (D, u \equiv \lambda)$ uma rede (k, λ, s) -suficiente em que D é uma multiarborescência. Então, \mathcal{N} admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Por indução na altura h da multiarborescência com raiz s , vamos construir k fluxos s -ramificados arco-disjuntos x^1, \dots, x^k em \mathcal{N} .

O caso base é $h = 1$. Nesse caso, o digrafo D é uma estrela e, uma vez que ele é (k, λ, s) -suficiente, há pelo menos k arcos de s para cada um dos outros vértices. Assim, usamos um arco para cada fluxo x^j , para $1 \leq j \leq k$.

Suponha que o resultado vale para multiarborescências de altura $h < q$. Assuma que D tenha altura igual a q , e seja $N_D^+(s) = \{r_1, r_2, \dots, r_p\}$. Seja B_{r_i} o subgrafo de D que é uma multiarborescência com raiz r_i , para $1 \leq i \leq p$. Note que, pelo Lema 6.5, B_{r_i} é (k, λ, s) -suficiente e tem altura $h_i \leq q - 1$. Pela hipótese da indução, B_{r_i} tem k fluxos r_i -ramificados arco-disjuntos x_1^i, \dots, x_k^i . Como, $d_D^-(r_i) \geq k \lceil |B_{r_i}|/\lambda \rceil$, nós podemos usar $\lceil |B_{r_i}|/\lambda \rceil$ desses arcos para enviar a quantidade adequada de fluxo de s para r_i em cada fluxo x^j para completar os k fluxos s -ramificados arco-disjuntos. \square

O resultado a seguir trata de outro caso positivo, quando a rede é definida sobre um *multispindle*.

Teorema 6.7 (Carvalho et al. (2023b)). *Seja $\mathcal{N} = (D, u \equiv \lambda)$ uma rede (k, λ, s) -suficiente em que D é um multispindle. Então, \mathcal{N} admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Por definição, D é composto de multicaminhos P_1, P_2, \dots, P_ℓ e $V(P_i) \cap V(P_j) = \{s, t\}$ para todo $1 \leq i < j \leq \ell$. Denotamos os p_i vértices internos de P_i por $v_{p_i}^i, v_{p_i-1}^i, \dots, v_2^i, v_1^i$, nessa ordem, para $i \in \{1, \dots, \ell\}$. Para cada $j \in \{1, \dots, p_i\}$, nós definimos $r_j^i \geq 0$ de modo que $d_D^-(v_j^i) = k \lceil j/\lambda \rceil + r_j^i$. Note que $d_D^-(v_j^i) \geq k \lceil j/\lambda \rceil$. Caso contrário, o conjunto $\{v_j^i, v_{j-1}^i, \dots, v_1^i\}$ seria uma contradição com o fato de D ser (k, λ, s) -suficiente, uma vez que os únicos arcos entrando nele são aqueles com cabeça a v_j^i . Considerando apenas os vértices com índice múltiplo de λ , ou seja, os vértices $v_{j\lambda}^i$, para todo $1 \leq j \leq \lfloor p_i/\lambda \rfloor$, definimos:

$$r(P_i) = \begin{cases} k, & \text{if } p_i < \lambda, \\ \min\{r_{j\lambda}^i \mid 1 \leq j \leq \lfloor p_i/\lambda \rfloor\}, & \text{caso contrário.} \end{cases}$$

Para todo $i \in \{1, \dots, \ell\}$, considere e_i o número de arcos de v_1^i para t , $m_i = \min\{r(P_i), e_i\}$, e $k' = \sum_{i=1}^\ell m_i$. Sem perda de generalidade, assuma que P_1, P_2, \dots, P_q são os caminhos nos quais $r(P_i) \leq e_i$, para $i \in \{1, \dots, \ell\}$. Então, $k' = \sum_{i=1}^q r(P_i) + \sum_{i=q+1}^\ell e_i$. Agora, para todo $i \in \{1, \dots, q\}$, escolhemos um $j_i \in [\lfloor p_i/\lambda \rfloor]$ que satisfaz $d_D^-(v_{j_i\lambda}^i) = k j_i + r(P_i)$ e definimos $X_i = \{v_{j_i\lambda}^i, v_{(j_i\lambda)-1}^i, \dots, v_1^i\}$. Seja $X = \bigcup_{i=1}^q X_i \cup \{t\}$. Temos que:

$$d_D^-(X) = \sum_{i=1}^q d_D^-(X_i) + \sum_{i=q+1}^\ell e_i = k \sum_{i=1}^q j_i + \sum_{i=1}^q r(P_i) + \sum_{i=q+1}^\ell e_i = k \sum_{i=1}^q j_i + k'. \quad (6.1)$$

Como D é (k, λ, s) -suficiente e $|X_i| = j_i \lambda$,

$$d_D^-(X) \geq k \left\lceil \frac{|X|}{\lambda} \right\rceil = k \left\lceil \frac{1}{\lambda} + \frac{\lambda \sum_{i=1}^q j_i}{\lambda} \right\rceil = k \sum_{i=1}^q j_i + k. \quad (6.2)$$

Combinando as equações (6.1) e (6.2), concluímos que $k' \geq k$. Para finalmente construir os k fluxos ramificados arco-disjuntos, precisamos do seguinte resultado:

Proposição 6.8. *A rede $\mathcal{N}_i = (P_i, u \equiv \lambda)$ admite k fluxos arco-disjuntos x_1^i, \dots, x_k^i tais que $x_1^i, \dots, x_{m_i}^i$ são fluxos ramificados na rede \mathcal{N}_i e $x_{m_i+1}^i, \dots, x_k^i$ são fluxos ramificados na rede $\mathcal{N}_i \setminus \{t\}$.*

demonstração da Proposição 6.8. Nós mostramos como construir os fluxos $x_1^i, x_2^i, \dots, x_k^i$. Para todo $j \in \{1, \dots, p_i\}$, o vértice v_j^i deve receber $j+1$ unidades de fluxo nos primeiros m_i fluxos e j unidades de fluxo nos $k-m_i$ fluxos restantes. Uma vez que $d_D^-(v_j^i) = d_{P_i}^-(v_j^i) \geq k \lceil j/\lambda \rceil$, nós podemos usar um grupo distinto de $\lceil j/\lambda \rceil$ arcos para enviar j unidades de fluxo em cada um desses k fluxos, e precisamos argumentar como enviar a unidade extra para os fluxos $x_1^i, \dots, x_{m_i}^i$. Se j for um múltiplo de λ , então há pelo menos outros $r(P_i)$ arcos extras entrando em v_j^i e, como $m_i \leq r(P_i)$, eles podem ser usados para enviar unidades extras de fluxo (note que, se $r(P_i) = 0$, então $m_i = 0$ e não há necessidade de unidade extra). Caso contrário, j não é múltiplo de λ e, quando enviamos j unidades de fluxo por $\lceil j/\lambda \rceil$ arcos, há um arco que não será usado em sua capacidade máxima e então podemos utilizá-lo para enviar a unidade extra de fluxo. Aplicando esse método iterativamente de $v_{p_i}^i$ para v_1^i , chegamos a v_1^i com 2 unidades nos fluxos $x_1^i, \dots, x_{m_i}^i$. Agora, podemos enviar a unidade extra de cada fluxo para t , pois $m_i \leq e_i$. \square

Para todo $i \in \{1, \dots, \ell\}$, nós calculamos os fluxos $x_1^i, x_2^i, \dots, x_k^i$ como descrito na Proposição 6.8. Observe que k' é o número de fluxos que alcançam t . Se $k' = k$, então nós podemos renomear esses k fluxos de tal forma que cada $x_j = \bigcup_{i=1}^{\ell} x_j^i$ seja um fluxo s -ramificado em \mathcal{N} . Por fim, se $k' > k$, nós tomamos um fluxo que alcança t e o modificamos de modo que ele não alcance mais t , e repetimos esse processo até que haja somente k fluxos que alcançam t . \square

É importante destacar que as provas para multiarborescências (Teorema 6.6) e *multispindles* (Teorema 6.7), juntamente com a Proposição 5.5, além de darem uma completa caracterização dos digrafos nessas classes cujas redes definidas sobre eles admitem k fluxos ramificados arco-disjuntos, elas levam a algoritmos de tempo polinomial para encontrar tais fluxos, uma vez que, nesses casos, é possível verificar em tempo polinomial se a rede é (k, λ, s) -suficiente. Os algoritmos funcionam para todo valor de λ , mesmo aqueles para os quais encontrar os k fluxos ramificados arco-disjuntos é em geral difícil.

Com o resultado a seguir, mostramos que a Conjectura 5.6 vale para os casos em que $k = 1$ ou $\lambda = 1$.

Teorema 6.9 (Carvalho et al. (2023b)). *Seja D um digrafo (k, λ, s) -suficiente. Se $\lambda = 1$ or $k = 1$, então a rede $\mathcal{N} = (D, u \equiv \lambda)$ admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Iniciamos mostrando um resultado que é comum a ambos os casos. Seja b o seguinte vetor de balanço: $b(s) = k(n - 1)$ e $b(v) = -k$, para todo $v \in V(D) \setminus \{s\}$, onde $n = |V(D)|$. Seja \mathcal{N}' a rede obtida a partir de \mathcal{N} e de b , como descrito no Lema 5.2, e seja (S, \bar{S}) um (s', t') -corte mínimo em \mathcal{N}' . Mostramos que $u(S, \bar{S}) = k(n - 1)$ que, pelo Teorema 2.2, implica que existe um (s', t') -fluxo de valor $k(n - 1)$ em \mathcal{N}' . Note que, pela construção de \mathcal{N}' , $u'(s's) = k(n - 1)$, $u'(vt') = k$ para todo vértice $v \in D \setminus \{s\}$ e $u'(a) = \lambda$ para todo arco $a \in A(D)$. É fácil ver que $u(S, \bar{S}) \leq k(n - 1)$, pois $u(\{s'\}, V(D') \setminus \{s'\}) = k(n - 1)$. Logo, devemos assumir que $s \in S$ e definimos $X = \bar{S} \setminus \{t'\}$. Daí temos:

$$u(S, \bar{S}) = u(S, X) + u(S, \{t'\}) = \lambda d_D^-(X) + k(n - |X| - 1). \quad (6.3)$$

Caso $\lambda = 1$. Como D é $(k, 1, s)$ -suficiente, $d_D^-(X) \geq k|X|$ e, pela equação (6.3), temos que $u(S, \bar{S}) \geq k|X| + k(n - |X| - 1) = k(n - 1)$. Pelo Lema 5.2, a rede \mathcal{N} admite um fluxo com vetor de balanço b . O resultado segue do Teorema 5.3, segundo o qual \mathcal{N} admite k fluxos s -ramificados arco-disjuntos se e somente se \mathcal{N} admite um fluxo x com vetor de balanço b .

Caso $k = 1$. Como D é $(1, \lambda, s)$ -suficiente, então $d_D^-(X) \geq (|X|/\lambda)$, e portanto $\lambda d_D^-(X) \geq |X|$. Substituindo isso na equação (6.3), obtemos $u(S, \bar{S}) \geq |X| + n - |X| - 1 = n - 1$. Assim, $u(S, \bar{S}) = k(n - 1)$. Similarmente ao caso anterior, concluímos pelo Lema 5.2 que \mathcal{N} admite um fluxo x com vetor de balanço b . Como $k = 1$, temos que x é um fluxo ramificado e o resultado segue. \square

Com o Teorema 6.10 a seguir, mostramos que redes $(2, n - 2, s)$ -suficientes admitem 2 fluxos s -ramificados arco-disjuntos. Este é um resultado melhorado do que foi mostrado por Bang-Jensen e Bessy (2014) para $k = 2$, segundo o qual a caracterização de Edmonds de digrafos contendo 2 s -ramificações arco-disjuntas garante a existência de 2 fluxos s -ramificados arco-disjuntos.

Teorema 6.10 (Carvalho et al. (2023b)). *Seja $\mathcal{N} = (D, u \equiv n - 2)$ uma rede $(2, n - 2, s)$ -suficiente. Então, \mathcal{N} admite 2 fluxos s -ramificados arco-disjuntos.*

Demonstração. Como $\mathcal{N} = (D, u)$ é uma rede $(2, n - 2, s)$ -suficiente, então, para todo $X \subset V(D) \setminus \{s\}$, temos $d_D^-(X) \geq 2$. Note que, para $X = V(D) \setminus \{s\}$, temos $d_D^-(X) \geq 4$. Pelo Teorema 5.1 (Teorema de Edmonds), D tem duas s -ramificações arco-disjuntas, digamos B^1 e B^2 . Vamos colorir com a cor i os arcos de B^i , para $i \in \{1, 2\}$, e com a cor 3 os arcos de $A(D) \setminus (A(B^1) \cup A(B^2))$. Observe que todo vértice em $V(D) \setminus \{s\}$ é a cabeça de exatamente um arco de B^1 e de B^2 . A ideia é construir dois fluxos s -ramificados x^1 e x^2 , usando arcos de B^1 e de B^2 , respectivamente. Isso será possível, a menos que seja necessário enviar $n - 1$ unidades de fluxo por um único arco, que deve ter cauda s . Nesse caso particular, mostramos como modificar as ramificações para evitar esse problema.

