

FUNDAÇÃO GETÚLIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA
EMAP
PROGRAMA DE GRADUAÇÃO EM MATEMÁTICA APLICADA

POLYANA SAMPAIO RAMOS BARBOZA

O PARADOXO DE BRAESS

Rio de Janeiro
2016

FUNDAÇÃO GETÚLIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA
EMAP
PROGRAMA DE GRADUAÇÃO EM MATEMÁTICA APLICADA

POLYANA SAMPAIO RAMOS BARBOZA

O PARADOXO DE BRAESS

Trabalho de Conclusão de Curso de Matemática Aplicada submetido ao corpo docente da Escola de Matemática Aplicada da Fundação Getúlio Vargas como parte dos requisitos necessários para obtenção do título de Bacharel em Matemática Aplicada.

Orientador: Moacyr Alvim

Rio de Janeiro
2016

Barboza, Polyana Sampaio Ramos

O PARADOXO DE BRAESS / Polyana Sampaio Ramos Barboza.
– 2016.

62 f.: il.

Trabalho de Conclusão de Curso (Graduação em Matemática Aplicada) – Fundação Getúlio Vargas, Escola de Matemática Aplicada-EMAp, Programa de Graduação em Matemática Aplicada, Rio de Janeiro, 2016.

Orientador: Moacyr Alvim.

FGV-EMAp

POLYANA SAMPAIO RAMOS BARBOZA

O PARADOXO DE BRAESS

Trabalho de Conclusão de Curso de Matemática Aplicada submetido ao corpo docente da Escola de Matemática Aplicada da Fundação Getúlio Vargas como parte dos requisitos necessários para obtenção do título de Bacharel em Matemática Aplicada.

Aprovado em: Rio de Janeiro, ____ de _____ de _____.

Moacyr Alvim (Orientador)

Paulo Cezar (Professor Tutor)

Rio de Janeiro
2016

AGRADECIMENTOS

Agradeço à Deus, por ter imposto a Matemática à humanidade e por ter me dado a sensibilidade para perceber que ela está em cada detalhe da sua criação.

À todos os professores da EMAP, por terem disponibilizado seu tempo para passar seus conhecimentos. Em particular, ao professor e orientador Moacyr Alvim, por toda ajuda e brilho nos olhos que encanta ao passar seu amor pela Matemática.

Aos meus pais, por todos os valores que me ensinaram e recursos que disponibilizaram, principalmente o amor e a paciência nos meus momentos de grande estresse e aflição.

À minha gêmea Natalie Barboza, por todo o apoio que me deu pessoalmente e por orações e pela confiança que sempre depositou em mim e na minha capacidade.

À minha amiga-irmã Bruna Paixão, por todo o carinho e apoio, por ter me consolado nas diversas crises de choro e questionamentos do caminho, por ter me incentivado incansavelmente e por ter acreditado em mim até quando nem eu mesma acreditava.

À minha sister Irmã Maria Aparecida, pelas incansáveis orações.

À minha madrinha Marilena Barboza, por ser minha inspiração de vida acadêmica e dedicação aos estudos.

À minha madrinha Ianaê Faraj, por ter me apoiado, mesmo que de longe, atendendo à todas as minhas ligações desesperadas e inseguras.

À toda a minha família, por ter entendido todas as minhas ausências em momentos, muitas vezes, importantes.

À minha Equipe de Jovens de Nossa Senhora, por todas as orações e acompanhamento de trajetória.

À minha irmã de alma Fernanda Duarte, por partilhar de todos os sentimentos de alegria e de tristeza do caminho, pela presença em todas as experiências mais marcantes da faculdade, pelo exemplo que é de dedicação e de força para nunca desistir, por ser a minha dupla perfeita e, principalmente, por ser exemplo de amor e sensibilidade com o próximo e de simplicidade com a vida.

Ao meu amigo Pedro Amaro, por ser o irmão e incansável companheiro que encontrei na faculdade.

À minha amiga Anna Cecília, por cada abraço pelos corredores da FGV.

Aos amigos João Amaro e Gabriel Mesquita, por terem me mostrado a importância de valorizar a amizade e os momentos únicos juntos.

E ao meu amigo José Valentim, por ser a luz de Cristo que encontrei na faculdade.

RESUMO

Barboza, Polyana Sampaio Ramos. **O PARADOXO DE BRAESS**. 2016. 61 f. Trabalho de conclusão de curso (Graduação em Matemática Aplicada) - EMAP, Escola de Matemática Aplicada, Fundação Getúlio Vargas, Rio de Janeiro, 2016.

O caos do trânsito das grandes cidades é notável. Muitos diriam que o motivo é a grande quantidade de veículos, e isso é inegável. Mas existe algum modo de melhorar o trânsito apesar dessa grande quantidade de veículos? A resposta é que sim, existem muitas análises que podem ser feitas sobre o comportamento dos motoristas e sobre a configuração das malhas rodoviárias em prol da minimização do tempo gasto no trânsito. Uma dessas análises é a análise do comportamento egoísta dos motoristas, que nos leva à base do estudo desse trabalho, o Paradoxo de Braess.

O Paradoxo de Braess ocorre quando a adição de uma nova estrada, em união com o desejo que cada motorista tem de minimizar apenas o seu próprio tempo de viagem - roteamento egoísta -, faz o tempo de viagem individual aumentar, quando comparado ao tempo antes da adição da estrada.

Vamos começar analisando os dois diferentes tipos de equilíbrio alcançados em uma rede de fluxo de automóveis, o equilíbrio de Nash e o equilíbrio ótimo. Feito isso, vamos compará-los com o que chamaremos de preço da anarquia. Depois começará a tentativa de detectar a ocorrência do paradoxo de Braess em uma rede qualquer. Porém, vão aparecer muitas más notícias pelo caminho, que tornará esse objetivo mais difícil, já que $P \neq NP$.

Esse trabalho é a base inicial para o estudo de casos mais particulares - com funções de custo das ruas delimitadas, assim como o tamanho das redes - que serão feitos futuramente, objetivando encontrar realmente a ocorrência do Paradoxo na malha rodoviária das grandes cidades.

Palavras-chave: Paradoxo de Braess, Teoria dos Jogos, Grafo, Fluxo, Nash, Ótimo, Algoritmo de Aproximação, NP.

LISTA DE FIGURAS

Figura 1.1: Exemplo de ocorrência do Paradoxo de Braess	11
Figura 3.1: Exemplo de ocorrência do Paradoxo de Braess	25
: (a)Funções de Custo	35
: (b)Funções de Custo Marginais	35
Figura 4.2: Paradoxo de Braess	35
Figura 4.3: Exemplo do método sugerido	42
Figura 4.4: Código para encontrar os fluxos e o custo total do grafo da figura 4.3	43
Figura 4.5: Exemplo do método sugerido sem aresta	44
Figura 4.6: Código para encontrar os fluxos e custo total do grafo da figura 4.5	44
Figura 5.1: Prova da Proposição 5.2.2. Em uma "não"instância de 2DDP, a existência de caminhos $s_1 - t_1$ e $s_2 - t_2$ implica na existência de um caminho $s_2 - t_1$	54
Figura 5.2: Sem Aresta Extra	55
Figura 5.3: Com Aresta Extra	56

SUMÁRIO

1	INTRODUÇÃO	9
1.1	O Paradoxo	9
1.2	O Paradoxo aplicado ao tráfego e a Teoria dos Jogos	10
1.3	Um Exemplo	11
1.4	Motivação	12
1.5	Objetivo	13
2	PRELIMINARES	14
2.1	Equilíbrio de Nash	14
2.1.1	Teoria dos Jogos	14
2.1.2	Estratégias Pura e Mista	14
2.1.3	Jogos de n Indivíduos	14
2.1.4	O Equilíbrio	15
2.2	Ótimo de Pareto	16
2.3	Jogos Cooperativos X Jogos Não-Cooperativos	16
2.4	Preço da Anarquia	17
2.5	Problemas P, NP, NP-Difíceis e NP-Completo	17
2.5.1	Problemas P	17
2.5.2	Problemas NP	17
2.5.3	Problemas NP-Completo	18
2.5.4	Problemas NP-Difíceis	18
2.5.5	P Vs. NP	19
3	O QUE É O PARADOXO DE BRAESS?	21
3.1	O Modelo	21
3.2	Roteamento Egoísta (Selfish Routing)	23
3.3	Definição do Paradoxo de Braess	24
3.4	Exemplo do artigo de Braess	25
4	TENTATIVA POR EQUILÍBRIO DE NASH	27
4.1	Fluxo de Nash	27
4.2	Fluxo Ótimo e Relações com o Fluxo de Nash	30
4.2.1	Fluxo Ótimo	30
4.2.2	Função de Custo Marginal	34
4.2.3	Preço da Anarquia	36
4.3	Existência e Unicidade	37

4.4	Relação com o Paradoxo de Braess	40
5	TENTATIVA POR ALGORITMOS DE APROXIMAÇÃO	46
5.1	O que é um Algoritmo de Aproximação?	47
5.2	Redes com Funções de Custo Afins ($ax + b$)	48
5.2.1	O Preço da Anarquia	48
5.2.2	Limitando Superiormente o Paradoxo de Braess	52
5.2.3	Detectando o Paradoxo de Braess	53
5.3	O Mar de <i>there is no</i>	57
6	CONCLUSÃO	59
	REFERÊNCIAS	60

1 INTRODUÇÃO

O tema proposto para este Trabalho de Conclusão de Curso se desenvolve em torno de uma das aplicações do Paradoxo de Braess, o Paradoxo de Braess no tráfego. Intuitivamente, pode-se pensar que um atalho melhora o fluxo dos veículos em uma rede rodoviária, mas, na verdade, ele pode piorar, e é nisso que o paradoxo de Braess se baseia.

1.1 O Paradoxo

O paradoxo de Braess é uma explicação para uma aparente melhora em uma rede ser capaz de piorar o fluxo através dela. Uma das possíveis aplicações dele é no fluxo de veículos por uma rede rodoviária e, nesse caso, ele é enunciado da seguinte forma: Dada uma rede rodoviária, um ponto inicial, o número de veículos começando por ele e o destino deles, deseja-se estimar a distribuição do fluxo de tráfego. Supondo que a preferência de cada motorista é baseada não apenas na qualidade da estrada, mas também na densidade do fluxo, e que cada motorista escolhe o caminho que mais favorece a ele mesmo (motoristas egoístas), ou seja, o resultado do tempo de percurso não será necessariamente ótimo (no sentido de Pareto), pode-se mostrar que uma extensão na rede rodoviária pode causar uma redistribuição do tráfego que resulta em um maior tempo de percurso individual.

1.2 O Paradoxo aplicado ao tráfego e a Teoria dos Jogos

O paradoxo é um caso claro da Tragédia dos Comuns, onde indivíduos agindo de forma independente e racionalmente de acordo com seus próprios interesses se comportam em contrariedade aos melhores interesses de uma comunidade, resultando em consequências desastrosas para todos.

Levando-se em conta esses motoristas egoístas, sabemos que eles alcançarão o que é conhecido por Equilíbrio de Nash, no qual nenhum motorista sozinho pode fazer melhor mudando sua estratégia unilateralmente. Mas esse Equilíbrio de Nash encontrado é menos eficiente que o equilíbrio alcançado quando os motoristas agem de forma não egoísta, isto é, quando coordenam seus movimentos para beneficiar todo um grupo, conhecido como Ótimo de Pareto.

Sendo assim, podemos calcular o Preço da Anarquia na rede rodoviária, que é, em linhas gerais, o valor do egoísmo, ou seja, como a eficiência de um sistema degrada devido ao comportamento egoísta dos agentes.

Esse conceitos importantes de teoria dos jogos vão ser tratados no próximo capítulo.

