

Tópicos da Teoria dos Jogos em Computação

Publicações Matemáticas

Tópicos da Teoria dos Jogos em Computação

R. C. S. Schouery
UNICAMP

O. Lee
UNICAMP

F. K. Miyazawa
UNICAMP

E. C. Xavier
UNICAMP



30^o Colóquio Brasileiro de Matemática

Copyright © 2015 by R. C. S. Schouery, O. Lee, F. K. Miyazawa e E. C. Xavier

Impresso no Brasil / Printed in Brazil

Capa: Noni Geiger / Sérgio R. Vaz

30º Colóquio Brasileiro de Matemática

- Aplicações Matemáticas em Engenharia de Produção - Leonardo J. Lustosa e Fernanda M. P. Raupp
- Boltzmann-type Equations and their Applications - Ricardo Alonso
- Dissipative Forces in Celestial Mechanics - Sylvio Ferraz-Mello, Clodoaldo Grotta-Ragazzo e Lucas Ruiz dos Santos
- Economic Models and Mean-Field Games Theory - Diogo A. Gomes, Levon Nurbekyan and Edgard A. Pimentel
- Generic Linear Recurrent Sequences and Related Topics - Letterio Gatto
- Geração de Malhas por Refinamento de Delaunay - Marcelo Siqueira, Afonso Paiva e Paulo Pagliosa
- Global and Local Aspects of Levi-flat Hypersurfaces - Arturo Fernández Pérez e Jiří Lebl
- Introdução às Curvas Elípticas e Aplicações - Parham Salehyan
- Métodos de Descida em Otimização Multiobjetivo - L. M. Graña Drummond e B. F. Svaiter
- Modern Theory of Nonlinear Elliptic PDE - Boyan Slavchev Sirakov
- Novel Regularization Methods for Ill-posed Problems in Hilbert and Banach Spaces - Ismael R. Bleyer e Antonio Leitão
- Probabilistic and Statistical Tools for Modeling Time Series - Paul Doukhan
- **Tópicos da Teoria dos Jogos em Computação - R. C. S. Schouery, O. Lee, F. K. Miyazawa e E. C. Xavier**
- Topics in Spectral Theory - Carlos Tomei

Distribuição: IMPA
Estrada Dona Castorina, 110
22460-320 Rio de Janeiro, RJ
E-mail: ddic@impa.br
<http://www.impa.br>

ISBN: 978-85-244-0412-2

Prefácio

Por combinar aspectos da computação e da economia, a Teoria dos Jogos Algorítmica (TJA) é uma área de grande interesse tanto do ponto de vista prático quanto teórico. O objetivo deste livro é introduzir conceitos da TJA focando em alguns problemas da área. O livro apresenta alguns dos principais conceitos relacionados à Teoria dos Jogos e aborda problemas da computação do ponto de vista da Teoria dos Jogos bem como problemas da economia do ponto de vista da teoria da computação. Assim, esperamos que este livro seja uma introdução interessante ao assunto.

Este livro é voltado para alunos de computação, matemática, engenharia ou economia no final da graduação ou na pós-graduação, bem como pesquisadores já formados, que gostariam de conhecer aspectos da Teoria dos Jogos aplicados à computação. Assim, este livro pode também ser utilizado em disciplinas de graduação ou pós-graduação.

Apesar de não ser necessário um conhecimento prévio de assuntos relacionados à economia, o conhecimento de conceitos de grafos, complexidade computacional, algoritmos de aproximação e programação linear será útil. Apresentamos esses conceitos no apêndice, que acreditamos ser suficiente para um primeiro contato com este livro. Porém, para uma compreensão maior dos assuntos abordados, recomendamos que os assuntos mencionados acima sejam estudados em livros especializados. No apêndice recomendamos alguns livros para o leitor interessado.

Os problemas abordados neste livro são bastante diversos, porém todos eles estão ligados a algum aspecto interessante da TJA.

No Capítulo 1, apresentamos uma introdução aos principais conceitos relacionados à Teoria dos Jogos. Assim, tal capítulo é base para todo o conteúdo posteriormente abordado no livro.