Sejam $N_{B^1}^+(s) = \{r_1^1, \dots, r_\ell^1\}$ e B_1^1, \dots, B_ℓ^1 as sub-ramificações maximais de $B^1 \setminus \{s\}$ com raízes r_1^1, \dots, r_ℓ^1 , respectivamente. Se $\ell > 1$, então podemos construir um fluxo ramificado x^1 em B^1 enviando $|B_i^1|$ unidades de fluxo pelo arco sr_i^1 , para $1 \leq i \leq \ell$. Isso é possível, pois $u \equiv n - 2$. Procedemos de forma análoga para o fluxo x^2 em B^2 .

Se s tem apenas um vizinho de saída r^1 em B^1 e há um arco sv de cor 3, construímos o fluxo x^2 usando os arcos de $A(B^1) \cup \{sv\}$. Para isso, devemos considerar dois casos: se $v \neq r^1$, seja $uv \in A(B^1)$ e seja B_v^1 o subgrafo conexo maximal de $B_1 \setminus \{u\}$ (que é uma arborescência) com raiz v . Então, enviamos $|B_v^1|$ unidades de fluxo pelo arco sv e $n - 1 - |B_v^1|$ unidades de fluxo pelo arco sr^1 . Caso contrário, $v = r^1$ e temos dois arcos de s para v . Então, podemos enviar 1 unidade de fluxo por um desses arcos e $n - 2$ unidades de fluxo pelo outro. Em ambos os casos, para os arcos de cor 1 restantes, é necessário apenas ajustar a quantidade adequada de fluxo. O mesmo vale para B^2 .

Assuma que todo arco em $A(D)$ com cauda s tem cor 1 ou 2, e, sem perda de generalidade, suponha que apenas um destes, seja sr^1 , tem a cor 1. Para $X = V(D) \setminus \{s\}$, temos que $d_D^-(X) \geq 2\lceil(n-1)/(n-2)\rceil = 4$. Logo, há pelo menos 3 arcos com a cor 2 de s para X . Uma vez que B^2 é uma s -ramificação, r^1 aparece em apenas um B_i^2 . Tomemos um r_j^2 de modo que $B_j^2 \neq B_i^2$. Com exceção de sr_j^2 , não há arcos de cor 2 com apenas uma das extremidades em $V(B_j^2)$. Seja E o conjunto de arcos de cor 1 entrando em $V(B_j^2)$ juntamente com todos os arcos de cor 1 entre vértices de $V(B_j^2)$. Atribuímos a cor 4 aos arcos em E , e a cor 1 aos arcos em $A(B_j^2) \cup \{sr_j^2\}$. Feito isso, um vértice $w \neq s$ é cabeça de apenas um arco de cor 1 e outro de cor 4, se ele está em $V(B_j^2)$. Caso contrário, ele é cabeça de apenas um arco de cor 1 e um de cor 2. Assim, os arcos de cor 1 formarão uma s -ramificação B'_1 . Em particular, há dois arcos de cor 1 saindo de s em B'_1 e é possível construir um fluxo s -ramificado x^1 como descrito anteriormente.

Agora, atribuímos a cor 2 aos arcos de E . Novamente, temos a garantia de que todo vértice em B_2^j tem exatamente um arco de cor 2 chegando a ele (os arcos que estavam com a cor 4) e, uma vez que os arcos de cor 2 que não estão em $A(B_2^j) \cup \{sr_2^j\}$ permanecem inalterados, o novo conjunto de todos os arcos com cor 2 formam uma s -ramificação B'_2 . Mais uma vez, temos pelo menos dois arcos saindo de s em B'_2 , e, portanto, também podemos construir um fluxo s -ramificado x^2 . \square

De acordo com Bang-Jensen e Bessy (2014), é possível decidir em tempo polinomial se uma rede $\mathcal{N} = (D, u \equiv n - 2)$ admite 2 fluxos arco-disjuntos. Com o Teorema 6.10 completamos a prova de que uma rede é $(2, n - 2, s)$ -suficiente se e somente se ela admite 2 fluxos s -ramificados arco-disjuntos. Logo, é possível decidir em tempo polinomial se uma rede é $(2, n - 2, s)$ -suficiente.

Por fim, para escolhas particulares de $d_D^-(X)$ e de λ , mostramos que uma simples condição mais forte do que a apresentada na Proposição 5.5 garante a existência dos k fluxos s -ramificados arco-disjuntos em uma dada rede.

Teorema 6.11 (Carvalho et al. (2023b)). *Seja $\mathcal{N} = (D, u \equiv \lambda)$ uma rede, com $\lambda \geq 2$, e p um inteiro tal que $p\lambda \geq n - 1$. Se $d_D^-(X) \geq pk$ para todo $X \subseteq V(D) \setminus \{s\}$, com $X \neq \emptyset$, então \mathcal{N} admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Pelo Teorema 5.1, há pk fluxos s -ramificados arco-disjuntos B_1, \dots, B_{pk} in D . Para $i \in \{1, \dots, k\}$, seja D_i o digrafo com $V(D_i) = V(D)$ e $A(D_i) = A(B_{(i-1)p+1}) \cup \dots \cup A(B_{ip})$, e considere a rede $\mathcal{N}_i = (D_i, u)$.

Agora, para cada conjunto não vazio $X \subseteq V(D) \setminus \{s\}$, nós concluímos que $d_{D_i}^-(X) \geq p$, uma vez que cada B_i , com $i \in \{1, \dots, pk\}$, deve alcançar X pelo menos uma vez. Além disso, pela nossa escolha de p , temos que toda rede \mathcal{N}_i é $(1, \lambda, s)$ -suficiente. Assim, pelo Teorema 6.9, temos que cada \mathcal{N}_i admite um fluxo s -ramificado x_i . Por fim, como todo arco de D aparece em no máximo um digrafo D_i , nós concluímos que os fluxos s -ramificados x^1, \dots, x^k são arco-disjuntos dois a dois. O resultado segue do fato de que cada um desses fluxos pode ser usado para construir um fluxo s -ramificado em \mathcal{N} , copiando arcos e seus respectivos valores de fluxo usados por cada fluxo x^j , com $j \in \{1, \dots, k\}$. \square

6.3 Complexidade

Acreditamos que nem sempre é fácil, computacionalmente falando, verificar se uma rede é (k, λ, s) -suficiente. Além disso, ainda que a rede possua essa propriedade, é difícil decidir se os fluxos desejados existem, mesmo se a rede for definida em um digrafo obtido pela adição de um multicaminho ao sumidouro de um *multisplindle*, como pode ser observado na demonstração do resultado a seguir.

Teorema 6.12 (Carvalho et al. (2023b)). *Dados dois inteiros k e λ , é \mathcal{NP} -Completo decidir se uma rede (k, λ, s) -suficiente $\mathcal{N} = (D, u \equiv \lambda)$ admite k fluxos s -ramificados arco-disjuntos.*

Demonstração. Vamos mostrar uma redução do problema 3-Partição para o problema em questão. Dado um conjunto $S = \{a_1, \dots, a_{3k}\}$, com $\sum_{i=1}^{3k} v(a_i) = k\lambda$ e $\lambda/4 < v(a_i) < \lambda/2$, uma instância do problema 3-Partição¹, construímos uma rede $\mathcal{N} = (D, u \equiv \lambda)$ que é (k, λ, s) -suficiente de modo que \mathcal{N} admite k fluxos s -ramificados arco-disjuntos x^1, \dots, x^k se e somente se é possível partitionar o conjunto S em k subconjuntos com 3 elementos, cada, de modo que a soma dos valores dos elementos de cada conjunto seja igual a λ .

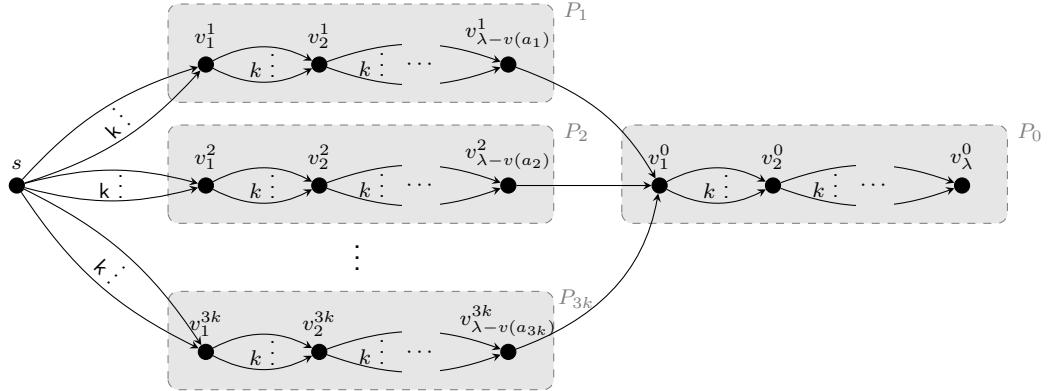
Iniciamos adicionando um vértice s , um multicaminho P_0 com vértices $v_1^0, \dots, v_\lambda^0$ e k arcos paralelos entre vértices consecutivos. Em seguida, para $1 \leq i \leq 3k$, adicionamos:

- um multicaminho P_i com vértices $v_1^i, \dots, v_{\lambda-v(a_i)}^i$ e k arcos paralelos entre vértices consecutivos;
- um arco de $v_{\lambda-a_i}^i$ para v_1^0 ;
- k arcos paralelos de s para v_1^i .

¹ Problema definido na página 36

Finalmente, definimos capacidade $u_a = \lambda$ para todo arco a da rede. Essa construção está ilustrada na Figura 40.

Figura 40 – Rede construída a partir de uma instância do 3-Partição.



Fonte: Carvalho et al. (2023b)

O número de vértices da rede $\mathcal{N} = (D, u)$ é $1 + \lambda + \sum_{i=1}^{3k} (\lambda - v(a_i)) = 1 + \lambda + 3k\lambda - \sum_{i=1}^{3k} v(a_i) = 1 + \lambda + 3k\lambda - k\lambda = (2k+1)\lambda + 1$. Note ainda que, com exceção de v_1^0 , que tem grau de entrada igual a $3k$, todo vértice em $V(D) \setminus \{s\}$ tem grau de entrada igual a k . Assim, o número de arcos da rede é $[(2k+1)\lambda - 1]k + 3k = (2k\lambda + \lambda - 1)k + 3k = 2\lambda k^2 + k(\lambda + 2)$. A construção é, portanto, polinomial no tamanho da entrada.

Primeiro, vamos verificar se a rede $\mathcal{N} = (D, u)$ definida acima é (k, λ, s) -suficiente. Por contradição, assuma que existe $X \subseteq V(D) \setminus \{s\}$ tal que $d_D^-(X) < k \lceil |X|/\lambda \rceil$. Note que $X \neq V(D) \setminus \{s\}$, pois $|V(D) \setminus \{s\}| = (2k+1)\lambda$ e $d_D^-(V(D) \setminus \{s\}) = k(3k)$.

Como $d_D^-(v) = k$, para todo $v \in V(D) \setminus \{s, v_1^0\}$, e $d_D^-(v_1^0) = 3k$, $d_D^-(X) \geq k$. Portanto X não está contido em um P_i , com $0 \leq i \leq 3k$, uma vez que, nesse caso, $|X| \leq \lambda$ e então $\lceil |X|/\lambda \rceil = 1$. Note ainda que $X \neq V(P_0)$, pois $d_D^-(v_1^0) = 3k$. Agora, considere $I = \{i \mid 1 \leq i \leq 3k, V(P_i) \cap X \neq \emptyset\}$. Observe que $d_D^-(X) \geq k|I|$. Consideremos os seguintes casos:

- (i) Se $V(P_0) \cap X = \emptyset$, então $|X| \leq \lambda|I|$ e $\lceil |X|/\lambda \rceil \leq |I|$. Consequentemente, $d_D^-(X) \geq k|I| \geq k \lceil |X|/\lambda \rceil$, uma contradição com a escolha de X .
- (ii) Se $X \cap V(P_0) \neq \emptyset$, então $|X| \leq \lambda|I| + \lambda = \lambda(|I| + 1)$. Tomemos o menor índice j de modo que $v_j^0 \in X$. Devemos considerar duas possibilidades:

(ii.1) $j > 1$. Nesse caso, há k arcos de v_{j-1}^0 a v_j^0 que entram em X . Uma vez que $|I| > 0$ e $|X| \leq \lambda(|I| + 1)$, temos que $d_D^-(X) \geq k|I| + k = k(|I| + 1) \geq k \lceil |X|/\lambda \rceil$, uma contradição.

(ii.2) $j = 1$. Nesse caso, $v_1^0 \in X$ e $d_D^-(X) \geq k|I|$. Devemos considerar três casos:

(ii.2.1) Se há pelo menos k diferentes valores de i para os quais $v_{\lambda-a_i}^i \notin X$, então cada um desses vértices contribuem com k unidades para o grau de entrada de X e, novamente, temos $d_D^-(X) \geq k|I| + k$, uma contradição.