1.3 Um Exemplo

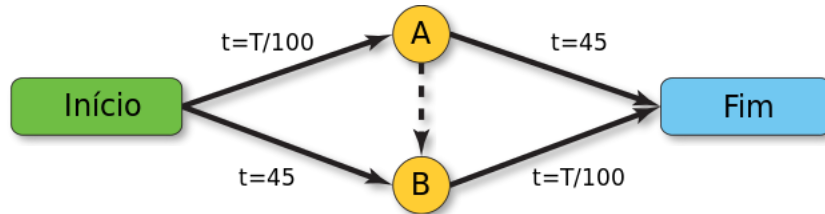


Figura 1.1: Exemplo de ocorrência do Paradoxo de Braess

Sejam 4000 motoristas indo do ponto *Início* para o *Fim*.

Se não existir $A \rightarrow B$, o tempo necessário para dirigir de $Início \rightarrow A \rightarrow Fim$ com n_1 motoristas seria $\frac{n_1}{100} + 45$. E o tempo de $Início \rightarrow B \rightarrow Fim$ com n_2 motoristas seria $\frac{n_2}{100} + 45$. Podemos deduzir que $n_1 = n_2 = 200$ quando o sistema está em equilíbrio, onde nenhum motorista tem incentivo a mudar, pois, sendo os motoristas racionais (preferem as rotas mais curtas), se existisse, para algum, uma rota mais curta, não seria um equilíbrio de Nash, já que ele mudaria de rota.

Assim, cada rota leva $\frac{2000}{100} + 45 = 65$ minutos.

Agora, suponhamos que o atalho $A \rightarrow B$ exista e seja uma rua com um tempo de viagem extremamente curto, de aproximadamente 0 minutos. Nessa situação, todos os motoristas escolherão $Início \rightarrow A$ em vez de $Início \rightarrow B$, porque $Início \rightarrow A$ terá no máximo $\frac{4000}{100} = 40$ minutos, enquanto $Início \rightarrow B$ tem garantidamente 45 minutos. No ponto A, todo motorista racional escolherá a rua "livre" para B para depois ir para *Fim*, por um raciocínio análogo ao anterior. Cada motorista, então, fará $\frac{4000}{100} + \frac{4000}{100} = 80$ minutos > 65 minutos.

Ninguém tem incentivo para mudar, já que os dois caminhos iniciais, $Incio \rightarrow A \rightarrow Fim$ e $Incio \rightarrow B \rightarrow Fim$, tem agora 85 minutos. Se todos concordassem em não usar $A \rightarrow B$, todos iriam se beneficiar reduzindo 15 minutos do tempo de viagem. Entretanto, pelo motivo de que algum motorista vai sempre querer se beneficiar pegando o caminho $A \rightarrow B$, já que eles são egoístas, o ótimo social não será estável e o paradoxo de Braess ocorre.

1.4 Motivação

Não é surpreendente que o egoísmo pode causar aumento no tempo de tráfego dos motoristas. No artigo [1], é feita uma análise que mostra que no Rio de Janeiro, por exemplo, no horário de pico da manhã, 18% do tempo extra de trânsito que os motoristas gastam, quando comparado ao horário em que as ruas estão livres (em geral, de madrugada), é causado pelo roteamento egoísta.

Como uma das consequências desse egoísmo é o Paradoxo de Braess, é possível encontrar casos de ocorrências reais do paradoxo de Braess. Estudos recentes de redes de tráfego em grandes cidades, como Boston, Nova Iorque e Londres, encontraram várias estradas cuja presença aumentava o tempo de viagem no Equilíbrio de Nash [7]. Em 1990 na cidade de Nova Iorque, a *42d Street*, uma rua muito movimentada da cidade, foi fechada para o *Earth Day* e muitos acharam que isso geraria um trânsito histórico na cidade, o que, felizmente, não ocorreu. O fechamento da *42d Street* em 1990 foi a causa de uma melhora no fluxo de tráfego na cidade de Nova Iorque [10].

1.5 Objetivo

O objetivo do presente trabalho é estudar o Paradoxo de Braess no seu caso mais geral para, futuramente, estudar casos mais particulares e, finalmente, tentar encontrar ocorrências do Paradoxo em grandes cidades (em particular, no Rio de Janeiro).

2 PRELIMINARES

2.1 Equilíbrio de Nash

2.1.1 Teoria dos Jogos

Em diversos momentos do cotidiano, nos deparamos com tomada de decisões nas quais as consequências não dependem apenas da nossa escolha, mas passam por escolhas de outros indivíduos. Um exemplo bem claro é a escolha de qual caminho pegar para chegar ao trabalho, por exemplo, no qual, a depender da escolha de outros motoristas, você pode chegar mais rápido ou levar mais tempo. A teoria do jogos é uma teoria matemática que modela fenômenos observados a partir da interação de dois ou mais indivíduos, descrevendo processos de decisão [15].

2.1.2 Estratégias Pura e Mista

Estratégia é o conjunto de possíveis decisões de cada jogador. Estratégia pura é uma estratégia não baseada em aleatoriedade e estratégia mista é aquela definida, para cada jogador, por uma distribuição de probabilidade sobre o conjunto de suas estratégias puras. [8]

2.1.3 Jogos de n Indivíduos

Primeiramente, definimos o conceito de um jogo de n indivíduos. Nesse jogo, cada jogador tem um conjunto finito de estratégias puras e existe um conjunto

definido de pagamentos aos n jogadores (*pay-offs*), correspondente a cada n -tupla de estratégias puras, uma estratégia de cada jogador. Para estratégias mistas, as funções de *pay-off* são as expectativas dos jogadores, com as quais eles jogam suas várias estratégias puras. Cada jogador tem um conjunto de estratégias que definem um espaço em \mathbb{R}^m , onde m é a quantidade de estratégias. Assim, cada n -tupla de estratégias é um ponto no espaço obtido pela multiplicação dos n espaços de estratégias dos jogadores [8].

2.1.4 O Equilíbrio

John Nash, em seu artigo [8], define o equilíbrio de Nash da seguinte forma: “Uma n -tupla contraria outra se a estratégia de cada jogador na n -tupla contrariadora gera a maior expectativa possível para seus jogadores contra as $n - 1$ estratégias dos outros jogadores na n -tupla contrariada. Uma n -tupla auto-contrariadora é chamada de um ponto de equilíbrio.”

Isso quer dizer que o Equilíbrio de Nash é um conjunto de n estratégias, uma para cada jogador, em que um jogador i , dadas as estratégias dos outros $n - 1$ jogadores, não tem incentivo a mudar de estratégia, i.e., a escolha de cada jogador é a melhor resposta às escolhas dos outros $n - 1$ jogadores. Enquanto isso, em um conjunto de estratégias que não é um equilíbrio, sempre vai haver um jogador para quem mudar de estratégia o faz ter um resultado melhor, dadas as estratégias dos outros $n - 1$ jogadores [6].

2.2 Ótimo de Pareto

O ótimo de Pareto é um equilíbrio que minimiza o *pay-off* global, ou seja, minimiza o bem-estar social. Em linhas gerais, quando o ótimo de Pareto é alcançado, nenhum jogador consegue melhorar o seu *pay-off* sem piorar o de algum outro jogador [2].

2.3 Jogos Cooperativos X Jogos Não-Cooperativos

Um equilíbrio de Nash não é, necessariamente, a configuração que determina o melhor resultado para a sociedade, apesar de ser o melhor para cada indivíduo quando ele pensa em maximizar apenas o seu próprio retorno. É essa a principal diferença entre jogos cooperativos e jogos não-cooperativos.

Nos jogos cooperativos, os jogadores pensam no bem-estar da sociedade e, em algumas vezes, fazem sacrifícios para maximizar o resultado geral. Já nos jogos não-cooperativos, os jogadores são egoístas e tomam decisões pensando apenas em si mesmos. Como, em geral, os jogadores são egoístas, pode-se forçar as escolhas dos jogadores em prol do ótimo social através de um agente centralizador ou através de incentivos.

Sendo assim, o equilíbrio de Nash pode não ser um ótimo de Pareto. Dois exemplos onde a cooperação levaria a resultados melhores para os jogadores são no clássico Dilema dos Prisioneiros e no Paradoxo de Braess, sobre o qual falaremos mais adiante.

2.4 Preço da Anarquia

O preço da anarquia é uma medida da discrepância entre o equilíbrio alcançado no jogo não-cooperativo e o ótimo alcançado no jogo cooperativo. Assim, o preço da anarquia é a maior razão entre o ótimo social e o equilíbrio egoísta, i.e., é a pior perda de bem-estar social causada pelo egoísmo [11].

2.5 Problemas P, NP, NP-Difíceis e NP-Completo

Primeiramente, problema de decisão é um problema que demanda “sim” ou “não” como resposta.

2.5.1 Problemas P

Conjunto de todos os problemas de decisão que podem ser resolvidos em tempo polinomial. Isto é, dada uma instância do problema, a resposta de "sim" ou "não" pode ser dada em tempo $O(n^k)$ para alguma constante k , onde n é o tamanho da entrada para o problema.

2.5.2 Problemas NP

NP significa “non-deterministic polynomial time”, o que quer dizer que são verificáveis em tempo polinomial, ou seja, é o conjunto de problemas de decisão para os quais as instâncias onde as respostas são “sim” tem provas de verificação eficientes. Isso significa que se alguém der uma instância de um problema e um certificado para a resposta ser "sim", podemos checar se ele está correto em tempo

polinomial.

2.5.3 Problemas NP-Completos

É uma classe de complexidade que representa o conjunto de todos os problemas X em NP para o qual é possível reduzir qualquer outro problema NP Y para X em tempo polinomial. Intuitivamente, isso significa que podemos resolver Y rapidamente se sabemos como resolver X rapidamente.

Precisamente, Y é redutível a X , se existir algoritmo de tempo polinomial f para transformar instâncias y de Y em instâncias $x = f(y)$ em X em tempo polinomial, com a propriedade de que a resposta para y é "sim" se, e somente se, a resposta para $f(y)$ é "sim".

Um exemplo clássico de problema NP-completo é o de dizer se um dado grafo tem ou não um caminho hamiltoniano.

2.5.4 Problemas NP-Difíceis

Existem ainda os problemas que são pelo menos tão difíceis quanto os problemas NP-completos. Note que os problemas NP-difíceis não tem que estar em NP, e eles não tem que ser problemas de decisão.

A definição precisa é que um problema X é NP-difícil, se existe um problema NP-completo Y , tal que Y é redutível a X em tempo polinomial.

Mas como qualquer problema NP-completo pode ser reduzido a qualquer ou-

tro problema NP-completo em tempo polinomial, todos os problemas NP-completos podem ser reduzidos a qualquer problema NP-difícil em tempo polinomial. Então, se existir um solução para algum problema NP-difícil em tempo polinomial, existe uma solução para todos os problemas NP e NP-completos em tempo polinomial.

Um exemplo de um problema NP-difícil é o de encontrar um caminho hamiltoniano para um dado grafo.

2.5.5 P Vs. NP

Provar se $P = NP$ ou $P \neq NP$ é o problema mais famoso da ciência da computação teórica e uma das perguntas mais importantes de toda a matemática, ainda em aberto. O instituto Clay oferece um milhão de dólares para quem der uma solução para o problema. Mas esse problema vai muito além de um problema matemático abstrato, ele procura determinar de uma vez por todas que tipos de problemas podem ser resolvidos por computadores e que tipos não podem.

Está claro que P é um subconjunto de NP . Então, a pergunta “ P é igual a NP ?” significa “Se a solução para um problema pode ser verificada em tempo polinomial, ela pode ser encontrada em tempo polinomial?”, ou seja, “os problemas NP tem ou não soluções determinísticas em tempo polinomial?” [4, 5].

Mas e se $P = NP$?