No Capítulo 2, estudamos jogos de roteamento em que jogadores tentam escolher rotas em uma rede com o objetivo de enviar uma demanda do seu ponto de origem ao seu ponto de destino. Estudamos a ineficiência destes jogos considerando o comportamento egoísta dos jogadores.

No Capítulo 3, abordamos um jogo de balanceamento de carga, onde jogadores disputam pelo uso do recurso de um conjunto de máquinas. Mostramos que a falta de coordenação dos jogadores pode levar a um grande custo para a sociedade. Posteriormente, abordamos o uso de mecanismos de coordenação para a obtenção de configurações melhores.

No Capítulo 4, abordamos leilões, onde o objetivo é vender um item (ou mais) para um conjunto de compradores. Em particular, estamos interessados em como vender itens a compradores de forma que o comportamento estratégico dos jogadores não afetem o resultado do leilão com o objetivo de maximizar o bem-estar social.

No Capítulo 5, consideramos novamente a venda de itens a compradores. Porém, diferentemente do Capítulo 4, neste capítulo estamos interessados em maximizar o lucro do vendedor. Abordamos vários modelos diferentes para tal problema, apresentando algoritmos de aproximação para os mesmos.

No Capítulo 6, abordamos a teoria de jogos cooperativos, bem como conceitos de soluções para tais jogos. Neste caso os jogadores agem em conjunto para construção de um bem ou aquisição de um serviço. Os jogadores também tentam minimizar o seu custo, que é em função do bem construído. Uma questão central sobre estes jogos que é o compartilhamento de custos entre os jogadores. Apresentamos estes tópicos usando como exemplo jogos de localização de instalações.

Finalmente, no Apêndice A, apresentamos alguns conceitos de computação que utilizamos durante o livro. Abordamos complexidade computacional, algoritmos de aproximação, e programação linear e inteira.

O conteúdo desse livro é baseado no livro “*Algorithmic Game Theory*”, editado por Nisan, Roughgarden, Tardos e Vazirani [NRTV07], no livro “*Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*” de Shoham e Leyton-Brown [SLB08] no livro “*A Course in Game Theory*” de Osborne e Rubinstein [OR94] no capítulo “Introdução à Teoria dos Jogos Algorítmica” da “XXIX Jornada de Atualização em Informática

da SBC” [Miy10], na dissertação de mestrado de Rafael Lima Curi [Cur13], na tese de doutorado de Rafael C. S. Schouery [Sch14], além de diversos artigos citados ao longo do texto.

Agradecemos à organização do 30º Colóquio Brasileiro de Matemática pelo apoio e pela oportunidade de divulgar esta área e às agências de fomento à pesquisa CNPq (Proc. 303947/2008-0, 477692/2012-5, 311499/2014-7 e 306358/2014-0) e FAPESP (Proc. 2013/21744-8).

Esperamos que este livro sirva como uma porta de entrada e motivação para o estudo da Teoria dos Jogos Algorítmica.

Campinas, Maio de 2015

R.C.S.S., O.L., F.K.M. e E.C.X.

Conteúdo

1	Introdução	1
1.1	Conceitos de teoria dos jogos	3
2	Jogos de Roteamento	13
2.1	Notação e definições	14
2.2	Jogos potenciais	17
2.3	Roteamento atômico	19
2.3.1	Existência de equilíbrio em jogos atômicos	20
2.3.2	Preço da anarquia de jogos atômicos	25
2.4	Roteamento não-atômico	29
2.4.1	Equilíbrio de Wardrop	31
2.4.2	Existência de equilíbrio	32
2.4.3	Preço da anarquia de jogos não-atômicos	35
3	Balanceamento de Carga	39
3.1	Preliminares	40
3.2	Jogo com máquinas uniformes	43
3.2.1	Tempo de convergência	44
3.2.2	Preço da anarquia	45
3.3	Jogos com máquinas relacionadas	47
3.3.1	Tempo de convergência	47
3.3.2	Preço da anarquia	48
3.4	Estratégias mistas	48