- (ii.2.2) Se há somente $1 \leq l < k$ valores de i para os quais $v_{\lambda-v(a_i)}^i \notin X$, então, para completar o argumento, é necessário estimar mais precisamente a cardinalidade de X . Lembre que $\lambda/4 < v(a_i) < \lambda/2$. Assim, $|X| \leq \lambda|I| - \frac{\lambda|I|}{4} + \lambda = \lambda(\frac{3|I|}{4} + 1)$. Observe que o número de arcos entrando em X é $k|I| + l = \frac{3k|I|}{4} + \frac{k|I|}{4} + l$. Como $|I| > 2k$, temos que $\frac{k|I|}{4} > \frac{k^2}{2} \geq k$, se $k \geq 2$ (que assumimos ser verdade, pela definição do 3-Partição). Logo, $d_D^-(X) \geq \frac{3k|I|}{4} + k = k(\lambda(\frac{3|I|}{4} + 1)/\lambda)$, que é uma contradição.
- (ii.2.3) Se $v_{\lambda-v(a_i)}^i \in X$, para $1 \leq i \leq 3k$, então $d_D^-(X) = d_D^-(V \setminus \{s\}) = k(3k)$, e temos novamente uma contradição.

Como obtivemos uma contradição em cada caso, a rede é (k, λ, s) -suficiente.

Assuma que é possível particionar o conjunto S em k subconjuntos S_1, \dots, S_k tais que $\sum_{a_j \in S_i} v(a_j) = \lambda$, $1 \leq i \leq k$. No sentido de construir os k fluxos s -ramificados arco-disjuntos x^1, \dots, x^k na rede \mathcal{N} , cada uma das k cópias de um arco múltiplo deve ser usado por um fluxo diferente. Para cada $1 \leq i \leq k$ e $1 \leq j \leq 3k$, seja a o arco de s para v_1^j que será utilizado pelo fluxo x^i . Se $a_j \notin S_i$, são enviadas $\lambda - v(a_j)$ unidades de fluxo pelo arco a . Caso contrário, são enviadas λ unidades de fluxo pelo arco a . Uma vez que $\sum_{a_j \in S_i} v(a_j) = \lambda$, os arcos com cauda $v_{\lambda-v(a_j)}^j$, para $a_j \in S_i$, são usados para enviar a quantidade de fluxo necessária para completar o fluxo x^i ao longo de P_0 .

Assuma agora que há k fluxos s -ramificados arco-disjuntos x^1, \dots, x^k na rede \mathcal{N} . Nós afirmamos que $x^r(v_{\lambda-v(a_i)}^i v_1^0) \leq v(a_i)$, para todo $1 \leq r \leq k$, $1 \leq i \leq 3k$. Como os fluxos x^1, \dots, x^k são arco-disjuntos, há apenas um fluxo x^t que é positivo no arco $v_{\lambda-v(a_i)}^i v_1^0$. Se $x^t(v_{\lambda-v(a_i)}^i v_1^0) > v(a_i)$, então, como $|V(P_i)| = \lambda - v(a_i)$, o vértice v_1^i deve receber mais do que λ unidades de fluxo em x^t , ou seja, x^t usa pelo menos dois arcos de s para v_1^i . Isso é uma contradição com o fato de que há somente k arcos de s para v_1^i , e cada um deles deve ser usado por exatamente um fluxo. Cada fluxo x^i deve alcançar o vértice v_1^0 com λ unidades de fluxo. Uma vez que $\lambda/4 < v(a_i) < \lambda/2$, cada fluxo x^r deve usar três arcos do tipo $v_{\lambda-v(a_j)}^j v_1^0$, e os elementos a_j formam um subconjunto S_r . Assim, S_1, \dots, S_k é uma solução para a instância S do problema 3-Partição. \square

Observe que, pela demonstração do Teorema 6.12, as redes construídas a partir de instâncias negativas do problema 3-Partição também são redes (k, λ, s) -suficientes que não admitem k fluxos s -ramificados arco-disjuntos.

7 CONCLUSÃO

Esta tese abordou dois tipos de problemas relacionados a decomposição de fluxos em redes. O primeiro foi o de fluxos divisíveis, uma variação dos problemas clássicos de fluxos em que o produto enviado de uma origem a um destino pode passar por múltiplos caminhos. O objetivo do problema é encontrar uma decomposição de custo mínimo de fluxos em redes arco-coloridas, em que o custo é dado pela soma do custo dos caminhos e o custo de cada caminho é o número de cores distintas deste. O segundo problema é referente à caracterização de redes que admitem k fluxos s -ramificados arco-disjuntos. Estudamos uma condição semelhante à de Edmonds (para s -ramificações), adaptada para fluxos s -ramificados, que chamamos de (k, λ, s) -suficiência.

Esses dois problemas centrais da tese são variantes do problema de decomposição de fluxos em redes. A primeira classe de problemas, em grafos coloridos, foi proposta originalmente nesta tese, motivados pelas aplicações e pelo desafio de lidar com problemas que são fáceis de resolver em redes monocromáticas e difíceis em redes coloridas. Esta tese revelou mais alguns casos como esses. A segunda classe de problemas, de decomposição em ramificações, nos permitiu a possibilidade de ver aplicações e limitações da caracterização de Edmonds, com estudos para a elaboração de requisitos adicionais para a garantia de existência de fluxos s -ramificados arco-disjuntos em redes.

7.1 Resultados obtidos

Em relação ao problema de decomposição de fluxos em redes arco-coloridas, dividimos o nosso estudo considerando o número de cores distintas nos arcos da rede.

Para redes monocromáticas, o problema consiste em minimizar o número de caminhos da decomposição. Segundo Hartman et al. (2012), isso é resolvido em tempo polinomial quando há apenas dois valores distintos de fluxo nos arcos da rede e o menor valor divide o maior, e é \mathcal{NP} -Difícil para redes com três valores distintos de fluxo. Nós mostramos que em redes acíclicas com dois valores quaisquer de fluxo o problema pode ser resolvido em tempo polinomial.

Para redes com duas cores, mostramos o problema pode ser resolvido em tempo polinomial quando o fluxo é λ -uniforme ou quando cada cor está associada a um valor de fluxo e o menor valor divide o maior. Diferentemente da versão monocromática, o problema permanece em aberto para o caso geral em que há dois valores de fluxo.

Para redes com três cores, provamos que o problema é \mathcal{NP} -Difícil mesmo quando o fluxo é uniforme e o grau de cada vértice da rede, com exceção da fonte e do sumidouro, é no máximo 6. Quando o problema é restrito a redes acíclicas com pelo menos 5 cores e fluxos uniformes, ele permanece \mathcal{NP} -Difícil.

Em relação à segunda parte, referente aos fluxos s -ramificados arco-disjuntos, mostramos que em alguns casos a propriedade de (k, λ, s) -suficiência garante a existência de tais fluxos. Em outros casos, uma condição mais forte é necessária. Ainda assim, os casos para os quais a propriedade funciona são interessantes, entre outras razões, pois resultam em algoritmos de tempo polinomial para encontrar tais fluxos. Mostramos também que o problema de decidir se uma rede \mathcal{N} (k, λ, s) -suficiente admite k fluxos s -ramificados arco-disjuntos é \mathcal{NP} -Completo.

7.2 Próximas etapas

Sobre o problema de decomposição de fluxos em redes arco-coloridas, devemos seguir investigando o problema para as situações que relacionam quantidades de cores e quantidade de valores de fluxo nos arcos, correspondentes aos casos em aberto indicados no Quadro 1 (página 65). São elas:

- a) a rede possui exatamente duas cores, sendo que cada cor está associada a um valor de fluxo e o menor valor não divide o maior;
- b) a rede possui exatamente duas cores e dois valores de fluxo, mas não há associação entre cor e valor de fluxo;
- c) fluxos uniformes em redes acíclicas com três ou quatro cores.

Uma outra direção da pesquisa é trabalhar na busca de um algoritmo aproximativo, com um bom fator de aproximação, para o problema em questão.

Devemos também investigar três variações do problema de decomposição de um fluxo x em uma rede arco-colorida em fluxos caminhos x^1, \dots, x^ℓ , em que cada x^i é enviado ao longo de um caminho P_i , originalmente formuladas nesta tese, cujos objetivos são respectivamente:

- a) minimizar $\sum_{i=1}^{\ell} \sum_{j \in \text{cores}(P_i)} \text{span}(j, P_i);$
- b) minimizar $\sum_{i=1}^{\ell} n_c(P_i)^2;$
- c) maximizar $\sum_{i=1}^{\ell} \frac{|x^i|}{n_c(P_i)}.$

Na primeira abordagem, o custo de um caminho é dado pela soma do *span* de cada cor deste. Considere, por exemplo, um sistema de transporte multimodal. Encontrar um fluxo caminho com poucas cores corresponde a fazer uma rota usando poucos tipos de meios de transporte. Adicionalmente, pode não ser interessante ficar alternando entre meios de transporte a todo instante. O *span* de cada cor, no caminho, corresponde à quantidade de vezes que será necessário pegar o meio de transporte correspondente a esta ao longo do caminho.

Na segunda abordagem, buscamos obter uma decomposição em que os números de cores dos caminhos sejam próximos. Pensando nas cores como fatores de risco, os caminhos têm, em geral, aproximadamente a mesma quantidade de riscos. Por exemplo, no problema original, proposto nesta tese, podemos ter duas decomposições de um fluxo em dois fluxos caminhos com um custo 10, uma em que os custos dos caminhos são 1 e 9 e outra em que esses custos são 4 e 6. Nesse caso, a segunda decomposição tem um custo menor, pois $4^2 + 6^2 = 52 < 1^2 + 9^2 = 82$.

Na última abordagem, levamos em conta tanto um aspecto quantitativo (o valor) quanto um qualitativo (as cores) de cada fluxo caminho. Estamos interessados em uma decomposição que passe grandes valores de fluxo por caminhos com poucas cores. Pensando novamente nas cores como fatores de risco, estamos interessados em encontrar rotas mais seguras para enviar grandes quantidades de uma mercadoria.

Em relação aos fluxos s -ramificados arco-disjuntos, embora uma condição necessária tenha sido apresentada por Costa et al. (2019) e esta tese tenha apresentado uma condição suficiente (Teorema 6.11), o problema de encontrar uma condição simultaneamente necessária e suficiente para a existência de tais fluxos permanece em aberto. Além dessa, algumas questões de complexidade ainda estão em aberto, tais como decidir se uma rede $\mathcal{N} = (D, u \equiv \lambda)$ é (k, λ, s) -suficiente e encontrar uma possível dicotomia em *DAGs* entre os casos fáceis e os difíceis. Em outras palavras, seria interessante saber se há uma classe de digrafos acíclicos \mathcal{H} , de modo que toda rede construída sobre um digrafo $D \in \mathcal{H}$ contenha k fluxos s -ramificados arco-disjuntos, mas o mesmo não possa ser observado para digrafos $D \notin \mathcal{H}$.

REFERÊNCIAS

- AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. **Network Flows: Theory, Algorithms, and Applications**. 1. ed. New Jersey: Prentice hall, 1993.
- BAIER, G.; KÖHLER, E.; SKUTELLA, M. The k-splittable flow problem. **Algorithmica**, v. 42, n. 3-4, 2005.
- BANG-JENSEN, J.; BESSY, S. (Arc-)disjoint flows in networks. **Theoretical Computer Science**, v. 526, p. 28–40, 2014.
- BANG-JENSEN, J.; GUTIN, G. Z. **Digraphs: Theory, Algorithms and Applications**. 2. ed. New York: Springer Publishing Company, Incorporated, 2008. (Monographs in Mathematics). ISBN 1848009976.
- BANG-JENSEN, J.; HAVED, F.; YEO, A. The complexity of finding arc-disjoint branching flows. **Discrete Applied Mathematics**, v. 209, p. 16–26, 2016.
- BANG-JENSEN, J.; HAVET, F.; MAIA, A. K. Finding a subdivision of a digraph. **Theoretical Computer Science**, v. 562, p. 283–303, 2015. ISSN 0304-3975.
- BESSY, S.; HÖRSCH, F.; MAIA, A. K.; RAUTENBACH, D.; SAU, I. **FPT algorithms for packing k -safe spanning rooted sub(di)graphs**. 2021.
- BONDY, J. A.; MURTY, U. S. R. **Graph Theory**. 1. ed. New York, NY: Springer, 2008. v. 244. (Graduate Texts in Mathematics, v. 244).
- BORDINI, A.; PROTTI, F.; SILVA, T. G. da; SOUSA FILHO, G. F. New algorithms for the minimum coloring cut problem. **International Transactions in Operational Research**, v. 26, n. 5, p. 1868–1883, 2019. Disponível em: <https://onlinelibrary.wiley.com/doi/abs/10.1111/itor.12494>.
- BROERSMA, H.; LI, X. Spanning trees with many or few colors in edge-colored graphs. **Discuss. Math. Graph Theory**, v. 17, n. 2, p. 259–269, 1997.
- CAMPOS, V.; LOPES, R.; MARINO, A.; SILVA, A. Edge-disjoint branchings in temporal graphs. In: **Proc. of the 31st International Workshop on Combinatorial Algorithms (IWOCA)**. [S.I.: s.n.], 2020. (LNCS, v. 12126), p. 112–115.
- CARVALHO, C.; SILVA, J. C. F. da; MAIA, A. K.; SALES, C. L.; LOPES, R.; NISSE, N. Characterizing networks admitting k arc-disjoint branching flows. In: **Anais do V Encontro de Teoria da Computação**. Porto Alegre, RS, Brasil: SBC, 2020. p. 57–60. ISSN 2595-6116. Disponível em: <https://sol.sbc.org.br/index.php/etc/article/view/11089>.
- CARVALHO, C.; SILVA, J. C. F. da; MAIA, A. K.; SALES, C. L. Decomposição de fluxos em digrafos arco-coloridos. In: **Anais do VIII Encontro de Teoria da Computação**. Porto Alegre, RS, Brasil: SBC, 2023. p. 6–9. ISSN 2595-6116. Disponível em: <https://sol.sbc.org.br/index.php/etc/article/view/24731>.
- CARVALHO, C.; SILVA, J. C. F. da; MAIA, A. K.; SALES, C. L.; LOPES, R.; NISSE, N. From branchings to flows: a study of an edmonds' like property to arc-disjoint branching flows. **Discrete Mathematics & Theoretical Computer Science**, v. 25:1, May 2023. Disponível em: <https://dmtcs.episciences.org/11237>.