- Então a chave pública da criptografia seria impossível;
- Então, como todos os problemas de otimização NP-completos se tornariam fáceis, tudo ficaria mais eficiente (vale ressaltar que uma solução eficiente para qualquer problema NP-completo implicaria que $P=NP$ e em uma solução efi-

ciente para todo problema NP-completo);

- Então o transporte de pessoas e bens seria muito mais rápido e barato;
- “If $P=NP$, then the world would be a profoundly different place than we usually assume it to be, and we’d probably have noticed by now. There would be no special value in “creative leaps,” no fundamental gap between solving a problem and recognizing the solution once it’s found. Everyone who could appreciate a symphony would be Mozart; everyone who could follow a step-by-step argument would be Gauss” – Scott Aaronson, pesquisador de complexidade do Laboratório de Ciência da Computação e Inteligência Artificial do MIT.

Assim, acredita-se largamente que P não é igual a NP , apesar de ainda não existir prova a respeito. Richard Lipton, um cientista da computação de Georgia Tech que estuda o problema P vs. NP diz: “The assumption that P does not equal NP help us develop new mental technologies. Even though we’ve been writing algorithms for decades, we don’t fully understand what they’re capable of. So even if you proved that P does not equal NP —something that everyone already believes—it would have to radically expand our understanding of those capabilities, and make many new things possible with computers, in addition to all the clever workarounds we’ve already found.”

3 O QUE É O PARADOXO DE BRAESS?

3.1 O Modelo

O foco desse trabalho é na aplicação do paradoxo de Braess a redes de tráfego. Assim, vamos modelar uma rede de tráfego como um grafo direcionado $G = (V, E)$, tal que V é um conjunto finito de vértices (nós) e E é um conjunto finito de arestas (links). Uma aresta é um par ordenado de vértices distintos, direcionado do primeiro vértice, a *calda*, para o segundo vértice, a *cabeça*.

Alguns vértices s_1, \dots, s_k são chamados de *origens*, vértices de onde o tráfego sai. Todo tráfego que sai de uma origem s_i pretende chegar a um *destino* (sumidouro) t_i . Sendo assim, chamamos o par origem-destino s_i, t_i de *comódite* i . Duas comódites podem compartilhar vértices, mas não compartilham a mesma origem e o mesmo destino. Uma rede é *comódite única* se tem um único par origem-destino, e *multicomódite*, caso contrário.

Um *caminho* (*path*) é uma rota de um vértice para outro que não circula em volta de si mesmo. Já um *passeio* (*walk*) é sequência de arestas na qual a calda de cada aresta depois da primeira coincide com a cabeça da aresta anterior. Um passeio é *fechado* (*closed*) se ele começa e termina no mesmo vértice. Um passeio fechado é um *ciclo* se nenhum vértice pertence a mais de duas arestas. Um caminho é então um passeio que não contém ciclos. Um caminho $u - v$ é um caminho que começa no vértice u e termina no vértice v .

Para uma rede G , denotamos o conjunto de caminhos $s_i - t_i$ por φ_i e definimos

$\wp = U_i \wp_i$, tal que $\wp_i \neq \emptyset$. Um *fluxo* é um vetor real não-negativo, indexado por \wp . Interpretamos um fluxo f como rotas agregadas escolhidas por um grande número de indivíduos ou agentes, com f_P medindo a quantidade de agentes que usam a rota P – note que cada indivíduo controla apenas uma parte desprezível do tráfego total (jogos não-atômicos - [11]).

Um fluxo f induz um fluxo nas arestas $f_{e \in E}$, onde $f_e = \sum_{P \in \wp: e \in P} f_P$ é a quantidade de fluxo usando caminhos que incluem a aresta e . Se um fluxo f induz o fluxo nas arestas $f_{e \in E}$, vamos, às vezes, chamar o fluxo f de uma decomposição de caminhos de $f_{e \in E}$. Um fluxo induz um único fluxo nas arestas, mas um fluxo nas arestas pode admitir muitas decomposições de caminhos diferentes.

Chamaremos de r_i a *taxa de tráfego* finita e positiva do par s_i, t_i , a qual define a quantidade de fluxo que precisa ir da origem s_i ao destino t_i . Temos que um fluxo f é *factível* se, para todo i , a soma dos fluxos de rotas que pertencem ao conjunto de caminhos de s_i até t_i (\wp_i) é igual à taxa de tráfego r_i :

$$\sum_{P \in \wp_i} f_P = r_i. \quad (3.1)$$

Temos que cada aresta $e \in E$ tem um *custo*, que pode depender da quantidade de fluxo na aresta. Vamos denotar a função de custo na aresta e por $c_e(\cdot)$, que depende da quantidade de tráfego na aresta. Para toda aresta $e \in E$, a função de custo c_e é não-negativa, contínua e não-decrescente (não negativa porque o custo que vamos considerar se refere ao tempo, o qual não pode ser negativo; não faz sentido se não for contínua, pois o aumento é proporcional ao tráfego; não-decrescente pois, se fosse decrescente, chegaria um momento que seria negativa, já que assumimos continuidade (e não faz sentido o custo diminuir com o aumento do tráfego)).

Depois de definir os componentes principais de uma rede, temos que uma instância é um trio da forma (G, r, c) , onde G é a rede, r é o vetor de taxas de

tráfego e c é o vetor de funções de custo.

O *custo de um caminho* P com respeito ao fluxo f é definido como a soma do custo das arestas do caminho:

$$c_P(f) = \sum_{e \in P} c_e(f_e). \quad (3.2)$$

O custo total de um fluxo f em uma instância (G, r, c) é a soma de todos os custos incorridos pelo tráfego:

$$C(f) = \sum_{P \in \varphi} c_P(f) f_P. \quad (3.3)$$

Usando (3.2) e revertendo a ordem do somatório, podemos reescrever (3.3):

$$C(f) = \sum_{e \in E} c_e(f_e) f_e. \quad (3.4)$$

O custo de um fluxo difere do custo médio do tráfego por um fator multiplicativo igual à soma $\sum_i r_i$ de todas as taxas de tráfego.

Um fluxo factível para uma instância (G, r, c) é *ótimo* se ele é minimiza $C(f)$, dado todo o conjunto de fluxos factíveis. Esse fluxo deve existir, já que o espaço de fluxos factíveis é um subconjunto fechado e limitado do espaço Euclidiano e o custo $C(\cdot)$ é uma função contínua. [12]

3.2 Roteamento Egoísta (Selfish Routing)

O tempo de viagem por uma certa rota (caminho) depende diretamente do tamanho do congestionamento, i.e., da quantidade de fluxo que escolhe rotas interferentes. Assume-se que cada motorista obtém a informação necessária para determinar sua rota e que eles o fazem de maneira a alcançar o caminho mais favorável para si, aquele que lhes resultará no menor tempo de viagem possível. Isso quer dizer

que eles escolhem seus caminhos de forma egoísta (selfish routing), sem pensar nas consequências de suas escolhas para os outros motoristas. Formalmente, a definição de roteamento egoísta é dada por escolhas não coordenadas, o que nos leva a concluir que não é surpreendente que o resultado de uma otimização local feita por muitos indivíduos com conflito de interesses não possua nenhum tipo de otimização global. Sendo assim, o comportamento egoísta não produz, necessariamente, um resultado social ótimo [14, 12].

É nesse contexto que surge o preço da anarquia, para medir a pior perda de bem-estar social causada pelo roteamento egoísta, e o objetivo de diminuir cada vez mais essa perda para ir se aproximando do ótimo social (no sentido de Pareto).

3.3 Definição do Paradoxo de Braess

O Paradoxo de Braess, que, na verdade, não é um paradoxo, é apenas um desafio à intuição, é um dos exemplos onde podemos ver claramente o efeito prejudicial do roteamento egoísta. O Paradoxo de Braess mostra que, em situações desfavoráveis, uma extensão (uma nova estrada) na rede rodoviária pode levar a maiores tempos de viagem. Essas situações desfavoráveis são dadas pela existência do que Dietrich Braess chama de um fluxo crítico diferente do fluxo ótimo [14], os quais estudaremos mais profundamente no próximo capítulo, o primeiro com o nome de fluxo de Nash. A aplicação do Paradoxo de Braess que estamos focando nesse trabalho é o tráfego de veículos. Entretanto, existem outras possíveis aplicações, como em redes elétricas (elétrons se comportam como motoristas egoístas (princípio de Fermat)) e em redes de cordas e molas [12].

Chegamos, então, a um questionamento algorítmico: como podemos planejar redes que evitem o paradoxo de Braess? Nos próximos capítulos, vamos tentar,

por diferentes métodos, identificar a ocorrência do paradoxo em uma rede e, então, transformá-la para que ela não tenha mais o risco de ser submetida ao paradoxo.

3.4 Exemplo do artigo de Braess

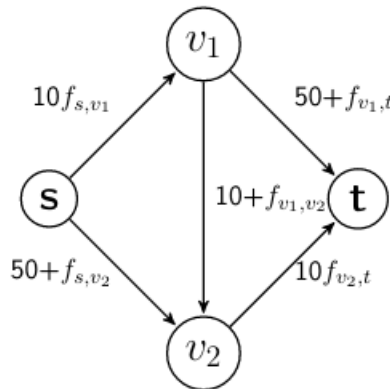


Figura 3.1: Exemplo de ocorrência do Paradoxo de Braess

Para o grafo da figura 3.1, vamos supor que a taxa de fluxo de tráfego que tem que ir de s para t é de 6 motoristas, ou seja, $r = 6$. Podemos observar que a distribuição ótima desse fluxo é ir metade pelo caminho $s \rightarrow v_1 \rightarrow t$ e os outros 3 motoristas por $s \rightarrow v_2 \rightarrow t$. Isso nos dá uma custo total médio de 83 unidades de tempo por indivíduo, $C(f) = 83$. Entretanto, existe um caminho pelo qual o tempo de viagem é menor. Mais especificamente, esse caminho é $s \rightarrow v_1 \rightarrow v_2 \rightarrow t$ e tem tempo de viagem de 70.

Suponha que os veículos estão distribuídos conforme o fluxo ótimo, originando um tempo de 83 para cada motorista. Os motoristas que conhecem os tempos de viagem teriam incentivo a mudar seu caminho para $s \rightarrow v_1 \rightarrow v_2 \rightarrow t$ e quebrariam o equilíbrio. Agora, o fluxo de tráfego vai ser distribuído da maneira que Braess chama

de crítica, o novo equilíbrio será $f_{sv_1t} = f_{sv_2t} = f_{sv_1v_2t} = 2$. Essa redistribuição nos leva a um tempo de 92 para cada indivíduo, $C(f) = 92$, que é maior que o alcançado pela distribuição ótima. Assim, nessa nova configuração, nenhum motorista alcançou um tempo tão bom quanto o ótimo, apesar de terem seguido o que achavam ser melhor para si mesmos, e, o pior, nenhum deles tem incentivo a voltar para o caminho anterior, dados os percursos dos outros [14].

4 TENTATIVA POR EQUILÍBRIO DE NASH

4.1 Fluxo de Nash

Nessa seção, baseada em [12], vamos definir mais especificamente o que significa dizer que um fluxo é um fluxo de Nash (ou fluxo crítico), ou seja, está no equilíbrio de Nash. Essa definição é muito importante para nossas análises, pois são esses equilíbrios que os jogadores alcançam quando visam a otimização do seu tempo, otimização local, e não do ótimo global.

Intuitivamente, um fluxo falha em estar no equilíbrio de Nash se algum tráfego pode diminuir seu custo mudando de caminho, e deve estar no equilíbrio de Nash caso contrário. Assim, um fluxo é um equilíbrio de Nash se, e somente se, sempre que o tráfego troca de caminho, o custo incorrido pelo tráfego só pode aumentar.