3.5	Mecanismos de coordenação	50
4	Leilões e Mecanismos	59
4.1	Leilões de um único item	59
4.2	Mecanismos	62
4.2.1	Leilões de segundo preço e mecanismo VCG	65
4.2.2	Jogo de caminho mínimo	66
4.3	Leilões combinatórios	69
5	Problemas de Precificação	77
5.1	O problema da compra máxima	78
5.1.1	Oferta ilimitada	79
5.1.2	Oferta limitada	82
5.2	O problema da compra por preferência	86
5.2.1	Oferta limitada	88
5.3	O problema da compra mínima	89
5.4	O problema da precificação livre de inveja	91
6	Compartilhamento de Custos	97
6.1	Soluções para jogos cooperativos	99
6.2	A solução de Shapley	100
6.3	Compartilhamento de custos	101
6.4	Jogo de localização de instalações	105
6.5	Núcleo aproximado	108
6.5.1	Núcleo aproximado e localização de instalações	111
A	Conceitos de Computação	115
A.1	Complexidade computacional	115
A.2	Algoritmos de aproximação	117
A.3	Programação linear e inteira	119
	Bibliografia	123

Introdução

Com a adoção da Internet em larga escala, houve uma grande transformação na forma como as informações são divulgadas e trabalhadas. O crescimento rápido e descentralizado da Internet, permitiu disponibilizar e compartilhar informações de maneira nunca antes vista. O computador deixou de ser uma máquina individual para se tornar parte de uma rede global de comunicação, disseminação de conteúdos e comércio. Com isso, surgiram também vários problemas computacionais onde não há um controle centralizado dos agentes envolvidos. Além disso, muitas das ações destes agentes são guiados por interesses individuais, onde a decisão de um agente pode mudar a decisão de outros. Assim, surgiu a necessidade de desenvolver novas técnicas de computação teórica para lidar com a Internet.

A Teoria dos Jogos, com seus estudos das interações de cooperação e competição entre indivíduos, teve um papel crucial nesse desenvolvimento. A área de pesquisa que combina a Teoria da Computação com a Teoria dos Jogos, denominada *Teoria dos Jogos Algorítmica*, teve um grande crescimento durante os últimos anos [Rou08].

A Teoria dos Jogos é uma área da economia usada para modelar fenômenos onde dois ou mais agentes interagem entre si. Em suma, um jogo é uma ferramenta matemática para modelar situações em que indivíduos tomam decisões buscando maximizar o seu ganho (ou minimizar o seu custo), sendo que o resultado obtido depende não apenas das suas decisões mas também das decisões dos outros indivíduos. Trata-se de uma área de grande importância para a teoria econômica, como pode ser visto pelos Prêmios Nobel de Ciências Econômicas recebidos por pesquisado-

res da área pelas suas contribuições acadêmicas: Harsanyi, Nash e Selten (1994), Aumann e Schelling (2005), Hurwicz, Maskin e Myerson (2007) e Roth e Shapley (2012) [Nob15].

A Teoria dos Jogos se popularizou após a publicação do livro *Theory of Games and Economic Behavior* [vNM44] por von Neumann e Morgenstern em 1944, e teve como base trabalhos anteriores de von Neumann. De fato, von Neumann criou teorias que tiveram grande impacto em várias áreas. Uma delas é na Computação, para a qual algumas das contribuições incluem a arquitetura de computadores, princípios de programação, análise de algoritmos, entre outras [Kow96]. Em 1945, von Neumann escreveu o primeiro rascunho do relatório sobre o EDVAC [vN45], inaugurando a era dos computadores digitais e seus algoritmos. Podemos, então, dizer que a Teoria dos Jogos Algorítmica é a conjunção das teorias iniciadas por von Neumann na década de 1940.