- CARVALHO, C.; SILVA, J. C. F. da; MAIA, A. K.; SALES, C. L. Flow decomposition on arc-coloured networks. In: **Annals of the LAWCG2024 - 11th Latin American Workshop on Cliques in Graphs**. SBC, 2024. p. 38. ISSN 2595-6116. Disponível em: <https://www.lawcg.mat.br/lawcg24/LAWCG2024-annals.pdf>.
- CARVALHO, C.; SILVA, J. C. F. da; MAIA, A. K.; SALES, C. L. **Minimum cost flow decomposition on arc-coloured networks [manuscrito em preparação]**. 2025.
- CERULLI, R.; GENTILI, M.; IOSSA, A. Efficient preflow push algorithms. **Comput. Oper. Res.**, Elsevier Science Ltd., GBR, v. 35, n. 8, 2008.
- CHERIYAN, J.; MEHLHORN, K. An analysis of the highest-level selection rule in the preflow-push max-flow algorithm. **Inf. Proc. Letters**, v. 69, n. 5, p. 239–242, 1999.
- COOK, S. A. The complexity of theorem-proving procedures. **Proceedings of the third annual ACM symposium on Theory of computing**, 1971.
- CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Introduction to Algorithms**. 4. ed. London, England: MIT Press, 2022. ISBN 9780262046305.
- COSTA, J.; SALES, C. L.; LOPES, R.; MAIA, A. Um estudo de redes com fluxos ramificados arco-disjuntos. **Matemática Contemporânea**, v. 46, p. 230–238, 2019.
- COUDERT, D.; DATTA, P.; PERENNES, S.; RIVANO, H.; VOGÉ, M.-E. Shared risk resource group: Complexity and approximability issues. **Parallel Processing Letters**, v. 17, n. 2, p. 169–184, 6 2007.
- DINIC, E. A. Algorithm for solution of a problem of maximum flow in networks with power estimation. **Soviet Math. Doklady**, v. 11, p. 1277–1280, 1970.
- EDMONDS, J. Edge-disjoint branchings. **Combinatorial Algorithms**, Academic Press, 1973.
- EDMONDS, J. R.; KARP, R. M. Theoretical improvements in algorithmic efficiency for network flow problems. **J. ACM**, v. 19, n. 2, p. 248–264, 1972.
- FARIA, L.; SUCUPIRA, R.; KLEIN, S.; SAU, I.; SOUZA, U. Sobre cortes em grafos aresta-coloridos. In: **Anais do VLVIII Simpósio Brasileiro de Pesquisa Operacional (SBPO)**. [s.n.], 2016. p. 3164–3175. Disponível em: <https://www.din.uem.br/sbpo/sbpo2016/pdf/156357.pdf>.
- FORD, L. R.; FULKERSON, D. R. **Flows in Networks**. 1. ed. Princeton, NJ: Princeton University Press, 1956.
- FORTUNE, S.; HOPCROFT, J.; WYLLIE, J. The directed subgraph homeomorphism problem. **Theoretical Computer Science**, Elsevier, v. 10, n. 2, p. 111–121, 1980.
- FRANK, A. Covering branchings. **Acta Scientiarum Mathematicarum (Szeged)**, v. 41, p. 77–81, 1979.
- GAREY, M. R.; JOHNSON, D. S. **Computers and Intractability: A Guide to the Theory of NP-Completeness**. 1. ed. [S.I.]: W. H. Freeman, 1979. (Series of Books in the Mathematical Sciences).

- GOLDBERG, A. V.; TARJAN, R. E. A new approach to the maximum-flow problem. **J. ACM**, Association for Computing Machinery, New York, NY, USA, v. 35, n. 4, p. 921–940, 1988.
- GRANATA, D.; CERULLI, R.; SCUTELLÀ, M. G.; RAICONI, A. Maximum flow problems and an NP-Complete variant on edge-labeled graphs. In: PARDALOS, P. M.; DU, D.-Z.; GRAHAM, R. L. (Ed.). **Handbook of Combinatorial Optimization**. New York, NY: Springer New York, 2013. p. 1913–1948. ISBN 978-1-4419-7997-1.
- HARTMAN, T.; HASSIDIM, A.; KAPLAN, H.; RAZ, D.; SEGALOV, M. How to split a flow. In: **IEEE INFOCOM 2012**. [S.I.: s.n.], 2012. p. 828–836. ISBN 9781467307758.
- HAYNES, T. W.; HEDETNIEMI, S.; SLATER, P. **Fundamentals of Domination in Graphs**. 1. ed. New York: Marcel Dekker, 1998. v. 208. (Monographs and textbooks in pure and applied mathematics, v. 208).
- IMPAGLIAZZO, R.; PATURI, R. Complexity of k-sat. In: **Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)**. [S.I.: s.n.], 1999. p. 237–240.
- KAIBEL, V.; PEINHARDT, M. On the bottleneck shortest path problem. **ZIB-Report 06-22, Konrad-Zuse-Zentrum für Informationstechnik Berlin**, 01 2006.
- KARP, R. M. Reducibility among combinatorial problems. In: **Complexity of computer computations (Proc. Sympos., IBM Thomas J. Watson Res. Center, Yorktown Heights**. New York: Plenum, 1972. p. 85–103.
- KARZANOV, A. V. Determining the maximal flow in a network by the method of preflows. In: **Soviet Math. Doklady**. [S.I.: s.n.], 1974. v. 15, p. 434–437.
- KING, V.; RAO, S.; TARJAN, R. A faster deterministic maximum flow algorithm. In: **Proceedings of the Third Annual ACM-SIAM Symposium on Discrete Algorithms**. USA: Society for Industrial and Applied Mathematics, 1992. (SODA '92), p. 157–164. ISBN 089791466X.
- KLEINBERG, J. M. Single-source unsplittable flow. In: **37th Annual Symposium on Foundations of Computer Science, FOCS '96**. Burlington, Vermont: IEEE Computer Society, 1996. p. 68–77.
- LEVIN, L. A. Universal sequential search problems. **Problems of Information Transmission**, v. 9, n. 3, p. 265–266, 1973.
- LOVÁSZ, L. On two minimax theorems in graph. **Journal of Combinatorial Theory, Series B**, v. 21, n. 2, p. 96–103, 1976. ISSN 0095-8956. Disponível em: <http://www.sciencedirect.com/science/article/pii/0095895676900496>.
- MENGER, K. Zur allgemeinen kurventheorie. **Fundamenta Mathematicae**, v. 10, n. 1, p. 96–115, 1927. Disponível em: <http://eudml.org/doc/211191>.
- MORAWIETZ, N.; GRÜTTEMEIER, N.; KOMUSIEWICZ, C.; SOMMER, F. Refined parameterizations for computing colored cuts in edge-colored graphs. In: . [S.I.]: Springer Nature, 2020. p. 248–259.

- ORLIN, J. B. Max flows in $o(nm)$ time, or better. In: **Proceedings of the 45th Annual ACM Symposium on Theory of Computing**. New York, NY: ACM, 2013. p. 765–774.
- PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial Optimization: Algorithms and Complexity**. 1. ed. Englewood Cliffs, NJ: Prentice Hall, 1982.
- QUEYRANNE, M. Theoretical efficiency of the algorithm “capacity” for the maximum flow problem. **Math. Oper. Res.**, v. 5, n. 2, p. 258–266, 1980.
- SCHAEFER, T. J. The complexity of satisfiability problems. In: **Proceedings of the Tenth Annual ACM Symposium on Theory of Computing**. New York, NY, USA: Association for Computing Machinery, 1978. (STOC '78), p. 216–226. ISBN 9781450374378.
- SCHRIJVER, A. **Combinatorial optimization: polyhedra and efficiency**. [S.l.]: Springer Science & Business Media, 2003. v. 24.
- SHILOACH, Y. Edge-disjoint branching in directed multigraphs. **Inf. Process. Lett.**, v. 8, n. 1, p. 24–27, 1979. Disponível em: [https://doi.org/10.1016/0020-0190\(79\)90086-3](https://doi.org/10.1016/0020-0190(79)90086-3).
- VATINLEN, B.; CHAUVET, F.; CHRÉTIENNE, P.; MAHEY, P. Simple bounds and greedy algorithms for decomposing a flow into a minimal set of paths. **European Journal of Operational Research**, v. 185, n. 3, p. 1390–1401, 2008. ISSN 0377-2217.
- WILLIAMSON, D. P. **Network Flow Algorithms**. 1. ed. New York: Cambridge University Press, 2019.
- WILLIAMSON, D. P.; SHMOYS, D. B. **The Design of Approximation Algorithms**. New York, NY: Cambridge University Press, 2011. ISBN 978-0-521-19527-0.
- YUAN, S.; VARMA, S.; JUE, J. P. Minimum-color path problems for reliability in mesh networks. In: **Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies**. [S.l.: s.n.], 2005. v. 4, p. 2658–2669.
- ZWICK, U. The smallest networks on which the Ford-Fulkerson maximum flow procedure may fail to terminate. **Theoretical Computer Science**, v. 148, n. 1, p. 165–170, 1995.

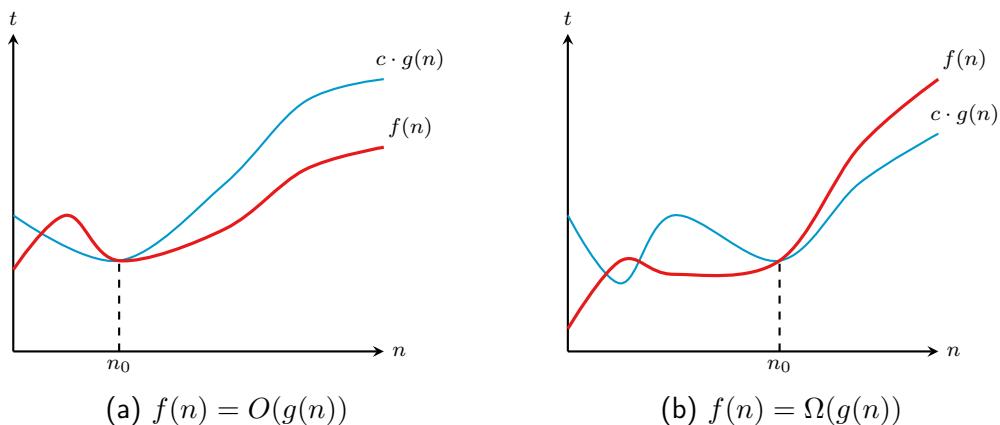
APÊNDICE A – COMPLEXIDADE DE ALGORITMOS

A *Complexidade de Algoritmos* pode ser vista sob dois aspectos – espaço e tempo. A *Complexidade de Espaço* está relacionada à quantidade de memória que é utilizada por um algoritmo, ao receber uma entrada de tamanho n . Já a *Complexidade de Tempo* é uma medida do número de instruções executados pelo algoritmo para uma entrada de tamanho n . Esse número de instruções é normalmente representado por uma função de complexidade $f(n)$, que depende do tamanho e da natureza da entrada.

Notação Assintótica

Dadas duas funções $f(n)$ e $g(n)$, diz-se que $g(n)$ é um limite superior para $f(n)$, denotado por $f(n) = O(g(n))$, se existem constantes positivas c e n_0 tais que $0 \leq f(n) \leq c \cdot g(n)$, para todo $n \geq n_0$. De forma análoga, diz-se que $g(n)$ é um limite inferior para $f(n)$, denotado por $f(n) = \Omega(g(n))$, se existem constantes positivas c e n_0 tais que $0 \leq c \cdot g(n) \leq f(n)$. Se $f(n) = O(g(n))$ e $f(n) = \Omega(g(n))$, então $f(n) = \Theta(g(n))$. Nesse caso, $g(n)$ é um limite restrito para $f(n)$. A Figura A.1 traz exemplos de limites superior e inferior em notação assintótica.

Figura A.1 – Notação Assintótica.



Fonte: Elaborada pelo autor.

Os limites assintóticos definidos pelas notações O e Ω podem não ser “justos”. Por exemplo, o limite $3n = O(n)$ é justo, mas $3n = O(n^3)$ não o é. O mesmo acontece com os limites inferiores $3n = \Omega(n)$ e $3n = \Omega(1)$. Nesse sentido, utilizam-se as notações o e ω para definir limites superior e inferior, respectivamente, que têm folga. Diz-se que $f(n) = o(g(n))$ se existe uma constante positiva n_0 tal que $0 \leq f(n) \leq c \cdot g(n)$, para todo $c > 0$. A notação ω é definida de forma análoga.