Definição 4.1.1. Um fluxo f factível para a instância (G, r, c) está no *equilíbrio de Nash*, ou é um *fluxo de Nash*, se para toda comódite $i \in \{1, \dots, k\}$, caminhos de $s_i - t_i$ $P_1, P_2 \in \wp_i$ com $f_{P_1} > 0$, e todas as quantidades $\delta \in (0, f_{P_1}]$ de tráfego em P_1 ,

$$c_{P_1}(f) \leq c_{P_2}(\tilde{f}),$$

onde o fluxo \tilde{f} é obtido por f movendo δ unidades de fluxo do caminho P_1 para o caminho P_2 :

$$\tilde{f}_P = \begin{cases} f_P - \delta, & \text{se } P = P_1 \\ f_P + \delta, & \text{se } P = P_2 \\ f_P, & \text{se } P \notin \{P_1, P_2\}. \end{cases}$$

Com δ tendendo a 0 e com a continuidade e monotonicidade das funções de custo das arestas, temos a seguinte caracterização do fluxo no equilíbrio de Nash:

Proposição 4.1.1. *Um fluxo f factível para a instância (G, r, c) está no equilíbrio de Nash se, e somente se, para todo $i \in \{1, \dots, k\}$ e $P_1, P_2 \in \wp_i$ com $f_{P_1} > 0$,*

$$c_{P_1}(f) \leq c_{P_2}(f). \quad (4.1)$$

Prova. Primeiro, suponhamos que f é um fluxo factível para a instância (G, r, c) que está no equilíbrio de Nash. Então, vale a definição 4.1.1 para f . Seja \tilde{f} um fluxo obtido movendo-se, no fluxo f , δ de fluxo do caminho P_1 para o caminho P_2 . Fazendo δ tender a 0, teremos que \tilde{f}_P é igual a f_P para todo caminho P . E então $c_{P_1}(f) \leq c_{P_2}(f)$.

Agora, supondo que $\forall i \in \{1, \dots, k\}$ e $P_1, P_2 \in \wp_i$ com $f_{P_1} > 0$, $c_{P_1}(f) \leq c_{P_2}(f)$. Seja \tilde{f} um fluxo factível para a instância (G, r, c) alcançado quando movemos δ unidades de fluxo de P_1 para P_2 . Teremos então, considerando que as funções de custo são contínuas e não-decrescentes, $c_{P_1}(\tilde{f}) \leq c_{P_1}(f)$ e $c_{P_2}(f) \leq c_{P_2}(\tilde{f})$. O que nos leva a $c_{P_1}(f) \leq c_{P_2}(\tilde{f})$, que significa que f está no equilíbrio de Nash. ■

Concluimos então, pela proposição 4.1.1, que, em um fluxo de Nash, todo fluxo viaja pelos caminhos de custo mínimo. Particularmente, se f está no equilíbrio de Nash, então todos os caminhos de fluxo de $s_i - t_i$ - caminhos de $s_i - t_i$ para os quais f tem uma quantidade positiva de fluxo - tem custos iguais.

Corolário 4.1.1.1. *Se f é um fluxo no equilíbrio de Nash para (G, r, c) , então para cada comódite i , todos os caminhos de fluxo de f de $s_i - t_i$ tem um custo em comum $c_i(f)$.*

Prova. Seja f um fluxo no equilíbrio de Nash para (G, r, c) e P_1 e P_2 caminhos com fluxos maiores que zero em f . Assim, a desigualdade do teorema 4.1.1 vale para os dois lados, o que prova que $c_{P_1}(f) = c_{P_2}(f)$. ■

Usando a equação (3.3) para expressar o custo $C(f)$ de um fluxo f no equilíbrio de Nash de um jeito particular, temos:

Proposição 4.1.2. *Seja f um fluxo no equilíbrio de Nash para a instância (G, r, c) . Dado que todos os caminhos de fluxo de f de $s_i - t_i$ tem custo comum $c_i(f)$. Então,*

$$C(f) = \sum_{i=1}^k c_i(f)r_i$$

Prova. Para cada $s_i - t_i$ de uma instância (G, r, c) , temos que o custo total, dados os caminhos que pertencem à comódite i , é igual a $\sum_{P \in \wp_i} c_P(f)f_P$. Mas, pelo corolário 4.1.1.1, sabemos que no equilíbrio todo caminho de fluxo em $s_i - t_i$ é igual. Assim, $c_P(f) = c_i(f)$ para todo $P \in \wp_i$, o que nos dá $\sum_{P \in \wp_i} c_P(f)f_P = c_i(f) \sum_{P \in \wp_i} f_P$. Pela equação (3.1), temos que $\sum_{P \in \wp_i} f_P = r_i$. Logo, chegamos a $C_i(f) = c_i(f)r_i$ para toda comódite i . Como queremos o custo total da instância (G, r, c) , concluímos que ele é igual à soma dos custos de todas as comódites i , i.e., $C(f) = \sum_{i=1}^k c_i(f)r_i$, onde k é a quantidade de comódites da instância (G, r, c) . ■

Agora, vamos enunciar duas proposições que falam sobre a existência e a unicidade, no sentido de custo, dos fluxos de Nash, as quais provaremos mais à frente.

Proposição 4.1.3. *Toda instância (G, r, c) admite, pelo menos, um fluxo de Nash.*

Proposição 4.1.4. *Se f e \tilde{f} são fluxos no equilíbrio de Nash para a instância (G, r, c) , então $C(f) = C(\tilde{f})$.*

4.2 Fluxo Ótimo e Relações com o Fluxo de Nash

4.2.1 Fluxo Ótimo

Como o próprio nome já diz, fluxos ótimos são fluxos factíveis que determinam o menor custo global possível. Para simplificar, assumiremos que as funções de custo são continuamente diferenciáveis, embora análogos dos seguintes resultados também se mantenham sob hipóteses mais fracas.

Então, desejamos encontrar um vetor f que atenda a todas as condições para que seja factível e que determine o menor valor possível para a função custo total, dada por (3.4), a qual caracteriza o bem-estar social. Logo, temos o seguinte problema de programação não-linear:

$$\begin{aligned}
 \text{Min} \quad & \sum_{e \in E} h_e(f_e) \\
 \text{sujeito a:} \quad & \sum_{P \in \varphi_i} f_P = r_i, \quad \forall i \in \{1, \dots, k\} \\
 & f_e = \sum_{P \in \varphi: e \in P} f_P, \quad \forall e \in E \\
 & f_P \geq 0, \quad \forall P \in \varphi,
 \end{aligned} \tag{NLP}$$

com $h_e(f_e) = c_e(f_e)f_e$.

Agora, vamos caracterizar o ótimo de (NLP). Para isso, vamos supor um pouco de convexidade na programação não-linear (mais a frente vamos entender o porquê). Começemos então definindo a convexidade:

Definição 4.2.1.

- (a) Se x e y são dois pontos do espaço Euclidiano \mathbb{R}^n , então uma *combinação*

convexa de x e y é um ponto no segmento de reta entre x e y - um ponto da forma $\lambda x + (1 - \lambda)y$ para algum $\lambda \in [0, 1]$.

- (b) Um subconjunto S de \mathbb{R}^n é *convexo* se ele contém todas as suas combinações convexas: sempre que x, y estiver em S , então $\lambda x + (1 - \lambda)y$ também estará, $\forall \lambda \in [0, 1]$.
- (c) Uma função $h : S \rightarrow \mathbb{R}$ definida em um subconjunto convexo S no espaço Euclidiano é *convexo* se todo segmento de reta entre dois pontos do gráfico de h estiver sobre o gráfico:

$$h(\lambda x + (1 - \lambda)y) \leq \lambda h(x) + (1 - \lambda)h(y) \quad (4.2)$$

para todo $x, y \in S$ e $\lambda \in [0, 1]$.

- (d) Uma função $c : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ é *semi-convexa* se a função $x.c(x)$ for convexa.

Assumindo que as nossas funções de custo das arestas são semi-convexas, temos, pela definição 4.2.1(d), que o custo total $C(\cdot)$ é uma função convexa, já que é a soma de funções convexas, dado por (3.4). Além disso, a região factível de (NLP) é uma intersecção de meio-espacos é então convexa. Assim, o problema de programação não-linear (NLP), é um problema de *programação convexa* quando todas as funções de custo forem semi-convexas.

Vamos caracterizar o ótimo de (NLP) quando a função objetivo é continuamente diferenciável e convexa, independente da formulação específica que damos a h_e . Para isso, vamos usar $h'_e = \frac{d}{dx}h_e(x)$ e $h'_P(f) = \sum_{e \in P} h'_e(f_e)$.

Proposição 4.2.1. *Seja f^* uma solução factível para a programação não-linear da forma (NLP) em que toda função h_e é continuamente diferenciável e convexa. Então, as seguintes afirmações são equivalentes:*

(a) O fluxo f^* é ótimo.

(b) Para todo $i \in \{1, \dots, k\}$ e $P_1, P_2 \in \wp_i$ com $f_{P_1}^* > 0$,

$$h'_{P_1}(f^*) \leq h'_{P_2}(f^*).$$

(c) Para todo fluxo factível f ,

$$\sum_{P \in \wp} h'_P(f^*) f_P^* \leq \sum_{P \in \wp} h'_P(f^*) f_P.$$

(d) Para todo fluxo factível f ,

$$\sum_{e \in E} h'_e(f_e^*) f_e^* \leq \sum_{e \in E} h'_e(f_e^*) f_e.$$

Prova. As afirmações (c) e (d) são equivalentes pela equivalência de (3.3) e (3.4). Para vermos a equivalência de (b) e (c), vamos definir a função H por

$$H(f) = \sum_{P \in \wp} h'_P(f^*) f_P.$$

Como f^* é fixado, $H(f)$ é o custo do fluxo f onde o custo de cada caminho P é a constante independente do congestionamento $h'_P(f^*)$. Deve estar claro que H é precisamente minimizado pelos fluxos factíveis que determinam, para cada i , todo o fluxo da comodite i nos caminhos de $s_i - t_i$ que minimiza $h'_P(f^*)$ por todos esses caminhos. Assim, $H(f^*)$ minimiza H sobre todos os fluxos factíveis, e, consequentemente, alcança a condição (c) se, e somente se, a condição (b) for atendida.

Vamos terminar a prova mostrando que a propriedade (a) implica na propriedade (b) e que a propriedade (d) implica na propriedade (a). Para provar a primeira implicação, suponha que f^* é um fluxo ótimo, e considere um caminho de $s_i - t_i$, $P_1 \in \wp_i$ com $f_{P_1}^* > 0$, e outro caminho de $s_i - t_i$, $P_2 \in \wp_i$. Uma vez que é assumido

que toda função h_e é continuamente diferenciável, transferir uma pequena quantidade de fluxo $\lambda \in (0, f_{P_1}^*)$ de P_1 para P_2 gera um fluxo factível com o valor da função objetivo igual a

$$\sum_{e \notin \{P_1, P_2\}} h_e(f_e^*) + \sum_{e \in P_2} h_e(f_e^* + \lambda) + \sum_{e \in P_1} h_e(f_e^* - \lambda). \quad (4.3)$$

Por expansão em séries de Taylor, temos:

$$\sum_{e \notin \{P_1, P_2\}} h_e(f_e^*) + \sum_{e \in P_2} [h_e(f_e^*) + \lambda h'_e(f_e^*) + \theta(\lambda^2)] + \sum_{e \in P_1} [h_e(f_e^*) - \lambda h'_e(f_e^*) + \theta(\lambda^2)].$$

Mas temos que o valor da função objetivo do fluxo ótimo f^* é dado por $\sum_{e \notin \{P_1, P_2\}} h_e(f_e^*) + \sum_{e \in P_1} h_e(f_e^*) + \sum_{e \in P_2} h_e(f_e^*)$, que é, por definição, menor ou igual ao valor determinado por qualquer outro fluxo factível. Assim:

$$\begin{aligned} & \sum_{e \notin \{P_1, P_2\}} h_e(f_e^*) + \sum_{e \in P_2} [h_e(f_e^*) + \lambda h'_e(f_e^*) + \theta(\lambda^2)] + \sum_{e \in P_1} [h_e(f_e^*) - \lambda h'_e(f_e^*) + \theta(\lambda^2)] \\ & \geq \\ & \sum_{e \notin \{P_1, P_2\}} h_e(f_e^*) + \sum_{e \in P_1} h_e(f_e^*) + \sum_{e \in P_2} h_e(f_e^*) \\ & \Downarrow \\ & \sum_{e \in P_1} h'_e(f_e^*) \leq \sum_{e \in P_2} h'_e(f_e^*) \end{aligned}$$

já que $\theta(\lambda^2)$ desaparece a medida que $\lambda \downarrow 0$.