A Teoria dos Jogos Algorítmica difere da Teoria dos Jogos e de outros conceitos de microeconomia clássica em vários aspectos. Nesta área, estamos interessados em investigar, analisar e projetar algoritmos e regras associadas a ambientes onde vários agentes têm interesses próprios, onde as decisões de um agente impactam os outros. Além de ser motivada por leilões não tradicionais (como o de bens digitais e buscas patrocinadas) e pela própria Internet, a Teoria dos Jogos Algorítmica modela aplicações através de problemas de otimização concretos, buscando soluções ótimas, resultados de impossibilidade, limitantes inferiores e superiores em aproximações e assim por diante. Ela também considera questões de complexidade computacional como restrições na viabilidade do comportamento dos participantes [Rou08].

Entre os seus objetivos, a pesquisa em Teoria dos Jogos Algorítmica preocupa-se em analisar resultados da Teoria dos Jogos do ponto de vista da computação, já que a aplicação desta teoria pode esbarrar na grandeza das instâncias envolvidas, principalmente nas aplicações da Internet. Soluções teóricas tradicionais em Teoria dos Jogos, quando vistas do ponto de vista computacional, podem ser inviáveis se pensarmos na representação do jogo, na forma como os usuários fazem suas escolhas ou na intratabilidade computacional dos vários problemas necessários para sua resolução.

Outro objetivo da pesquisa em Teoria dos Jogos Algorítmica é abordar problemas clássicos da computação, principalmente problemas de otimização, do ponto de vista da Teoria dos Jogos, isto é, considerando que os participantes desejam maximizar o seu ganho. Assim, problemas

de roteamento, escalonamento, localização de instalações e muitos outros problemas já bastante analisados na literatura foram revistos considerando tal ponto de vista. Por exemplo, Nisan e Ronen consideraram problemas algorítmicos onde não podemos assumir que os participantes seguirão a solução proposta pelo algoritmo mas sim seus próprios interesses e, portanto, o algoritmo precisa garantir que os interesses dos agentes são atingidos quando eles se comportam corretamente [NR99].

Por fim, a área da Teoria dos Jogos Algorítmica aborda questões relevantes que combinam conceitos de economia com computação para problemas atuais. Exemplos incluem questões ligadas a redes sociais e a leilões que ocorrem na internet.

1.1 Conceitos de teoria dos jogos

Nessa seção, abordamos a Teoria dos Jogos Não-Cooperativos que estuda o comportamento de agentes em situações onde a escolha de um agente é feita de forma totalmente individual ou seja sem cooperação com os demais agentes.

Um exemplo clássico da Teoria dos Jogos é o *Dilema do Prisioneiro*, que pode ser apresentado pela seguinte alegoria. Dois suspeitos foram apreendidos pela polícia cometendo crimes menores, porém a polícia não tem provas suficientes para condenar os dois suspeitos pelo crime principal. Os suspeitos são interrogados individualmente com o objetivo de convencê-los a confessarem o crime principal. Assim, a polícia faz a seguinte proposta para cada um dos suspeitos:

- Se ambos confessarem, ficarão presos por quatro anos.
- Se um suspeito confessar mas o outro não confessar, então quem confessou ficará preso por um ano e o outro ficará preso por cinco anos.
- Se nenhum confessar, então ambos ficarão presos por dois anos.

Sem se comunicarem, eles devem fazer a escolha entre confessar ou não, sem saber a escolha do outro. Como mostramos a seguir, se cada um dos suspeitos deseja apenas minimizar a quantidade de anos que ficará preso, sem se importar com o outro suspeito, e fizerem a escolha que é melhor para si, então ambos confessarão.

De forma geral, um jogo é uma ferramenta matemática para modelar situações em que vários indivíduos (chamados de jogadores ou agentes) tomam uma decisão (chamada de estratégia) buscando obter o melhor resultado para si (maximizando seu ganho ou minimizando o seu custo), que depende das decisões tomadas pelos outros jogadores. No caso do Dilema do Prisioneiro, os suspeitos são os jogadores e as estratégias possíveis são *confessar* ou *não confessar* e a pena é o custo a ser pago. A seguir formalizamos o conceito de um jogo.