Um algoritmo é dito de tempo polinomial se o seu tempo de execução está em $O(n^k)$, para algum inteiro positivo k e para uma entrada de tamanho n .

Problemas de Decisão e de Otimização

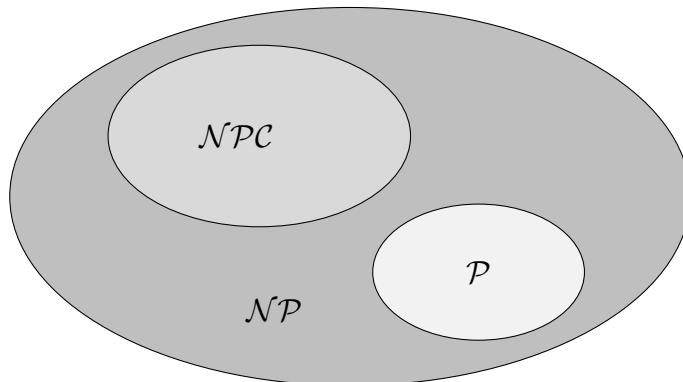
Segundo Cormen et al. (2022), *Problemas de Otimização* são aqueles para os quais há um valor associado a cada solução viável, e se deseja saber qual o melhor valor. *Problemas de Decisão* são aqueles para os quais a resposta é apenas “sim” ou “não”. Por exemplo: Dados um grafo $G = (V, E)$ ponderado e dois vértices $a, b \in V$. Se for perguntado qual o menor caminho de a até b , trata-se de um problema de otimização. Se, por outro lado, for perguntado se existe um caminho de a até b cujo tamanho é no máximo x , trata-se de um problema de decisão.

Há uma relação entre os problemas de otimização e os de decisão, pois é sempre possível converter um problema do primeiro tipo em um do segundo, impondo um limite no valor a ser otimizado. Assim, é possível verificar a dificuldade de um problema de otimização, baseado na complexidade do problema de decisão associado. Isto porque o problema de decisão é visto como, de certo modo, “mais fácil”, ou pelo menos não “mais difícil” do que o de otimização.

Classes de Complexidade de Problemas de Decisão

De acordo com Cormen et al. (2022, p. 1043), os problemas de decisão são agrupados em classes. A Classe \mathcal{P} consiste dos problemas que podem ser solucionados em tempo polinomial. A Classe \mathcal{NP} consiste daqueles que podem ser “verificáveis” em tempo polinomial, ou seja, para uma dada instância cuja resposta é “sim” e um certificado de uma solução, é possível verificar em tempo polinomial se ele está correto. Informalmente, um problema está na classe \mathcal{NPC} , e ele é dito \mathcal{NP} -Completo, se ele é tão difícil de resolver quanto qualquer outro problema em \mathcal{NP} . Estes são problemas para os quais não se conhecem ainda algoritmos que os resolva em tempo polinomial, nem há sequer prova de que tais algoritmos possam existir. Esta é uma das razões pelas quais a maioria dos pesquisadores acredita que $\mathcal{P} \neq \mathcal{NP}$. Essas classes estão ilustradas na Figura A.2.

Figura A.2 – Classes de complexidade para problemas de decisão.



Fonte: Elaborada pelo autor.

Segundo Garey e Johnson (1979, p. 95), um *algoritmo pseudo-polynomial* é aquele que soluciona um problema Π , para cada instância I , e tem sua função de complexidade limitada superiormente por uma função polinomial do tamanho e do maior valor de I . Estes valores são denotados, respectivamente, por $tamanho(I)$ e $max(I)$. Segundo os autores, Π é um *problema numérico* se não existe uma função polinomial p sobre os inteiros tal que $max(I) \leq p \cdot tamanho(I)$, para toda instância I de Π . Uma consequência imediata dessa definição é que se Π não é um problema numérico e é \mathcal{NP} -Completo, então Π não pode ser resolvido por um algoritmo de tempo polinomial a menos que $\mathcal{P} = \mathcal{NP}$.

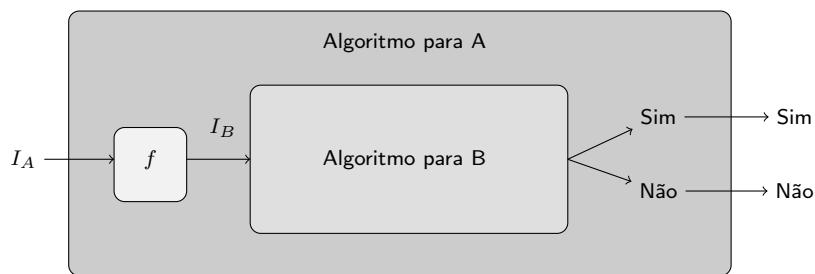
Sejam Π um problema numérico e Π_p um subproblema de Π restrito às instâncias I que satisfazem a desigualdade $max(I) \leq p \cdot tamanho(I)$, para algum polinômio p sobre os inteiros. Se Π pode ser resolvido por algum algoritmo pseudo-polynomial, então Π_p é resolvido em tempo polinomial. Se Π_p é \mathcal{NP} -Completo, então Π é fortemente \mathcal{NP} -Completo. Pode-se dizer então que um problema fortemente \mathcal{NP} -Completo é aquele que permanece \mathcal{NP} -Completo mesmo quando o maior valor numérico de qualquer de suas instâncias é limitado superiormente por um polinômio no tamanho destas. Todo problema não numérico \mathcal{NP} -Completo é fortemente \mathcal{NP} -Completo.

Redução Polinomial e NP-Completude

A noção de mostrar o quanto difícil ou fácil é um problema com base em outro também se aplica quando ambos os problemas são de decisão. Esse artifício é utilizado para mostrar resultados de \mathcal{NP} -Completude. De maneira intuitiva, um problema A pode ser reduzido a um outro problema B se qualquer instância de A puder ser transformada em uma de B , e a solução para este provê uma solução para aquele.

Dados dois problemas de decisão A e B . Diz-se que A se reduz polinomialmente a B (denotado por $A \leq_P B$) se há uma função polinomial f que transforma toda instância I_A do problema A em uma instância I_B do problema B , e a resposta para I_A em A é “sim” se, e somente se, a resposta para I_B em B é “sim”. Esse processo está ilustrado na Figura A.3. Basicamente, uma redução de A para B apresenta um método de solucionar A usando um algoritmo para B . Isso significa que B é pelo menos tão difícil quanto A .

Figura A.3 – Esquema de Redução Polinomial de A para B .



Fonte: Elaborada pelo autor.

Formalmente, um problema A está em NP-C se $A \in \text{NP}$ e, para qualquer problema $B \in \text{NP}$, $B \leq_P A$. Assim, para provar que um dado problema X , que está em NP , é NP -Completo, é suficiente mostrar uma redução polinomial de um problema NP -Completo Y para X . Nesse caso, todos os problemas em NP se reduzem a X via Y .

O conceito original de problemas NP -Completos foi introduzido independentemente por Cook (1971), que mostrou que o *Problema da Satisfatibilidade de Circuitos* é NP -Completo, e por Levin (1973). Em Karp (1972), foram mostrados, via reduções, 21 problemas NP -Completos. Garey e Johnson (1979), uma grande referência nesse assunto, traz um levantamento detalhado da teoria e uma vasta lista de problemas NP -Completos.

Uma fórmula booleana está na *Forma Normal Conjuntiva* (*FNC*) se ela for constituída de cláusulas conectadas por “e” (conjunção), e cada cláusula é formada por *literais* (uma variável ou a sua negação) conectados por “ou” (disjunção). Além disso, não é permitido, em uma mesma cláusula, a ocorrência de uma variável (*literal positivo*) e da sua negação (*literal negativo*). O problema da satisfatibilidade de uma fórmula booleana é formulado como segue:

Problema: SAT

Entrada: Uma fórmula booleana φ na *FNC*, com n variáveis e m cláusulas.

Pergunta: Existe uma atribuição de valores verdade às variáveis de φ que a tornam verdadeira?

Uma fórmula booleana está na *dFNC*, se está na *FNC* e se cada uma de suas cláusulas é formada por exatamente d literais. Assim, o problema *dSAT* corresponde ao *SAT*, restrito a fórmulas booleanas na *dFNC*. O problema *3SAT*, assim como o *SAT*, é muito utilizado em resultados de NP -Compleitude. Ele está no conjunto de problemas NP -Completos de Karp (1972). O problema *2SAT*, por sua vez, pertence à classe \mathcal{P} . Schaefer (1978) considera uma classe infinita de problemas de satisfatibilidade que contêm esses dois problemas particulares como casos especiais, e mostra que cada membro desta classe ou está em \mathcal{P} ou em NP-C , considerando $\mathcal{P} \neq \text{NP}$. Esse resultado é conhecido como o *Teorema da Dicotomia de Schaefer*.

Hipótese Tempo Exponencial (ETH)

Uma questão aberta é se, para resolver o problema *kSAT*, com $k \geq 3$, é requerido tempo exponencial, ou seja, $2^{\delta n}$, para algum δ positivo. A *Hipótese Tempo Exponencial (ETH)*, do inglês *Exponential-Time Hypothesis*, formulada por Impagliazzo e Paturi (1999), afirma que o problema *3SAT* não pode ser resolvido em um tempo subexponencial, ou seja, existe um número positivo s_3 tal que todos os algoritmos que resolvem o *3SAT* têm complexidade $\Omega(2^{s_3 n})$. De uma maneira geral, para $k \geq 3$, os autores definem $s_k = \inf\{\delta : \text{existe um algoritmo de complexidade } 2^{\delta n} \text{ para resolver o } k\text{SAT}\}$. Eles mostraram que a complexidade do *kSAT* cresce juntamente com o k . A validade da *ETH* implica $\mathcal{P} \neq \text{NP}$.

Algoritmos Aproximativos

Muitos problemas de otimização interessantes são \mathcal{NP} -Difíceis. Portanto, a menos que $\mathcal{P} = \mathcal{NP}$, não há algoritmos polinomiais para encontrar soluções ótimas para tais problemas. Uma abordagem que pode ser utilizada nesses casos é buscar por algoritmos que retornem soluções que se aproximem da solução ótima. São os chamados *algoritmos aproximativos*.

De acordo com Williamson e Shmoys (2011), um *algoritmo α -aproximativo* para um problema de otimização é aquele que, para todas as instâncias do problema, produz em tempo polinomial uma solução cujo valor é limitado por um fator α do valor de uma solução ótima. Se $\alpha < 1$, o problema é de maximização; e se $\alpha > 1$, o problema é de minimização. Assim, por exemplo, um algoritmo $(1/2)$ -aproximativo para um problema de maximização é um algoritmo polinomial que sempre retorna uma solução cujo valor é pelo menos metade do valor ótimo.

APÊNDICE B – ALGORITMOS DE FLUXO MÁXIMO

O Método de Ford-Fulkerson

A prova do Teorema 2.2 (Teorema do Fluxo Máximo e Corte Mínimo, de Ford e Fulkerson (1956)) sugere um método simples para encontrar um (s, t) -fluxo máximo em uma rede com os limites inferiores iguais a zero. Este processo está representado no Algoritmo 2.

Algoritmo 2: GenericAugmentingPath(\mathcal{N}, s, t)

```

1 início
2    $x \leftarrow 0;$ 
3   Inicializar  $\mathcal{N}(x)$ ; // Definir fluxo 0 para todo arco da rede
4   enquanto Existir um caminho aumentante faz
5     Identificar um caminho aumentante  $P$ ;
6     Aumentar  $\delta(P)$  unidades de fluxo ao longo de  $P$ ;
7      $x \leftarrow x + \delta(P);$ 
8   fim
9   retorna  $x$ ;
10 fim

```

Inicia-se com um fluxo $x \equiv 0$, que é viável, pois $l_{ij} \leq 0 = x_{ij} \leq u_{ij}$ para todo arco ij . De forma iterativa, busca-se um caminho aumentante P . Se tal caminho existir, então aumenta-se o fluxo, enviando $\delta(P)$ unidades de fluxo ao longo de P . Caso contrário, o processo termina. Uma consequência do Teorema 2.2 é o resultado a seguir:

Teorema B.1 (Teorema da integralidade do fluxo máximo). *Seja $\mathcal{N} = (D, l \equiv 0, u)$ uma rede com fonte s e sumidouro t . Se as capacidades de todos os arcos são inteiras, então existe um (s, t) -fluxo máximo inteiro em \mathcal{N} .*

Demonstração. Segue do Método de Ford-Fulkerson. Inicia-se com um fluxo $x \equiv 0$ e sempre aumenta o fluxo adicionando-se um valor $\delta(P)$ inteiro. Daí, o novo (s, t) -fluxo é também inteiro. Como as capacidades são inteiras, em um número finito de passos será obtido um fluxo máximo. O resultado segue por indução no número de aumentações. \square

Cormen et al. (2022), Bang-Jensen e Gutin (2008) se referem a esse processo como *Método de Ford-Fulkerson*, uma vez que ele não define como o caminho aumentante deve ser identificado a cada iteração, e por haver diferentes implementações e com complexidades distintas para isso. Algumas dessas serão vistas a seguir.