Concluimos usando a suposição de convexidade para provar que a condição (d) implica na condição (a). Suponha que f^* satisfaz a propriedade (d), e seja f um fluxo factível. Já que toda função h_e é convexa, podemos delimitar inferiormente o valor da função objetivo em f por uma aproximação linear da função objetivo em

f^* (semelhante à expansão em séries de Taylor que fizemos anteriormente):

$$\begin{aligned} \sum_{e \in E} h_e(f_e) &\geq \sum_{e \in E} [h_e(f_e^*) + h'_e(f_e^*)(f_e - f_e^*)] \\ &= \sum_{e \in E} h_e(f_e^*) + \sum_{e \in E} h'_e(f_e^*)(f_e - f_e^*). \end{aligned} \quad (4.4)$$

Por hipótese, f^* satisfaz a propriedade (d) e então o segundo termo do lado direito de (4.4) é não-negativo. Segue-se então que o valor da função objetivo de f é, pelo menos, o de f^* . Como f é arbitrário, temos que f^* é um fluxo ótimo. ■

4.2.2 Função de Custo Marginal

Pode-se observar que há semelhança entre a caracterização das soluções ótimas dos problemas de programação convexa da forma (NLP), dada pela proposição 4.2.1(b), e a caracterização dos fluxos de Nash, dada pela proposição 4.1.1. O que nos leva à seguinte conclusão: *os fluxos ótimo e de Nash são a mesma coisa, apenas com funções de custo diferentes.* Essa conclusão nos leva à definição de função de custo marginal e ao corolário que relaciona os dois tipos de fluxo, ótimo e de Nash:

Definição 4.2.2. Se c é uma função de custo diferenciável, então a *função de custo marginal* c^* correspondente é definida por

$$c^* = \frac{d}{dx}(x \cdot c(x)) = c(x) + x \cdot c'(x). \quad (4.5)$$

As proposições 4.1.1 e 4.2.1(b) levam ao seguinte corolário:

Corolário 4.2.1.1. *Seja (G, r, c) uma instância com funções de custo continuamente diferenciáveis, semi-convexas e e funções de custo marginais c^* . Então, um fluxo factível f para (G, r, c) é ótimo se, e somente se, está no equilíbrio de Nash para a instância (G, r, c^*) .*

Observação. A interpretação da função de custo marginal é a de que ao aumentar levemente a quantidade de fluxo em uma aresta, veremos que a função de custo marginal correspondente c^* terá um termo $c(x)$ capturando o custo experimentado (por unidade) pelo novo tráfego, assim como um segundo termo $x.c'(x)$ representando o aumento de custo (por unidade) incorrido pelo tráfego existente devido ao congestionamento adicional. Essencialmente, a única diferença entre um fluxo ótimo e um fluxo de Nash é que o primeiro está atento a este segundo termo responsável por gerar consciência do quão influente é um aumento no fluxo, por menor que ele seja, para o congestionamento, enquanto este último não o observa.

Exemplo 1.

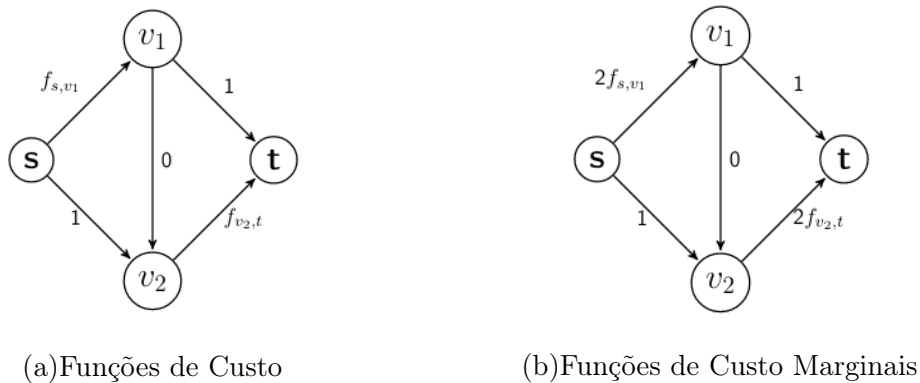


Figura 4.2: Paradoxo de Braess

Considere o exemplo do Paradoxo de Braess depois de adicionar a aresta de custo zero (figura 4.2(a)). Supondo que a taxa de tráfego r é igual a 1, vemos que o fluxo f que roteia todo o tráfego pelo caminho $s \rightarrow v_1 \rightarrow v_2 \rightarrow t$ iguala o custo dos três caminhos em 2, e então, pela proposição 4.1.1, f é um fluxo de Nash com custo $C(f) = 2$. Mudando para as funções de custo marginais, figura 4.2(b), encontramos que o fluxo f^* que roteia metade do tráfego em cada um dos caminhos $s \rightarrow v_1 \rightarrow t$ e $s \rightarrow v_2 \rightarrow t$ iguala os custos marginais dos três caminhos de $s - t$ em 2 e é então, pelo corolário 4.2.1.1, ótimo para o grafo original. Assim, temos que o custo $C(f^*)$

de f^* é $3/2$, e o preço da anarquia (veremos a definição formal na próxima subseção) é $4/3$.

Fato 4.2.1. *Se (G, r, c) for uma instância com funções de custo semi-convexas, então um fluxo ótimo para (G, r, c) pode ser computado, com um erro arbitrariamente pequeno, em tempo polinomial no tamanho da instância e no número de bits de precisão desejado. Ou seja, se o problema (NLP) é de programação convexa, seu fluxo ótimo pode ser encontrado eficientemente.*

O fato 4.2.1 é devido à existência de algoritmos de tempo polinomial para a programação convexa.

A seguir, temos um fato que justifica termos assumido a semi-convexidade para as funções de custo:

Fato 4.2.2. *Se $P \neq NP$, então não existe algoritmo para computar uma aproximação para um fluxo ótimo de uma instância (G, r, c) que sempre rode em tempo polinomial no tamanho da instância e no número de bits de precisão da aproximação (definiremos algoritmos de aproximação no próximo capítulo).*

O fato 4.2.2 é verdadeiro mesmo em redes de arestas paralelas.

4.2.3 Preço da Anarquia

Uma definição rápida para o preço da anarquia é a pior razão possível entre o custo do fluxo no equilíbrio de Nash e o do fluxo ótimo. A definição formal é a seguinte:

Definição 4.2.3.

- (a) Seja (G, r, c) uma instância que admite um fluxo ótimo f^* e um fluxo de Nash f . O *preço da anarquia* de (G, r, c) , denotado por $\rho(G, r, c)$, é definido pela razão

$$\rho(G, r, c) = \frac{C(f)}{C(f^*)}.$$

- (b) Seja Υ o conjunto de instâncias, então o *preço da anarquia* de Υ é

$$\rho(\Upsilon) = \sup_{(G, r, c) \in \Upsilon} \rho(G, r, c).$$

Como sabemos, pela proposição 4.1.4 que todos os fluxos de Nash de uma instância tem custo igual, a razão $\rho(G, r, c)$ da definição 4.2.3(a) está bem definida para $C(f^*) > 0$. Se f^* é um fluxo com custo zero, então $c_P(f^*) = 0$ sempre que $f_P^* > 0$, o que faz de f^* também um fluxo de Nash, pela proposição 4.1.1. Nesse caso, definimos que $\rho(G, r, c)$ é igual a 1.

4.3 Existência e Unicidade

Nessa seção, vamos, finalmente, provar a existência e a unicidade (em essência) do fluxo de Nash. Para isso, vamos explorar a similaridade entre os fluxos de Nash e ótimo, baseada nas proposições 4.1.1 e 4.2.1(b).

Proposição 4.3.1. *Seja (G, r, c) uma instância. Então, o problema de programação não-linear*

$$\begin{aligned} \text{Min} \quad & \sum_{e \in E} h_e(f_e) \\ \text{sujeito a:} \quad & \sum_{P \in \varphi_i} f_P = r_i, \quad \forall i \in \{1, \dots, k\} \\ & f_e = \sum_{P \in \varphi: e \in P} f_P, \quad \forall e \in E \\ & f_P \geq 0, \quad \forall P \in \varphi, \end{aligned} \tag{CP}$$

onde $h_e(x) = \int_0^x c_e(t)dt$, é convexo. Além disso, os fluxos de Nash de (G, r, c) são precisamente o ótimo de (CP).

Prova. No problema de programação não-linear (CP), $h_e(x) = \int_0^x c_e(t)dt$ e, então, temos que $h'_e(x) = c_e(x)$, contínua e não-decrescente, e que cada h_e é uma função convexa e, conseqüentemente, (CP) é uma programação convexa. Pela proposição 4.2.1(b), temos que se um fluxo f^* é ótimo para o problema de programação não-linear, então $h'_{P_1}(f^*) \leq h'_{P_2}(f^*)$, para todo $i \in \{1, \dots, k\}$ e $P_1, P_2 \in \wp_i$ com $f^*_{P_1} > 0$. Assim, em (CP), $c_{P_1}(f^*) \leq c_{P_2}(f^*)$, o que faz de f^* um fluxo de Nash para (G, r, c) , pela proposição 4.1.1. ■

Prova da Proposição 4.1.3 (existência do fluxo de Nash). O problema de programação convexa (CP) tem uma função objetivo contínua e uma região factível fechada (limitada), o que o faz admitir uma solução ótima. Agora, a proposição 4.1.3 segue da proposição 4.3.1. ■

Agora, vamos percorrer o caminho de demonstrar a unicidade dos fluxos de Nash.

Corolário 4.3.1.1. *Se f e \tilde{f} são fluxos de Nash para (G, r, c) , então $c_e(f_e) = c_e(\tilde{f}_e)$ para toda aresta e .*

Prova. Suponha que f e \tilde{f} sejam fluxos de Nash para (G, r, c) , e então são ótimos globais para o problema de programação convexa (CP). Já que a função objetivo de (CP) é convexa, a inequação (4.2) deve valer para toda combinação convexa de f e \tilde{f} . Como f e \tilde{f} são ótimos globais de (CP) e toda combinação convexa deles são factíveis para (CP), (4.2) deve continuar valendo com a igualdade para todas essas combinações convexas. Como toda função h_e em (CP) é convexa, isso pode ocorrer

apenas se toda função h_e for linear entre f_e e \tilde{f}_e . Pela definição de h_e e continuidade das funções de custo c_e , segue-se que toda função c_e é constante entre f_e e \tilde{f}_e . ■

Segue agora a demonstração da proposição 4.1.4, que estabelece que todos os fluxos de Nash tem custo igual.