Algoritmo de marcação

O *Algoritmo de Marcação* (do inglês, *The Labeling Algorithm*), descrito em Ahuja, Magnanti e Orlin (1993), é uma implementação especial para o Método de Ford-Fulkerson. Ele identifica caminhos aumentantes em $\mathcal{N}(x)$, buscando por vértices ainda não alcançados por

um caminho direcionado a partir do vértice fonte na rede residual. A cada passo, o algoritmo partitiona o conjunto de vértices da rede em dois grupos: *marcados* (alcançados a partir de s até então) e *não marcados*.

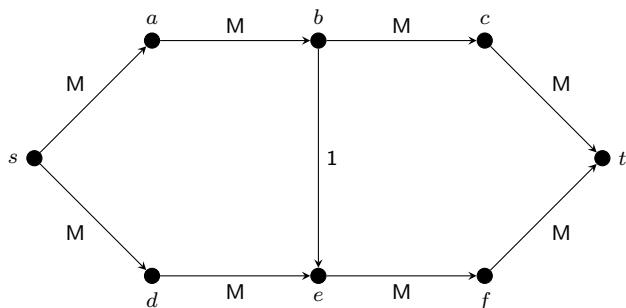
Inicia-se marcando o vértice s . Depois, seleciona-se um vértice marcado ainda não examinado e verifica-se a sua lista de adjacências na rede residual para marcar mais vértices. Se t for marcado, envia-se a maior quantidade de fluxo possível no caminho de s a t . As marcações dos vértices são apagadas e o processo é repetido. Se todos vértices marcados são examinados e t não for marcado, então não há mais caminhos aumentantes e o algoritmo terá encontrado o valor do fluxo máximo e um corte (S, \bar{S}) mínimo, em que S é o conjunto de vértices marcados na última iteração.

De acordo com Ahuja, Magnanti e Orlin (1993), entre as desvantagens desse algoritmo, ele não define critérios para escolher melhor os caminhos aumentantes a cada iteração, caso haja mais de um. Além disso, ele apaga as informações (marcações) geradas ao se mover de uma iteração para a outra para gerá-las novamente, quando muitas dessas informações geradas na iteração anterior são válidas para a atual.

Se as capacidades dos arcos forem inteiras, pelo Teorema B.1, o algoritmo sempre encontra um fluxo máximo x^* em tempo $O(m|x^*|)$. É um dos algoritmos mais simples de implementar para o problema em questão. No entanto, ele é pseudo-polynomial, pois o seu tempo de execução depende também do valor do fluxo máximo. Ele é rápido para valores pequenos de capacidade, mas pouco atrativo para capacidades grandes. Nesse caso, se todos os arcos possuem uma capacidade grande U , $|x^*|$ é limitado a $O(nU)$.

A Figura B.1 traz uma rede possivelmente ruim para esse algoritmo. Os valores nos arcos representam suas capacidades, e M é um inteiro grande. Se forem selecionados os caminhos aumentantes $sabef$ e $sdebct$, respectivamente, nas iterações ímpares e pares, um fluxo máximo é encontrado após $2M$ iterações.

Figura B.1 – Instância ruim para o Algoritmo de Ford-Fulkerson.



Fonte: Bang-Jensen e Gutin (2008, p. 144)

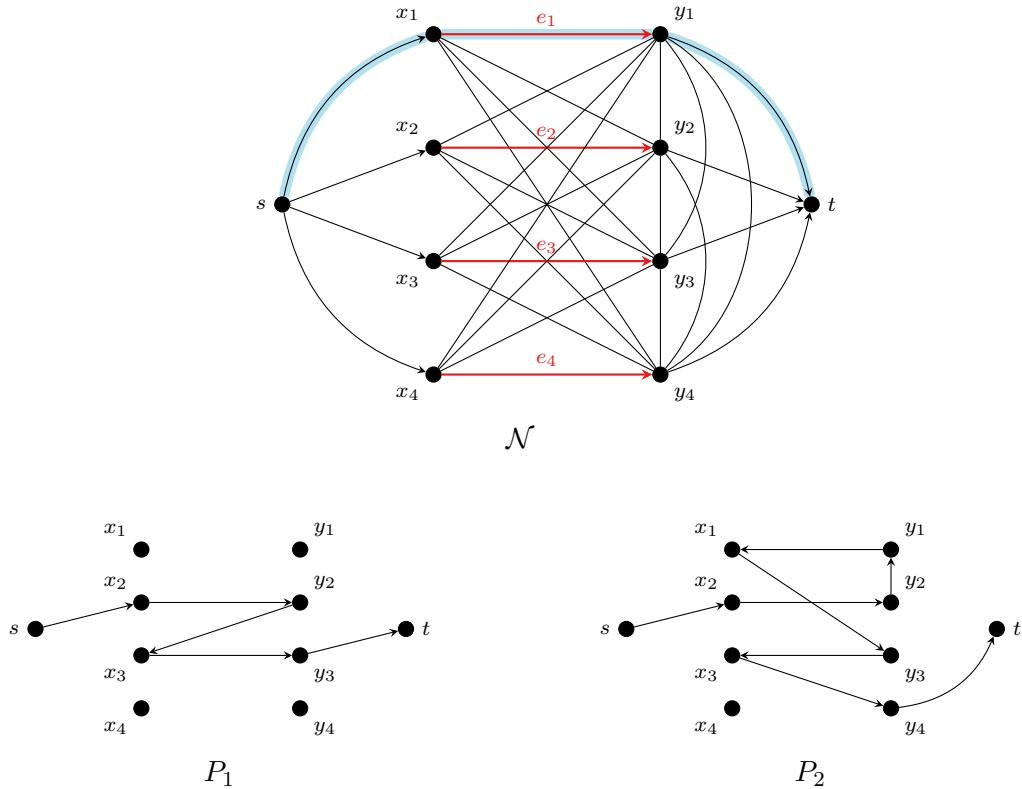
Se as capacidades forem números racionais, é possível escrevê-los com um mesmo denominador comum d ; multiplicar as capacidades dos arcos por esse valor e proceder da mesma forma como é feito para capacidades inteiras.

Se as capacidades dos arcos forem números irracionais, o processo pode continuar de forma indefinida e convergir para um valor estritamente menor que o ótimo. Papadimitriou e Steiglitz (1982, p. 126) trazem um exemplo de rede, descrito por Ford e Fulkerson (1956, p. 36), para o qual o algoritmo converge para um valor que é um quarto do valor ótimo. Os valores a_i , usados para algumas capacidades, satisfazem a diferença:

$$a_{n+2} = a_n - a_{n+1}. \quad (\text{B.1})$$

As condições iniciais são: $a_0 = 1$ e $a_1 = \sigma = (\sqrt{5} - 1)/2 < 1$. Por indução, é possível verificar que $a_i = \sigma^i$, para $i = 0, 1, \dots$. Ford e Fulkerson (1956) observaram que, em certas topologias de rede, sequências de caminhos aumentantes podem ser usados para simular a sequência a_0, a_1, a_2, \dots , o que pode ser visto na rede \mathcal{N} da Figura B.2. A rede possui quatro arcos especiais $e_i = x_i y_i$, com $1 \leq i \leq 4$, cujas capacidades são, respectivamente, a_0, a_1, a_2 e a_2 . A capacidade dos demais arcos é $S = 1/(1 - \sigma)$. Na rede, uma aresta entre vértices uv representa os arcos uv e vu . A figura traz também dois caminhos aumentantes, P_1 e P_2 , que podem ser identificados em certas configurações da rede residual de \mathcal{N} , à medida que o fluxo vai sendo definido.

Figura B.2 – Exemplo de rede para a qual o Método de Ford-Fulkerson pode falhar.



Fonte: Adaptada de Papadimitriou e Steiglitz (1982, p. 127)

As aumentações são feitas da seguinte forma:

- **Passo 0:** Escolher o caminho aumentante sx_1y_1t , de capacidade residual a_0 . Isso faz com que as capacidades residuais de e_1 a e_4 sejam, respectivamente, 0, a_1 , a_2 , a_2 .

- **Passo n** ($n \geq 1$): Como base para indução, seja $\langle e'_1, e'_2, e'_3, e'_4 \rangle$ uma ordenação dos arcos especiais, com capacidades $\langle 0, a_n, a_{n+1}, a_{n+1} \rangle$.

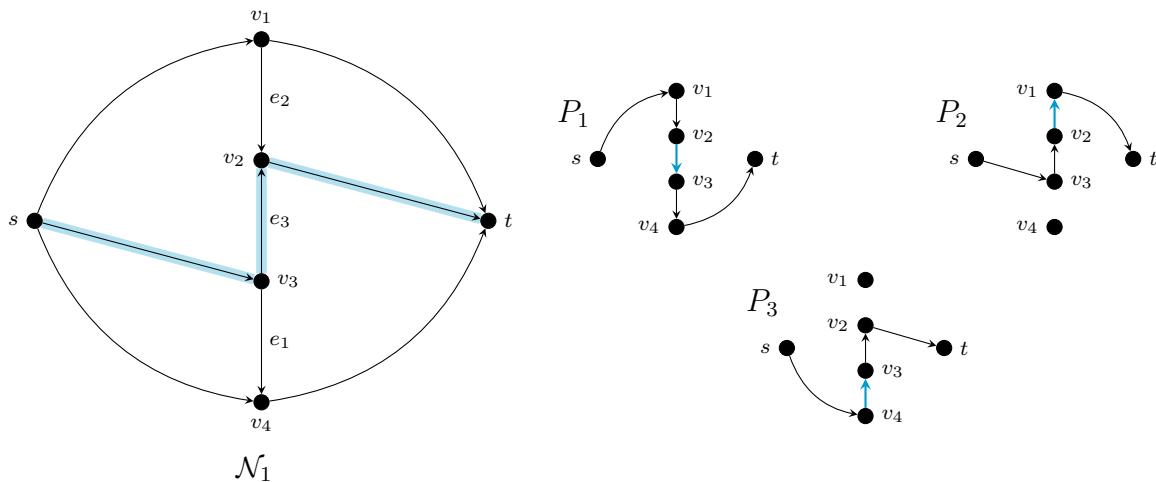
- Escolher o caminho aumentante P_1 , fazendo com que as capacidades residuais dos arcos e'_1 a e'_4 sejam $\langle 0, a_{n+2}, 0, a_{n+1} \rangle$.
- Escolher o caminho aumentante P_2 , fazendo com que as capacidades residuais dos arcos e'_1 a e'_4 sejam $\langle a_{n+2}, 0, a_{n+2}, a_{n+1} \rangle$.

O passo 0 introduz apenas uma aumentoção de fluxo de valor a_0 , pelo caminho destacado. Cada passo n , para $n \geq 1$, é constituído de duas aumentações, uma de valor a_{n+1} e outra de valor a_{n+2} . Dessa forma, pela Equação (B.1), cada um desses passos aumenta o fluxo em a_n unidades. Além disso, cada passo n finaliza com capacidades residuais apropriadas para iniciar o passo $n + 1$. Assim, apesar do valor do fluxo máximo na rede em questão ser $4S$, o valor total do fluxo obtido irá convergir para:

$$a_0 + (a_2 + a_3) + (a_3 + a_4) + \dots = a_0 + a_1 + a_2 + \dots = 1/(1 - \sigma) = S.$$

Usando uma ideia semelhante à de Ford e Fulkerson (1956), Zwick (1995) apresentou as menores redes para as quais o Método de Ford-Fulkerson também falha. Uma delas é a rede \mathcal{N}_1 ilustrada na Figura B.3. Ela possui três arcos especiais, e_1 , e_2 e e_3 , cujas capacidades são, respectivamente, $a_1 = \sigma$, $a_2 = \sigma^2$ e $a_0 = 1$. Todos os demais arcos possuem capacidade $M \geq 4$. O fluxo máximo nessa rede é claramente $2M+1$, que pode ser arbitrariamente grande. Os arcos destacados representam o primeiro caminho aumentante que deve ser utilizado para enviar o fluxo.

Figura B.3 – Uma das menores redes em que o Método de Ford-Fulkerson pode falhar.



arco de retorno, v_2v_1 , que será usado para transferir fluxo de e_2 para e_3 . P_3 possui um arco de retorno, v_4v_3 , que será usado para transferir fluxo de e_1 para e_3 .