Prova da Proposição 4.1.4 (unicidade do fluxo de Nash). A proposição segue diretamente da proposição 4.1.2 e do corolário 4.3.1.1. ■

Corolário 4.3.1.2. *Se f é um fluxo de Nash para (G, r, c) e \tilde{f} é um fluxo factível para (G, r, c) que induz o mesmo fluxo que f nas arestas então \tilde{f} também está no equilíbrio de Nash para (G, r, c) .*

Prova. A função objetivo para o problema de programação convexa (CP) na proposição 4.3.1 depende apenas do fluxo induzido nas arestas. ■

Corolário 4.3.1.3. *Se f e \tilde{f} são fluxos de Nash para (G, r, c) , então $f_e = \tilde{f}_e$ sempre que c_e for estritamente crescente.*

Prova. Imediatamente pelo corolário 4.3.1.1. ■

Observação. *Os corolários 4.3.1.1 e 4.3.1.3 são, de certo modo, os melhores resultados possíveis sobre a unicidade dos fluxos de Nash. Considerando uma rede de dois nós e duas arestas em que ambas as arestas tem função de custo $c(x) = 1$, é fácil de ver que em redes com funções de custo constantes, diferentes fluxos de Nash podem rotear diferentes quantidades de fluxo em uma aresta. Mesmo quando todas as funções de custo são estritamente crescentes, o corolário 4.3.1.2 mostra que podem existir muitos fluxos de Nash nos caminhos. Obviamente, o corolário 4.3.1.3 prova que todos esses fluxos de Nash devem induzir o mesmo fluxo nas arestas.*

Corolário 4.3.1.4. *Um fluxo f factível para (G, r, c) está no equilíbrio de Nash se, e somente se,*

$$\sum_{e \in E} c_e(f_e) f_e \leq \sum_{e \in E} c_e(f_e) \tilde{f}_e$$

para todo fluxo \tilde{f} factível para (G, r, c) .

Prova. Imediato pelas proposições 4.2.1 e 4.3.1. ■

No último corolário dessa seção, mostramos que os fluxos de Nash podem ser calculados eficientemente.

Corolário 4.3.1.5. *Se (G, r, c) é uma instância, então um fluxo de Nash para (G, r, c) pode ser computado, com um termo de erro pequeno, em tempo polinomial no tamanho da instância e no número de bits de precisão desejado.*

Prova. Segue pela proposição 4.3.1 e pelo fato de que o problema de programação convexa (CP) pode ser solucionado em tempo polinomial (fato 4.2.1). ■

4.4 Relação com o Paradoxo de Braess

Depois de todo esse estudo e análise feito em torno dos fluxos ótimo e de Nash, podemos pensar no seguinte método:

Como existe um modo eficiente de encontrar os fluxos ótimo e de Nash para uma dada instância (G, r, c) , baseado nos problemas de programação linear (NLP) e (CP), considerando que as funções de custo nas arestas são semi-convexas, então podemos aplicá-lo à (G, r, c) , encontrar os dois fluxos e compará-los. Caso exista uma aresta que tem fluxo igual a zero no fluxo ótimo e fluxo maior que zero no fluxo de Nash, chegaremos à conclusão de que há o paradoxo de Braess nessa instância e que uma

de suas causas é essa aresta. Então, retiramos-a da instância e teremos, assim, uma rede com novo fluxo de Nash mais eficiente, mais próximo do fluxo ótimo.

O teorema a seguir prova a veracidade desse método.

Proposição 4.4.1. *Seja \tilde{f} um fluxo de Nash do grafo G e seja f^* um fluxo ótimo social para G tais que $C(f^*) < C(\tilde{f})$. Se existir uma aresta j tal que $\tilde{f}_j \neq 0$ mas $f_j^* = 0$, onde f_j representa o fluxo na aresta j , então existe um subgrafo $H \subset G$ com fluxo de Nash \tilde{f}_H tal que $C(\tilde{f}_H) < C(\tilde{f})$ [3].*

Prova. Seja H um subgrafo de G que contém apenas arestas com fluxo diferente de zero em f^* . Então, ele é um subgrafo estrito de G , porque a aresta j no grafo G tem custo diferente de zero no fluxo \tilde{f} mas zero no fluxo f^* , o que implica que a aresta j não pertence a H mas pertence a G . Queremos encontrar que $C(\tilde{f}_H) < C(\tilde{f})$. Suponhamos, por absurdo, que não. Considere \tilde{f}_H com as funções de custo marginais. Suponha que exista um caminho k tal que $c_k^*(\tilde{f}_H) > c_{P_j}^*(\tilde{f}_H)$, onde o fluxo de \tilde{f}_H pelo caminho k é diferente de zero, e P_j é algum caminho que contém a aresta j . Então, isso implica que o custo social total pode ser reduzido quando o fluxo pela aresta j for diferente de zero, uma contradição. ■

Exemplo 2.

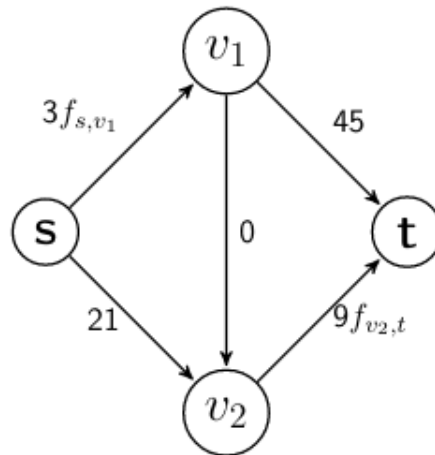


Figura 4.3: Exemplo do método sugerido

Supondo que $r = 8$, vamos encontrar os fluxos ótimos e de Nash e a quantidade de fluxo em cada aresta nesse exemplo [9]. Para isso, utilizamos o pacote "*minimize*" da biblioteca "*scipy.optimize*" da linguagem de programação Python.

Primeiro, vamos encontrar os custos dos fluxos ótimo e de Nash para o grafo da figura 4.3 e ver se tem alguma aresta que o fluxo de Nash usa e o fluxo ótimo não usa.

```

""" Braess-com atalho """
def custo_pareto(e, sign=1.0):
    return 3*e[0]*e[0]+e[1]*45+e[2]*21+e[3]*e[3]*9

def custo_nash(e, sign=1.0):
    return 3*e[0]*e[0]/2+e[1]*45+e[2]*21+9*e[3]*e[3]/2

cons = ( {'type': 'eq',
         'fun' : lambda e: np.array([e[0] + e[2] - 8])},
        {'type': 'eq',
         'fun' : lambda e: np.array([e[0] - e[1] - e[4]])},
        {'type': 'eq',
         'fun' : lambda e: np.array([e[2] - e[3] + e[4]])},
        {'type': 'ineq',
         'fun' : lambda e: e})

res_pareto = minimize(custo_pareto, [8, 8, 0, 0, 0], constraints=cons, jac = None,
                     method='SLSQP', options={'disp': False})
res_nash   = minimize(custo_nash, [8, 8, 0, 0, 0], constraints=cons, jac = None,
                     method='SLSQP', options={'disp': False})

print('pareto', np.round(res_pareto.x,3), np.round(res_pareto.fun,3)/8)
print('nash', np.round(res_nash.x,3), np.round(custo_pareto(res_nash.x),3)/8)

('pareto', array([ 5.,  5.,  3.,  3.,  0.]), 55.5)
('nash', array([ 7.,  3.,  1.,  5.,  4.]), 66.0)

```

Figura 4.4: Código para encontrar os fluxos e o custo total do grafo da figura 4.3

Como podemos observar no resultado do grafo da figura 4.3, dado pela figura 4.4, o fluxo ótimo não usa a aresta (v_1, v_2) , enquanto o fluxo de Nash usa. Assim, vamos retirá-la e rodar o programa de novo para a mesma quantidade de fluxo $r = 8$.

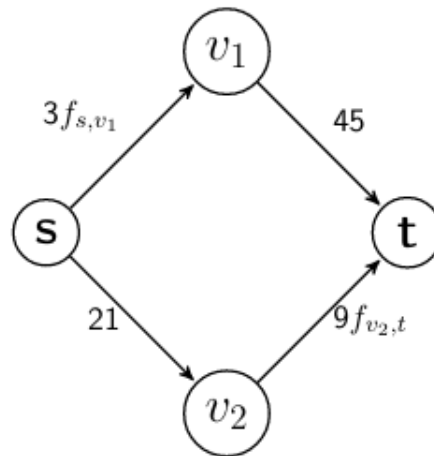


Figura 4.5: Exemplo do método sugerido sem aresta

```

""" Braess-sem atalho """
def custo_pareto(e, sign=1.0):
    return 3*e[0]*e[0]+e[1]*45+e[2]*21+e[3]*e[3]*9

def custo_nash(e, sign=1.0):
    return 3*e[0]*e[0]/2+e[1]*45+e[2]*21+9*e[3]*e[3]/2

cons = ( {'type': 'eq',
          'fun' : lambda e: np.array([e[0] + e[2] - 8])},
         {'type': 'eq',
          'fun' : lambda e: np.array([e[0] - e[1]])},
         {'type': 'eq',
          'fun' : lambda e: np.array([e[2] - e[3]])},
         {'type': 'ineq',
          'fun' : lambda e: e})

res_pareto = minimize(custo_pareto, [4, 4, 4, 4], constraints=cons, jac = None,
                     method='SLSQP', options={'disp': True})
res_nash   = minimize(custo_nash, [4, 4, 4, 4], constraints=cons, jac = None,
                     method='SLSQP', options={'disp': True})

print('pareto', np.round(res_pareto.x,3), np.round(res_pareto.fun,3)/8)
print('nash', np.round(res_nash.x,3), np.round(custo_pareto(res_nash.x),3)/8)

('pareto', array([ 5.,  5.,  3.,  3.]), 55.5)
('nash', array([ 4.,  4.,  4.,  4.]), 57.0)

```

Figura 4.6: Código para encontrar os fluxos e custo total do grafo da figura 4.5

Pelo novo resultado, dado pela figura 4.6, pode-se observar que a ineficiência diminuiu, o custo total do fluxo de Nash passou de 66 para 57.

Entretanto, existem alguns pontos para os quais devemos estar atentos ao usar esse método:

- Não necessariamente o fluxo ótimo deixará de usar alguma aresta;
- Fluxo de Nash é o fluxo dos egoístas, portanto não pode ser comparado diretamente com o fluxo ótimo no sentido de que não é verdade que se o fluxo ótimo não deixa de usar alguma aresta, então não existe alguma aresta que possa ser tirada para que o fluxo de Nash melhore;
- Como olhamos para o fluxo de Nash como aquele que os jogadores alcançarão em uma rede rodoviária, dado que o sistema adotado é o de não-cooperação, temos que encontrar um método que olhe para a minimização do fluxo dos egoístas (Nash). O que temos que nos perguntar é: tirando alguma aresta, vamos melhorar o custo do fluxo de Nash?
- Esse método subestima a quantidade de ocorrências de paradoxo de Braess, mas, caso ele encontre, não estará errado, realmente terá encontrado alguma ocorrência.

Com isso, precisamos de um método mais geral, que funcione também para os outros casos e não subestime a ocorrência do Paradoxo. Isso nos leva ao próximo capítulo, onde vamos explorar o uso de algoritmos de aproximação.

5 TENTATIVA POR ALGORITMOS DE APROXIMAÇÃO

No final do último capítulo, vimos que precisamos de um método mais geral para encontrar a ocorrência do paradoxo de Braess em uma rede, que não necessariamente tem alguma aresta que não seja usada no fluxo ótimo, e tornar o fluxo de Nash o mais próximo do fluxo ótimo possível, diminuindo, assim, o preço da anarquia. Para isso, vamos descrever o problema de um jeito mais detalhado e o chamaremos de Problema de Projetar Rede (*Network Design Problem*), pelo qual é um bom modo para definir a ocorrência do Paradoxo de Braess em uma rede [12].

Definição 5.0.1 (*Network Design Problem*). Dada uma instância (G, r, c) , encontrar eficientemente um subgrafo H de G que minimize o custo comum de todo o tráfego em um fluxo de Nash para (H, r, c) , denotado por $d(H, r, c)$ - bem definido pelas proposições 4.1.1 e 4.3.1.

Assim, se esse subgrafo H existir para a instância (G, r, c) , então ocorre nela o Paradoxo de Braess. Vamos focar as análises em instâncias de comódite única (um par origem-destino).

Podemos observar que esse problema é não-trivial, já que, não necessariamente, o grafo G inteiro vai ser uma solução ótima para ele. Além disso, o número de subgrafos H de G é, em geral, exponencial no tamanho da instância (G, r, c) , ou seja, esse problema não pode ser resolvido por força bruta (removendo cada aresta, calculando o custo comum do equilíbrio de Nash e o comparando com o do equilíbrio de Nash original).

Vamos, então, tentar olhar para a resolução desse problema através de algo-

ritmos de aproximação.

5.1 O que é um Algoritmo de Aproximação?

Algoritmos de aproximação são usados para encontrar soluções aproximadas para problemas de otimização.

Definição 5.1.1. Um *algoritmo de c -aproximação* para um problema de minimização roda em tempo polinomial e retorna uma solução não mais que c vezes tão custosa (ou menos custosa, a depender da situação) quanto uma solução ótima. O valor c é a *razão de aproximação* ou a *performance garantida* do algoritmo. Assim, seja A um algoritmo de aproximação e $f(x)$ o custo de uma aproximação $A(x)$ para a instância x , temos que $f(x)$ não vai ser mais (ou menos) que um fator c vezes o valor de uma solução ótima OPT :

$$\begin{cases} OPT \leq f(x) \leq cOPT, & \text{se } c > 1; \\ cOPT \leq f(x) \leq OPT, & \text{se } c < 1. \end{cases}$$

Acredita-se que problemas NP-difíceis não admitem algoritmos de tempo polinomial, como vimos nas preliminares, mas muitos desses problemas admitem algoritmos de c -aproximação para valores pequenos de c . Por outro lado, existem também muitas *inaproximações* ou *resultados duros* para problemas de otimização NP-difíceis, que não admitem algoritmos de c -aproximação para valores razoáveis de c , assumindo $P \neq NP$.

Exemplo 3 (Vertex Cover). Uma cobertura de vértices (*vertex cover*) de um grafo é um conjunto de vértices tal que cada aresta do grafo é incidente em, pelo menos, um vértice do conjunto. O problema de encontrar a cobertura de vértices mínima é um exemplo de um problema de otimização NP-difícil que admite um algoritmo de aproximação. Temos um algoritmo de 2-aproximação para esse problema: dado

o conjunto E de arestas, retiramos dele uma aresta (u, v) e colocamos os vértices correspondentes a ela no conjunto C e removemos de E as arestas incidentes em u ou em v , até que não tenha mais aresta nenhuma em E .

Algoritmo 1: Algoritmo de 2-aproximação do Vertex Cover para $G = (V, E)$

```

 $C = \emptyset;$ 
 $E' = G.E;$ 
while  $E' \neq \emptyset$  do
    | seja  $(u, v)$  uma aresta arbitrária de  $E'$ ;
    |  $C = C \cup \{u, v\};$ 
    | remova de  $E'$  todas as arestas incidentes em  $u$  ou em  $v$ 
end while
return  $C$ 

```

5.2 Redes com Funções de Custo Afins ($ax + b$)

Consideraremos neste capítulo redes em que a função de custo de cada aresta e é afim, ou seja, linear no congestionamento da aresta mais uma constante ($c_e(x) = a_e x + b_e$ para algum $a_e, b_e \geq 0$, onde x é o congestionamento da aresta), e as analisaremos com o objetivo de chegar a alguma conclusão a respeito de um algoritmo de aproximação para resolver o problema 5.0.1.

5.2.1 O Preço da Anarquia

Primeiramente, vamos ver como algumas caracterizações dadas anteriormente se mostram quando as funções de custo das arestas obedecem a funções afins. O custo total $C(f)$ de um fluxo f , dado pela equação (3.4), se apresenta da seguinte forma:

$$C(f) = \sum_{e \in E} a_e f_e^2 + b_e f_e$$

. Como $a_e \geq 0$, essa é uma função quadrática convexa e o problema de programação não-linear (NLP) é convexo.

Agora, vamos ver como ficam as proposições 4.1.1 e 4.2.1.

Lema 5.2.1. *Seja (G, r, c) uma instância com funções de custo afins $c_e(x) = a_e x + b_e$ para cada aresta $e \in E$. Então,*

(a) *um fluxo factível f está no equilíbrio de Nash para (G, r, c) se, e somente se, para cada comódite i e caminhos de $s_i - t_i$ $P_1, P_2 \in \wp_i$ com $f_{P_1} > 0$,*

$$\sum_{e \in P_1} a_e f_e + b_e \leq \sum_{e \in P_2} a_e f_e + b_e; \quad (5.1)$$

(b) *um fluxo factível f^* é ótimo para (G, r, c) se, e somente se, para cada comódite i e caminhos de $s_i - t_i$ $P_1, P_2 \in \wp_i$ com $f_{P_1}^* > 0$,*

$$\sum_{e \in P_1} 2a_e f_e^* + b_e \leq \sum_{e \in P_2} 2a_e f_e^* + b_e; \quad (5.2)$$

Com isso, temos o seguinte resultado para redes com funções de custo lineares (ax):

Corolário 5.2.1.1. *Se (G, r, c) é uma instância em que cada função de custo de aresta c_e é linear, i.e., da forma $c_e(x) = a_e x$, então um fluxo factível para (G, r, c) é ótimo se, e somente se, é um fluxo de Nash.*

Prova. Quando $b_e = 0$ para toda aresta e , a inequação (5.1) se mantém se, e somente se, a inequação (5.2) se mantém. ■

Lema 5.2.2. *Suponha que (G, r, c) tenha funções de custo afins e f é um fluxo no equilíbrio de Nash. Então,*

(a) $c_e^*(f_e/2) = c_e(f_e)$ para toda aresta e ;

(b) o fluxo $f/2$ é ótimo para $(G, r/2, c)$.

Prova. A parte (a) segue do fato de que todas as funções de custo são afins (lembrando que c_e^* é a função de custo marginal da aresta e dada por $c_e^*(x) = c(x) + xc'(x)$). Para a parte (b), note que se f satisfaz as condições do lema 5.2.1(a) para (G, r, c) , então $f/2$ satisfaz as condições do lema 5.2.1(b) para $(G, r/2, c)$. ■

Agora, vamos percorrer o caminho da prova do limite superior do preço da anarquia para redes com funções de custo afins.

Lema 5.2.3. *Seja f um fluxo para uma instância (G, r, c) com funções de custo afins. Então,*

$$C(f/2) \geq \frac{1}{4} \cdot C(f).$$

Prova. Temos que $c_e(x) = a_e x + b_e$ para cada aresta e . Então,

$$\begin{aligned} C(f/2) &= \sum_{e \in E} \left(\frac{1}{2} a_e f_e + b_e \right) \frac{f_e}{2} \\ &\geq \frac{1}{4} \sum_{e \in E} (a_e f_e + b_e) f_e \\ &= \frac{1}{4} \cdot C(f). \end{aligned}$$

■

O próximo lema

Lema 5.2.4. *Seja (G, r, c) uma instância com funções de custo afins e f^* um fluxo ótimo. Então para todo $\delta > 0$, um fluxo factível para a instância $(G, (1 + \delta)r, c)$ tem*

custo pelo menos

$$C(f^*) + \delta \sum_{e \in E} c_e^*(f_e^*) f_e^*.$$

Prova. Fixe uma instância (G, r, c) com funções de custo afins, um fluxo ótimo f^* e um valor para o parâmetro $\delta > 0$. Supondo que f seja um fluxo factível para $(G, (1 + \delta)r, c)$. A quantidade de fluxo f_e na aresta e pode ser maior ou menor que f_e^* . Para toda aresta $e \in E$, a convexidade da função $x.c_e(x) = a_e x^2 + b_e x$ implica que podemos obter um limite inferior dessa função através de uma aproximação linear no ponto f_e^* :

$$c_e(f_e) f_e \geq c_e(f_e^*) f_e^* + (f_e - f_e^*) c_e^*(f_e^*) \quad (5.3)$$

A inequação (5.3) estabelece que estimando o custo de mudar o valor do fluxo na aresta e de f_e^* para f_e por $(f_e - f_e^*) c_e^*(f_e^*)$ - pelo custo marginal do fluxo aumentado em f_e^* vezes o tamanho da perturbação - pode apenas subestimar o custo de um aumento (quando $f_e > f_e^*$) e sobrestimar o benefício de uma redução (quando $f_e < f_e^*$). Temos então

$$\begin{aligned} C(f) &= \sum_{e \in E} c_e(f_e) f_e \\ &\geq C(f^*) + \sum_{e \in E} (f_e - f_e^*) c_e^*(f_e^*). \end{aligned} \quad (5.4)$$

Como f^* é ótimo para (G, r, c) e as funções de custo afins são semi-convexas, aplicamos a proposição 4.2.1(d) ao fluxo $f/(1 + \delta)$, alcançando

$$\frac{1}{1 + \delta} \sum_{e \in E} c_e^*(f_e^*) f_e \geq \sum_{e \in E} c_e^*(f_e^*) f_e^*,$$

que, junto com a equação (5.4), implica no lema. ■

Agora, o principal teorema dessa subseção segue facilmente dos lemas 5.2.3 e 5.2.4 (lembrando que $\rho(G, r, c)$ é o preço da anarquia da instância (G, r, c) - definição 4.2.3).

Proposição 5.2.1. *Se (G, r, c) tem funções de custo afins, então*

$$\rho(G, r, c) \leq \frac{4}{3}.$$

Prova. Seja f um fluxo no equilíbrio de Nash para (G, r, c) . Pelo lema 5.2.2, $f/2$ é um fluxo ótimo para a instância $(G, r/2, c)$ com $c_e^*(f_e/2) = c_e(f_e)$ para cada aresta e . Aplicando o lema 5.2.4 com $\delta = 1$ e então o lema 5.2.3, encontramos que se f^* é factível para (G, r, c) , então

$$\begin{aligned} C(f^*) &\geq C(f/2) + \sum_{e \in E} c_e^*(f_e/2) \frac{f_e}{2} \\ &\geq \frac{1}{4} \cdot C(f) + \frac{1}{2} \sum_{e \in E} c_e(f_e) f_e \\ &= \frac{3}{4} \cdot C(f). \end{aligned}$$

Assim

$$\frac{C(f)}{C(f^*)} \leq \frac{4}{3},$$

que prova o teorema, pois, para todo f^* factível para (G, r, c) , essa desigualdade é válida, e então, em particular, é válida quando f^* é o fluxo ótimo de (G, r, c) . ■

5.2.2 Limitando Superiormente o Paradoxo de Braess

Vamos provar os limites superiores no pior caso possível do Paradoxo de Braess - dado pelo maior fator pelo qual remover arestas de uma rede pode diminuir o custo incorrido pelo tráfego em um fluxo de Nash - para redes com funções de custo afins.

Corolário 5.2.1. *Se (G, r, c) é uma instância de comódite única com funções de custo afins e H é um subgrafo de G , então*

$$d(G, r, c) \leq \frac{4}{3} \cdot d(H, r, c).$$

Prova. Seja (G, r, c) uma instância de comódite única com funções de custo afins e H um subgrafo de G . Sejam f e f^* fluxos no equilíbrio de Nash para (G, r, c) e (H, r, c) , respectivamente. Pela proposição 4.1.2, podemos escrever $C(f) = r.d(G, r, c)$ e $C(f^*) = r.d(H, r, c)$. Como f^* pode ser visto como um fluxo factível para (G, r, c) , o teorema 5.2.1 implica que $C(f) \leq 4C(f^*)/3$ e então $d(G, r, c) \leq 4d(H, r, c)/3$ ■

5.2.3 Detectando o Paradoxo de Braess

Depois de toda a análise do Paradoxo de Braess nas redes com funções de custo afins, vamos tentar projetar algoritmos para detectá-lo e para resolver o *network design problem* (5.0.1) nessas redes.

Provar os limites do pior caso possível do Paradoxo de Braess é equivalente a provar os limites da razão de aproximação do algoritmo trivial para o *network design problem* - algoritmo que, dada a instância (G, r, c) , retorna o grafo completo G como uma solução. Assim, a subseção anterior tem o seguinte corolário:

Corolário 5.2.2. *A razão de aproximação do algoritmo trivial para o network design problem com funções de custo afins é precisamente $4/3$.*

Com isso, para que um algoritmo de aproximação tenha qualquer credibilidade, ele deve melhorar a performance garantida pelo algoritmo trivial.