Após passar o fluxo utilizando o caminho $P = sv_3v_2t$, o arco e_3 fica saturado e as capacidades residuais de e_1 , e_2 e e_3 passam a ser $\langle a_1, a_2, 0 \rangle$. Aqui, e_1 a e_3 representam os arcos no sentido contrário, dependendo do caminho que será feito por eles. Suponha que, para algum $n \geq 1$, as capacidades residuais desses arcos especiais sejam $\langle a_n, a_{n+1}, 0 \rangle$, e que a capacidade residual de todos os demais arcos seja pelo menos 1. Note que isso é válido, considerando $n = 1$ depois de enviar fluxo por P . Claramente, o gargalo em qualquer caminho aumentante da rede que inclui pelo menos um arco especial na sua direção original é um desses arcos especiais incluídos. Agora tomemos, nesta ordem, os caminhos aumentantes P_1 , P_2 , P_1 e P_3 . As capacidades residuais de e_1 , e_2 e e_3 são:

$$\langle a_n, a_{n+1}, 0 \rangle \xrightarrow{P_1} \langle a_{n+2}, 0, a_{n+1} \rangle \xrightarrow{P_2} \langle a_{n+2}, a_{n+1}, 0 \rangle \xrightarrow{P_1} \langle 0, a_{n+3}, a_{n+2} \rangle \xrightarrow{P_3} \langle a_{n+2}, a_{n+3}, 0 \rangle.$$

Para verificar isso, note que o arco crítico de P_1 é e_2 , também com capacidade residual a_{n+1} . O arco crítico de P_2 é e_3 , com capacidade residual a_{n+1} . O arco crítico de P_1 passa a ser e_1 , com capacidade residual a_{n+2} . Por fim, o arco crítico de P_3 é e_3 , que também possui capacidade residual a_{n+2} . O fluxo na rede é então aumentado, como resultado dessas quatro aumentações, por $2(a_{n+1} + a_{n+2}) = 2a_n$. Isso gera uma sequência infinita de aumentações cujo valor não converge para o máximo, que é $2M + 1$, mas para um valor constante que é dado pela expressão a seguir:

$$1 + 2 \sum_{n=2}^{\infty} a_n = 1 + 2 \cdot (\sigma^2 + \sigma^3 + \dots) = 1 + 2 \cdot \left(\frac{\sigma^2}{1 - \sigma} \right) = 3. \quad (\text{B.2})$$

O valor do fluxo, em qualquer instante, é no máximo 3. Assim, como $M \geq 4$, a capacidade residual de cada arco não especial da rede é pelo menos 1, completando a condição para que a sequência de aumentações apresentada possa ser sempre aplicada.

Uma vez que a computação numérica é sempre realizada em números com precisão finita; do ponto de vista prático, números irracionais não são um problema.

Outros algoritmos baseados em caminhos aumentantes

Como foi visto, o Método de Ford-Fulkerson pode realizar um número grande de aumentações, mesmo para redes com capacidades inteiras. Para contornar esse problema, diversos algoritmos, também baseados em caminhos aumentantes, foram propostos. Nesse tipo de algoritmo, ressalta-se que a regra de conservação do fluxo é respeitada.

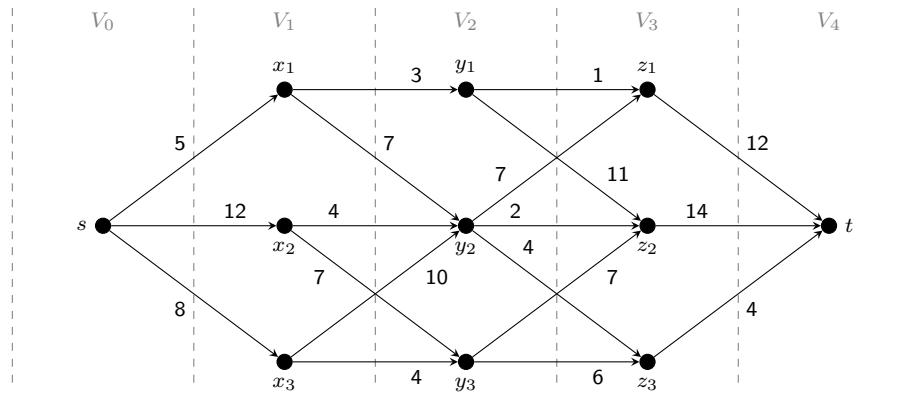
Uma *rede em camadas* é uma rede $\mathcal{N} = (D, l \equiv 0, u)$, com $D = (V, A)$, um vértice fonte s e um sumidouro t , e com as seguintes propriedades:

- (a) Há uma partição $V = V_0 \cup V_1 \cup \dots \cup V_k \cup V_{k+1}$ de modo que $V_0 = \{s\}$, $V_{k+1} = \{t\}$ (cada V_i , com $0 \leq i \leq k$, corresponde a uma camada) e

(b) todo arco de A tem cauda em V_i e cabeça em V_{i+1} , para algum $i = 0, 1, \dots, k$.

Nesse tipo de rede, cada camada i é constituída dos vértices que estão a uma distância i da fonte s . Um exemplo de rede em camadas está representado na Figura B.4. Nesta, observe que o conjunto de vértices foi partitionado em 5 subconjuntos, V_0 a V_4 . Os valores próximos aos arcos representam suas capacidades.

Figura B.4 – Exemplo de rede em camadas.



Fonte: Adaptada de Bang-Jensen e Gutin (2008, p. 147)

Dinic (1970) propôs um algoritmo que usa caminhos aumentantes mais curtos (aqueles que possuem o menor número de arcos). Ele observou que, a cada iteração, pode haver mais de um desses caminhos. Assim, o objetivo uni-los para enviar o fluxo. O algoritmo trabalha com rede em camadas. No início de cada uma de suas fases, o algoritmo constrói uma rede residual \mathcal{N}_x , em relação ao fluxo x corrente, e executa uma busca em largura nesta, a partir de s . Se t não for alcançado, então o fluxo x é máximo e o processo termina. Caso contrário, constrói-se uma rede em camadas L_x . Quando t é alcançado a uma distância k de s , ele é colocado na camada k e os vértices não alcançados a partir de s são descartados. São mantidos em L_x apenas os arcos que vão de uma camada para a próxima. O seu tempo de execução é $O(n^2m)$.

Em Edmonds e Karp (1972), os autores mostraram três variações para o Método de Ford-Fulkerson. Na primeira delas, assim como no Algoritmo de Dinic, o caminho aumentante escolhido a cada iteração é o mais curto. Isso é obtido usando-se uma busca em largura (BFS). O algoritmo realiza $O(mn)$ aumentações e, uma vez que cada caminho aumentante é encontrado em tempo $O(m)$, o fluxo máximo é obtido em tempo $O(m^2n)$.

A segunda abordagem adotada por Edmonds e Karp (1972) busca reduzir o número de aumentações, selecionando, a cada iteração, um caminho aumentante com capacidade residual máxima. Eles mostraram que, quando as capacidades e o fluxo inicial são inteiros, o número máximo de aumentações é limitado por $O(n^2 \log(n\mu))$, onde μ é a média das capacidades dos arcos. Queyranne (1980) mostrou que esse algoritmo pode rodar indefinidamente e, além disso, que não existe para ele um limite inferior independente dos valores de capaci-

dade. Ele também apresentou uma classe de redes com capacidades inteiras para as quais o algoritmo requer $\Omega(m \log(m\mu))$ aumentações.

O Algoritmo 3, que está em Williamson (2019), é uma pequena modificação do Algoritmo 2. Segundo o autor, se as capacidades forem inteiras, ele encontra um fluxo máximo em $O(m \log(mU))$ iterações, onde U é a maior capacidade da rede (esse parâmetro será utilizado em outras análises mais à frente). Observe que, se $m = O(U)$, esse número de iterações é $O(m \log U)$.

Algoritmo 3: MaxCapacity(\mathcal{N}, s, t)

```

1 início
2    $x \leftarrow 0;$ 
3   Iniciarizar  $\mathcal{N}(x)$  ; // Definir fluxo 0 para todo arco da rede
4   enquanto Existir um caminho aumentante faça
5     | Identificar um caminho aumentante  $P$  de capacidade residual máxima;
6     | Aumentar  $\delta(P)$  unidades de fluxo ao longo de  $P$ ;
7     |  $x \leftarrow x + \delta(P);$ 
8   fim
9   retorna  $x;$ 
10 fim

```

Trata-se também de um algoritmo genérico, por não especificar como o caminho aumentante de capacidade residual máxima é identificado. Para isso, o autor sugere ordenar os arcos pela capacidade residual, em ordem decrescente. Isso é feito em tempo $O(m \log m)$. Em seguida, seleciona-se um arco de cada vez (na ordem estabelecida) até que exista um caminho de s a t . Este é um caminho de capacidade residual máxima, cujo tempo para encontrá-lo é $O(m^2)$. De acordo com Kaibel e Peinhardt (2006), é possível encontrar esse caminho em tempo $O(m + n \log n)$, modificando-se o *Algoritmo de Dijkstra*.

Outra abordagem proposta por Edmonds e Karp (1972) é utilizar caminhos aumentantes com capacidade residual grande, não necessariamente a maior. Trata-se de um algoritmo de escalonamento de capacidades (*Capacity Scaling Algorithm*). O processo está descrito no Algoritmo 4, que está em Ahuja, Magnanti e Orlin (1993, p. 211).

O Algoritmo utiliza um parâmetro Δ , e $\mathcal{N}(x, \Delta)$ é a *rede Δ -residual* em relação a um fluxo x , cuja capacidade residual dos arcos é pelo menos Δ . Assim, $\mathcal{N}(x, 1) = \mathcal{N}(x)$, e $\mathcal{N}(x, \Delta)$ é um subdigrafo de $\mathcal{N}(x)$. Isso está ilustrado na Figura B.5.

Os autores se referem a uma fase durante a qual Δ permanece constante como uma Δ -*fase*. Em cada fase, uma aumentação carrega pelo menos Δ unidades de fluxo. Inicia-se com $\Delta = 2^{\lfloor \log_2 U \rfloor}$ e esse valor é reduzido à metade, ao final de cada Δ -fase, até $\Delta = 1$. Logo, ele realiza $O(\log U)$ fases.

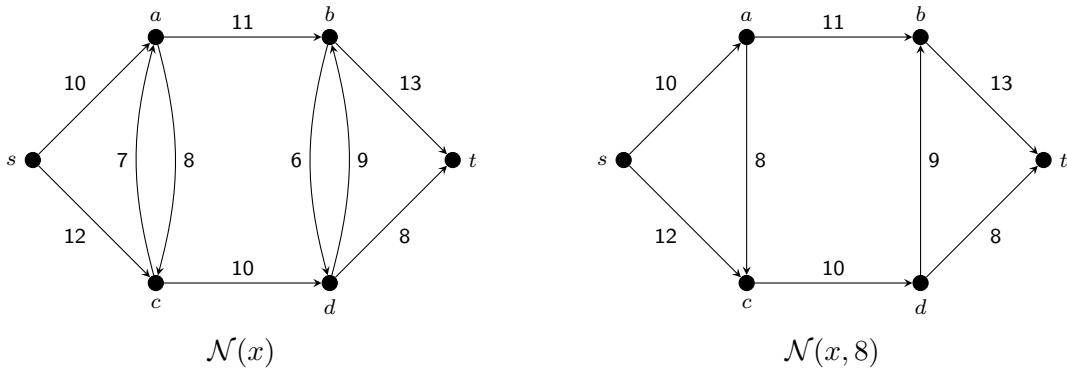
Sejam x^* um fluxo máximo, x' o fluxo obtido ao final de uma Δ -fase e S o conjunto de vértices alcançáveis a partir de s em $\mathcal{N}(x', \Delta)$. Como não há mais caminhos aumentantes em $\mathcal{N}(x', \Delta)$, essa rede possui um (s, t) -corte mínimo (S, \bar{S}) . Observe que $u(S, \bar{S}) \leq m\Delta$. Consequentemente, $|x^*| - |x'| \leq m\Delta$. Após cada fase, a capacidade residual de cada arco é pelo menos $\Delta/2$. Assim, podem ser realizadas no máximo $2m$ aumentações. Na última fase,

Algoritmo 4: CapacityScaling(\mathcal{N}, s, t)

```

1 início
2    $x \leftarrow 0;$ 
3   Iniciarizar  $\mathcal{N}(x)$  ; // Definir fluxo 0 para todo arco da rede
4    $\Delta \leftarrow 2^{\lfloor \log_2 U \rfloor};$ 
5   enquanto ( $\Delta \geq 1$ ) faca
6     Atualizar  $\mathcal{N}(x, \Delta)$ ;
7     enquanto Existir um caminho aumentante em  $\mathcal{N}(x, \Delta)$  faca
8       Identificar um caminho aumentante  $P$ ;
9       Aumentar  $\delta(P)$  unidades de fluxo ao longo de  $P$ ;
10       $x \leftarrow x + \delta(P);$ 
11    fim
12     $\Delta \leftarrow \Delta/2;$ 
13  fim
14  retorna  $x;$ 
15 fim

```

Figura B.5 – Rede Δ -Residual.

Fonte: Elaborada pelo autor.

$\mathcal{N}(x, \Delta) = \mathcal{N}(x)$, e o processo termina com um fluxo máximo. O número de aumentações necessárias para isso é $O(m \log U)$. Como cada caminho aumentante é encontrado em tempo $O(m)$, o tempo do algoritmo é $O(m^2 \log U)$.

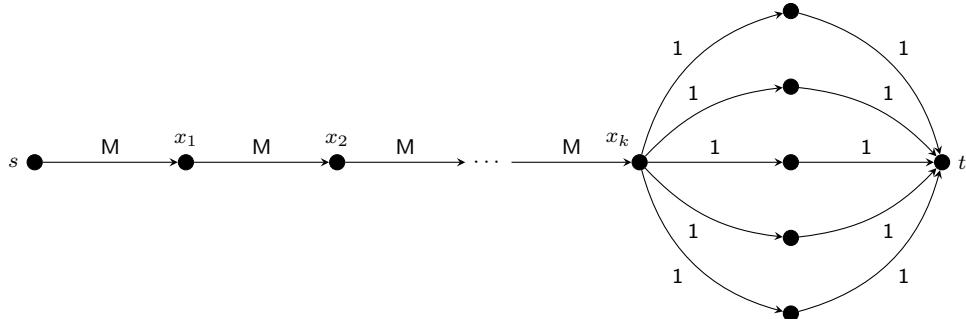
Algoritmos baseados em pré-fluxo

Como a atualização do fluxo é feita arco a arco; nos algoritmos baseados em caminhos aumentantes, o tempo de atualização do fluxo é proporcional ao tamanho do caminho identificado a cada iteração. No exemplo da Figura B.6, devem ser identificados 5 caminhos, cada um destes com tamanho $k+2$, demandando portanto $5k+10$ atualizações. Esse número de atualizações seria reduzido se fosse possível enviar de uma só vez 5 unidades de fluxo no caminho de s a x_k , e em seguida enviar 1 unidade de fluxo em cada um dos 5 caminhos de x_k a t , reduzindo o número de atualizações para $k+10$.