Mas nos deparamos com um resultado duro, que prova a não-aproximabilidade de algoritmos.

Proposição 5.2.2. *Assumindo $P \neq NP$, para todo $\epsilon > 0$, não existe $(\frac{4}{3} - \epsilon)$ -algoritmo de aproximação para o network design problem com funções de custo afins.*

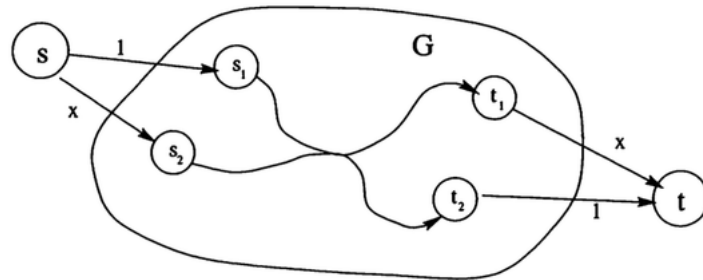


Figura 5.1: Prova da Proposição 5.2.2. Em uma "não"instância de 2DDP, a existência de caminhos $s_1 - t_1$ e $s_2 - t_2$ implica na existência de um caminho $s_2 - t_1$.

Prova. Como vimos nas preliminares, podemos mostrar que um problema é NP-difícil reduzindo a ele um problema que já sabemos ser NP-completo. Aqui, a redução será feita a partir do problema NP-completo 2 CAMINHOS DIRECIONADOS DISJUNTOS (2DDP), cuja prova de ser NP-completo está em [13]: dado um grafo direcionado $G = (V, E)$ e vértices distintos $s_1, s_2, t_1, t_2 \in V$, existem caminhos P_i para $i = 1, 2$ de $s_i - t_i$, tais que P_1 e P_2 são vértice-disjuntos? Assim, vamos mostrar que um $(\frac{4}{3} - \epsilon)$ -algoritmo de aproximação para o *network design problem* com funções de custo afins pode ser usado para diferenciar entre "sim" e "não"instância do problema 2DDP em tempo polinomial.

Considere uma instância Υ do problema 2DDP, como acima. Aumente o conjunto de vértices V adicionando uma origem s e um sumidouro t , e inclua as arestas direcionadas $(s, s_1), (s, s_2), (t_1, t)$ e (t_2, t) (veja a figura 5.1). Denote a nova rede por $G' = (V', E')$ e determine que as arestas de E' têm as seguintes funções de custo c : arestas de E têm funções de custo $c(x) = 0$, arestas (s, s_2) e (t_1, t) têm funções de custo $c(x) = x$ e arestas (s, s_1) e (t_2, t) têm funções de custo $c(x) = 1$. A instância $(G', 1, c)$ pode ser construída a partir de Υ em tempo polinomial.

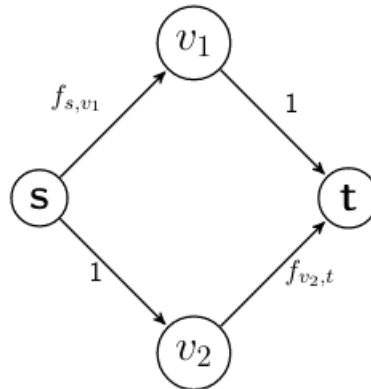


Figura 5.2: Sem Aresta Extra

Para completar a prova, basta demonstrar as duas afirmações:

- (i) Se Υ é uma instância "sim" do problema 2DDP, então G' admite um subgrafo H de G' com $d(H, 1, c) = 3/2$.
- (ii) Se Υ é uma instância "não" do problema 2DDP, então $d(H, 1, c) \geq 2$ para todo subgrafo H de G' .

Para provar (i), seja P_1 e P_2 caminhos vértice-disjuntos $s_1 - t_1$ e $s_2 - t_2$ em G , respectivamente, e obtenha H deletando todas as arestas de G que não estão em algum P_i . Então, H é um subgrafo de G' com exatamente dois caminhos $s - t$, e roteando metade de uma unidade de fluxo por cada um gera um fluxo no equilíbrio de Nash em que cada caminho tem custo $3/2$ (figura 5.2).

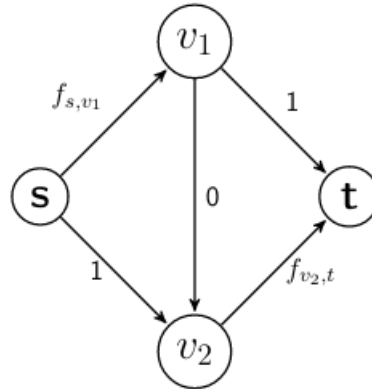


Figura 5.3: Com Aresta Extra

Para (ii), podemos assumir que H contém um caminho $s - t$. Se H tiver um caminho P de $s - t$ contendo um caminho $s_2 - t_1$, então rotear todo o fluxo em P gera um fluxo de Nash em que todo caminho $s - t$ tem custo 2 (figura 5.3). Então, $d(H) = 2$ para todo subgrafo H . Por outro lado, se Υ é uma instância "não" para o problema 2DDP, ainda não restam duas possibilidades: ou para precisamente um $i \in \{1, 2\}$, H tem um caminho P de $s - t$ contendo um caminho $s_i - t_i$, ou todos os caminhos P de $s - t$ em H contêm um caminho de $s_1 - t_2$ de G . Em ambos os casos, rotear um unidade de fluxo por um caminho P fornece um fluxo no equilíbrio de Nash mostrando que $d(H) = 2$. ■

O corolário 5.2.1 e o teorema 5.2.2 mostram que a eficiência na detecção do Paradoxo de Braess em redes com funções de custo afins é impossível, mesmo em instâncias que sofrem com as manifestações mais severas do paradoxo. Mais precisamente, seja uma instância (G, r, c) com funções de custo afins *livre do paradoxo* se $d(G, r, c) \leq d(H, r, c)$ para todo subgrafo H de G e *sujeita ao paradoxo* se para algum subgrafo H de G , $d(G, r, c) = 4d(H, r, c)/3$. Instâncias *livres do paradoxo* não sofrem do Paradoxo de Braess e, pelo corolário 5.2.1, instâncias *sujeitas ao paradoxo*

são precisamente aquelas com a pior perda possível na performance da rede devido ao Paradoxo de Braess. A redução feita na prova do teorema 5.2.2 nos dá o corolário a seguir.

Corolário 5.2.2.1. *Dada uma instância (G, r, c) que tem funções de custo afins e é ou livre do paradoxo ou sujeita ao paradoxo, é NP-difícil decidir se (G, r, c) é ou não sujeita ao paradoxo.*

5.3 O Mar de *there is no*

Tendo percorrido todo o caminho para chegar à dura conclusão da seção anterior para redes com funções de custo afins, temos conclusões duras semelhantes para a tentativa de solucionar o *network design problem*(5.0.1) para redes com outras funções de custo, cujos caminhos para as demonstrações podem ser encontradas no capítulo 5 de [12].

Proposição 5.3.1. *Assumindo $P \neq NP$, para todo $\epsilon > 0$, não existe $(\lfloor n/2 \rfloor - \epsilon)$ -algoritmo de aproximação para o *network design problem* com funções de custo arbitrárias, onde n é a quantidade de nós da rede.*

Proposição 5.3.2. *Assumindo $P \neq NP$, não existe $o(p/\ln p)$ -algoritmo de aproximação para o *network design problem* com funções de custo polinomiais de grau, no máximo, p e coeficientes não-negativos.*

Corolário 5.3.2.1. *Assumindo $P \neq NP$, para todo $\epsilon > 0$, não existe $(O(n^{1-\epsilon}))$ -algoritmo de aproximação para o *network design problem* com funções de custo arbitrárias, onde n é a quantidade de nós da rede.*

Proposição 5.3.3. *Assumindo $P \neq NP$, não existe $(o(\gamma))$ -algoritmo de aproximação para o *network design problem* com funções de custo com inclinação, no máximo, γ .*

Com isso, concluímos que a dificuldade de encontrar algoritmos não-triviais para o *network design problem*(5.0.1) está intrínseca, em vez de ser apenas para um certo conjunto de funções de custo permitidas. Então, temos a seguinte generalização das más notícias mostradas:

assumindo $P \neq NP$, nenhum algoritmo de tempo polinomial tem performance melhor que o algoritmo trivial.

Obviamente, todas as más notícias são no sentido do pior caso possível de algoritmos de aproximação. Isso quer dizer que, enquanto algoritmos de tempo polinomial podem fazer muito melhor que o algoritmo trivial em muitas instâncias, também vão existir instâncias em que eles serão tão ruins quanto o algoritmo trivial.

6 CONCLUSÃO

O objetivo primeiro desse trabalho era analisar o Paradoxo de Braess e encontrar locais do Rio de Janeiro em que ele ocorra. Ocorreu uma subestimação do problema e o trabalho acabou se transformando em um estudo geral dele, olhando para toda a complexidade que gira em torno da tentativa de encontrá-lo. Por outro lado, todo esse estudo é a base ideal para um futuro olhar mais atento na escolha de quais redes analisar para encontrar uma ocorrência do Paradoxo e, enfim, lidar com o egoísmo inerente à todo motorista.

REFERÊNCIAS

- [1] ANTONIO LIMA, S. Çolak; GONZÁLEZ, M. C. Understanding congested travel in urban areas. **Nature Communications**, [S.l.], 2016.
- [2] BARR, N. **Economics of the Welfare State**. 5th.ed. [S.l.]: Oxford University Press, 2012. 46p.
- [3] CHOTRAS, P. Random Simulations of Braess's Paradox. **Arizona State University, School of Mathematical and Statistical Sciences**, [S.l.], 2015.
- [4] FORTNOW, L. The Status of the P Versus NP Problem. **Communications of the ACM**, [S.l.], v.52, p.78–86, 2009.
- [5] HARDESTY, L. Explained: p vs. np. **MIT News Office**, [S.l.], 2009.
- [6] HOLT, C. A.; ROTH, A. E. The Nash Equilibrium: a perspective. **Proceedings of the National Academy of Sciences of the United States of America**, [S.l.], 2004.
- [7] HYEJIN YOUN, M. T. G.; HAWOONG, J. Price of Anarchy in Transportation Networks: efficient and optimality control. **Physical review letters**, [S.l.], 2008.
- [8] JOHN F. NASH, J. Equilibrium Points in n-Person Games. **Proceedings of the National Academy of Sciences of the United States of America**, [S.l.], p.48–49, 1950.
- [9] JOHN MORGANA, H. O.; SEFTON, M. Network Architecture and Traffic Flows: experiments on the pigou-knight-downs and braess paradoxes. **Haas School of Business and Department of Economics, University of California, Berkeley**, [S.l.], 2007.

- [10] KOLATA, G. What if They Closed 42d Street and Nobody Noticed? **The New York Times**, [S.l.], 1990.
- [11] R.C.S. SCHOUERY O. LEE, F. M.; XAVIER, E. **Tópicos de Teoria dos Jogos em Computação**. Rio de Janeiro, Brasil: Instituto Nacional de Matemática Pura e Aplicada, 2015.
- [12] ROUGHGARDEN, T. **Selfish Routing and the Price of Anarchy**. Cambridge, Massachusetts: The MIT Press, 2005.
- [13] S. FORTUNE, J. H.; WYLLIE, J. The Directed Subgraph Homeomorphism Problem. **Theoretical Computer Science**, **10(2):111-121**, [S.l.], 1980.
- [14] VON D. BRAESS, M. On a Paradox of Traffic Planning. **Faculty of Mathematics, Ruhr University Bochum**, [S.l.], 1968.
- [15] VON NEUMANN, J.; MORGENSTERN, O. **The Theory of Games and Economic Behaviour**. Princeton: Princeton University Press, 1943.