Na segunda abordagem, utiliza-se a noção de pré-fluxo. Segundo Ahuja, Magnanti e Orlin (1993), trata-se de algoritmos mais gerais, poderosos e flexíveis que os de caminho aumentante. São flexíveis, pois não garantem a conservação de fluxo em estágios interme-

diários. Isso se dá pelo fato desse tipo de algoritmo permitir que o fluxo de entrada exceda o de saída nos vértices intermediários. Ao final, o excesso de fluxo é enviado em direção ao sumidouro; e, quando isso não for possível, em direção à fonte.

Figura B.6 – Rede ruim para algoritmos baseados em caminhos aumentantes.



Fonte: Elaborada pelo autor.

Considerando o exemplo da Figura B.6, o algoritmo envia M unidades de fluxo de s a x_1 , deixando este com um excesso igual a M . Sucessivamente, o excesso de fluxo é empurrado em direção a x_k . À medida que o fluxo passa nos arcos $x_i x_{i+1}$, para $1 \leq i < k$, observe que x_i fica balanceado e x_{i+1} fica com um excesso de fluxo igual a M . Quando o fluxo chega a x_k , este pode enviar apenas 5 unidades em direção a t , e fica com excesso igual a $M - 5$. Como não é possível enviar mais fluxo a t , o excesso de x_k é enviado de volta a s , pelos arcos $x_k x_{k-1}$ a $x_1 s$.

Formalmente, um *pré-fluxo* x em uma dada rede \mathcal{N} é um fluxo que respeita as restrições de capacidade dos arcos, mas permite que o balanço em vértices intermediários seja negativo, ou seja, estes podem reter fluxo. Nesse contexto, define-se o *excesso de fluxo* em um vértice v em relação a x como $e_x(v) = -b_x(v)$. Um vértice v é dito *ativo* em relação a um pré-fluxo x , se $e_x(v) > 0$.

Ao contrário dos algoritmos baseados em caminho aumentante, que sempre mantêm a viabilidade da solução, os algoritmos baseados em pré-fluxo buscam alcançar a viabilidade da solução ótima. Nesses, a presença de um vértice ativo indica que a solução é inviável. Por isso, a operação básica deles é selecionar um vértice desse tipo e tentar remover o seu excesso de fluxo, empurrando-o para os seus vizinhos.

O primeiro algoritmo desse grupo foi proposto por Karzanov (1974), utilizando redes em camadas. Ele envia o máximo de fluxo possível da fonte s aos seus adjacentes, que estão na camada 1. O algoritmo trabalha basicamente com duas fases - *fase de avanço* e *fase de balanço*. Na primeira, cada vértice intermediário considera apenas os arcos que partem dele para a próxima camada, e satura-os (com exceção, possivelmente, do último arco examinado), propagando o pré-fluxo em direção ao sumidouro t . Na segunda, ele considera a camada mais alta contendo vértices ativos e, para cada um destes, o fluxo é reduzido até que o seu excesso seja igual a 0. O algoritmo itera as fases de avanço e de balanço até balancear todos os vértices intermediários, e isso é feito em tempo $O(n^3)$.

Segundo Ahuja, Magnanti e Orlin (1993), em alguns casos, para determinação do fluxo máximo, é conveniente utilizar rótulos de distância nos vértices na rede residual. A *distância* pode ser vista como uma função $d : V \rightarrow \mathbb{N}$. Diz-se que uma função de distância é válida com respeito a um fluxo x se ela satisfaz as condições:

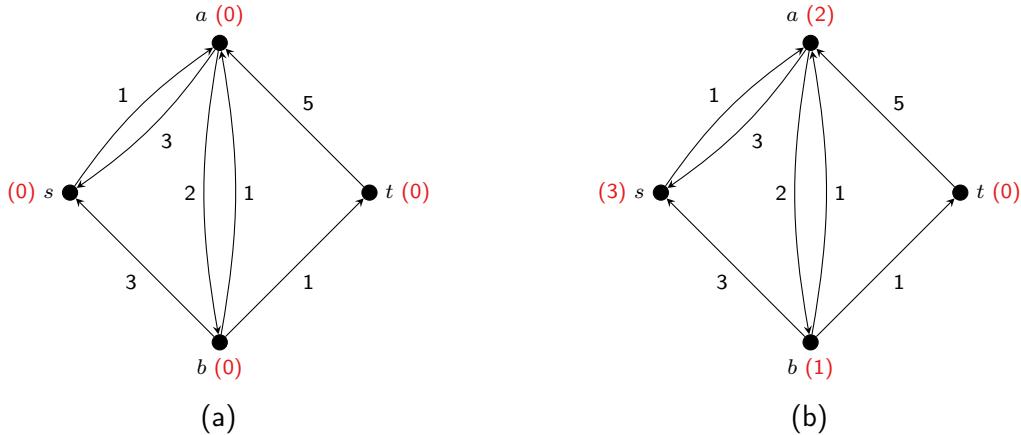
$$\begin{aligned} d(t) &= 0; \\ d(i) &\leq d(j) + 1, \quad \text{para todo arco } ij \text{ em } \mathcal{N}_x \end{aligned} \tag{B.3}$$

Se os rótulos de distância são válidos, $d(v)$ é um limite inferior para o tamanho do caminho direcionado mais curto de v a t na rede residual. Esses são ditos *rótulos de distância exatos* se, para cada vértice v da rede, $d(v)$ corresponde ao tamanho do menor caminho direcionado de v a t na rede residual. É possível determiná-los, para todos os vértices, em tempo $O(m)$, realizando uma busca em largura (BFS) a partir do vértice t .

Diz-se que ij é um *arco admissível* se $d(i) = d(j) + 1$. Caso contrário, ele é dito *inadmissível*. Um *caminho admissível* é composto somente de arcos admissíveis. Um caminho desse tipo é um *caminho aumentante mais curto*.

A Figura B.7 traz exemplos de redes com rótulos de distância válidos. Nesta, os valores próximos aos arcos representam a capacidade residual destes, e os valores entre parênteses representam os rótulos de distância dos vértices. Os rótulos em (a) são válidos, e em (b) são exatos. Nesta última, $sabt$ é um exemplo de caminho admissível.

Figura B.7 – Rótulos de distância válidos e exatos.



Fonte: Adaptada de Ahuja, Magnanti e Orlin (1993, p. 210)

Goldberg e Tarjan (1988) propuseram uma classe de algoritmos chamada de *push-relabel*, usando o conceito de rótulos de distância. Na operação *push*, são examinados vértices intermediários ativos e o excesso de fluxo é empurrado (operação *push*) deles para os seus vizinhos, através de arcos admissíveis. Se um vértice ativo não possuir arcos desse tipo, então o seu rótulo de distância é incrementado (operação *relabel*), no sentido de gerar novos arcos admissíveis para ele. O processo termina quando não há mais vértices ativos. Os passos gerais estão no Algoritmo 5.

Algoritmo 5: PushRelabel(\mathcal{N}, s, t)

```

1 início
2    $x \leftarrow 0;$ 
3    $x_{uv} \leftarrow 0$ , para todo  $uv \in A$ ;
4    $x_{su} \leftarrow u_{su}$ , para todo  $su \in A$ ;
5   Calcular os rótulos de distância exata  $d(v)$ , para todo  $v \in V$ ;
6    $d(s) \leftarrow n$ ;
7   enquanto A rede possuir um vértice ativo faça
8     Selecionar um vértice ativo  $u$ ;
9     se Existir um arco admissível  $uv$  então
10       $\delta \leftarrow \min\{e(u), r_{uv}\}$ ;
11      Enviar  $\delta$  unidades de fluxo de  $u$  para  $v$ ; // Push
12    senão
13       $d(u) \leftarrow \min\{d(v) + 1 : uv \in A \text{ e } r_{uv} > 0\}$ ; // Relabel
14    fim
15  fim
16  retorna  $x$ ;
17 fim

```

A intuição por traz dos algoritmos *push-relabel* é que, por exemplo, em uma rede de fluidos, os vértices podem ser vistos como reservatórios e os arcos como tubulações que os conectam e permitem o escoamento de fluido entre estes. Para que isso aconteça, é necessário que o reservatório origem esteja em um ponto mais elevado que o destino. Ocionalmente, o fluxo pode ficar preso localmente em algum reservatório que não está conectado a outros reservatórios abaixo dele. Nesse momento, o reservatório é elevado, e novamente o fluido pode escoar em direção ao sumidouro.

Na fase de pré-processamento, o fluxo inicial é definido como 0 para todos os arcos da rede, são calculados os rótulos de distância, e é enviado o máximo possível de fluxo da fonte s aos seus adjacentes, e o rótulo de distância de s é inicializado com n . Enquanto houver um vértice ativo, o algoritmo seleciona um desses vértices e verifica se há algum arco admissível partindo dele, de modo que a operação *push* possa ser realizada. A ação de empurrar δ unidades de fluxo ao longo de um arco uv reduz a capacidade residual deste e o excesso em u em δ unidades, e aumenta a capacidade residual no arco vu e o excesso em v na mesma medida. Diz-se que um envio de fluxo em um arco uv é *saturante*, se $\delta = r_{uv}$; e *não saturante*, caso contrário. A operação de *relabel* busca criar pelo menos um arco admissível, para viabilizar novos envios. Caso não seja possível enviar mais fluxo ao sumidouro, as operações de *relabel* farão o excesso de fluxo retornar à fonte.

O tempo de execução do Algoritmo 5 é limitado pelo número de envios de fluxo não saturantes, que, segundo Ahuja, Magnanti e Orlin (1993), é no máximo $O(n^2m)$. Esse limite é comparável ao do Algoritmo de Dinic. No entanto, os autores enfatizam que os algoritmos baseados em pré-fluxo possuem características como a flexibilidade e o potencial para melhorias. Estas se dão, por exemplo, ao se definir regras para a seleção dos vértices ativos. Três abordagens, nesse sentido, estão descritas a seguir:

- a) Goldberg e Tarjan (1988) propuseram uma abordagem que utiliza uma fila para armazenar o conjunto de vértices ativos. O vértice inicial é selecionado para realizar as operações de envio de fluxo, e todos os novos vértices ativos são enfileirados. O tempo de execução desse algoritmo é $O(n^3)$.
- b) Cherian e Mehlhorn (1999) propuseram um algoritmo que seleciona o vértice ativo que possui o rótulo de distância de maior valor. O tempo de execução desse algoritmo é $O(n^2\sqrt{m})$.
- c) Ahuja, Magnanti e Orlin (1993) propuseram uma abordagem usando o conceito de escalonamento de excessos. Nesta, o fluxo é enviado de um vértice com excesso suficientemente grande para um com excesso suficientemente pequeno. O tempo de execução desse algoritmo é $O(mn + n^2 \log U)$.

Cerulli, Gentili e Iossa (2008) traz a análise de uma potencial desvantagem dos algoritmos de pré-fluxo - o *efeito ping pong*. Isso ocorre quando unidades de fluxo são trocadas iterativamente entre dois vértices u e v ao longo de uma sequência de operações *push-relabel*. Para contornar o problema, os autores propuseram um algoritmo que combina as abordagens de pré-fluxo e de caminho aumentante, para garantir que, iniciando em um vértice u , em vez de enviar fluxo usando um arco uv com $d(u) = d(v) + 1$, usa-se um arco uw com $d(u) \geq d(w) + k$, para um dado inteiro k constante.

Algoritmos mais rápidos

Como visto na Seção 2.3.1 (Teorema 2.1 - Decomposição de Fluxos), um fluxo x em uma rede \mathcal{N} pode ser decomposto em até $n + m$ caminhos e ciclos e em tempo $O(mn)$. Uma pergunta que durante muito tempo ficou em aberto buscava saber se seria possível encontrar um fluxo máximo em uma rede nesse mesmo tempo. Essa pergunta foi respondida de forma positiva por Orlin (2013). O melhor algoritmo até então, para o problema do fluxo máximo era o proposto por King, Rao e Tarjan (1992), com tempo de execução em $O(mn \log_{m/n \log n} n)$. Orlin mostrou que o problema pode ser resolvido em tempo $O(mn + m^{31/16} \log^2 n)$. Quando $m = O(n^{(16/15)-\varepsilon})$, esse tempo de execução é $O(mn)$. Segundo o autor, pelo fato do algoritmo proposto por King, Rao e Tarjan (1992) resolver o problema do fluxo máximo em tempo $O(mn)$ para $m > n^{1+\varepsilon}$, as melhorias aplicadas por ele estabelecem que o problema pode ser resolvido em tempo $O(mn)$ para todo m e n . O autor mostrou ainda que, se $m = O(n)$, o seu algoritmo resolve o problema em tempo $O(n^2 / \log n)$.