Definição. Seja n um inteiro positivo. Um *jogo* (finito) é uma tripla $(P, \mathcal{S}, \mathcal{U})$, onde $P = \{1, \dots, n\}$ é o *conjunto de jogadores*, $\mathcal{S} = \{S_1, \dots, S_n\}$ é a família de conjuntos de estratégias e $\mathcal{U} = \{u_1, \dots, u_n\}$ é a família de funções de utilidade de S em \mathbb{R} com $S = S_1 \times \dots \times S_n$. Para cada jogador $i \in P$, S_i é o *conjunto de estratégias* de i e $u_i: S \rightarrow \mathbb{R}$ é a *função de utilidade* de i .

A interpretação para tal modelo é que, para um jogador $i \in P$, o conjunto S_i indica as possíveis estratégias que o jogador i pode escolher em tal jogo e se cada jogador $j \in P$ escolhe uma estratégia $s_j \in S_j$, então $s = (s_1, \dots, s_n) \in S$ é o *perfil de estratégias*. A função u_i indica se i prefere o perfil de estratégia s a um outro possível perfil de estratégias $s' \in S$, isto é, i prefere s a s' sempre que $u_i(s) > u_i(s')$.

Dizemos que um jogador é *racional* se ele busca maximizar a sua utilidade. Essa definição é bastante vaga, mas presume que o jogador irá usar todas as informações de seu conhecimento para buscar o melhor resultado no jogo. Em geral, assumimos que todos os jogadores do jogo são racionais.

Por vezes queremos considerar jogos onde cada jogador tem uma função de custo ao invés de utilidade (como no caso do Dilema do Prisioneiro) e, portanto desejam minimizar o seu custo. Nesse caso, basta considerar que o custo é representado por uma utilidade negativa.

Exemplo 1.1.1. Por exemplo, no caso do Dilema do Prisioneiro, temos dois jogadores e portanto $P = \{1, 2\}$. Representamos por C a estratégia *confessar* e por N a estratégia de *não confessar*. Temos que $S_1 = S_2 = \{C, N\}$

e $\mathcal{S} = \{S_1, S_2\}$. Por fim, $\mathcal{U} = \{u_1, u_2\}$, onde para $i, j \in P, i \neq j$, temos

$$u_i(s_1, s_2) = \begin{cases} -4 & \text{se } s_i = C \text{ e } s_j = C \\ -1 & \text{se } s_i = C \text{ e } s_j = N \\ -5 & \text{se } s_i = N \text{ e } s_j = C \\ -2 & \text{se } s_i = N \text{ e } s_j = N. \end{cases}$$

Uma outra forma mais compacta e bastante utilizada para representar um jogo de dois jogadores como o Dilema do Prisioneiro é através de uma matriz como a apresentada na Tabela 1.1.

	2		
1		C	N
	C	-4	-5
	N	-1	-2
		-4	-5
		-1	-2
		-5	-2

Tabela 1.1: Uma representação matricial do jogo Dilema do Prisioneiro. Temos dois jogadores sendo que o jogador 1 escolhe a linha e o jogador 2 escolhe a coluna. Em cada célula, representamos a utilidade do jogador 1 no canto inferior esquerdo e do jogador 2 no canto superior direito.

Note que a utilidade é sempre negativa em tal jogo e como um jogador busca maximizar sua utilidade, no Dilema do Prisioneiro um suspeito procura minimizar o número de anos preso.

Antes de apresentar outras definições, vale notar que existem jogos onde o conjunto de jogadores é infinito. De fato, no Capítulo 2 apresentamos um exemplo de tais jogos. Porém, por enquanto, focamos em jogos com um conjunto finito de jogadores.

Definição. Seja $(P, \mathcal{S}, \mathcal{U})$ um jogo. Considere um jogador $i \in P$ e um vetor $s = (s_1, \dots, s_i, \dots, s_n)$ em \mathcal{S} . Denotamos por \mathcal{S}_{-i} o conjunto $S_1 \times \dots \times S_{i-1} \times S_{i+1} \times \dots \times S_n$ e os seus elementos por s_{-i} . Para uma estratégia $s'_i \in S_i$, denotamos por (s'_i, s_{-i}) o vetor $s = (s_1, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_n)$.

Note que $s = (s_i, s_{-i})$. Tal notação é útil para comparar duas escolhas de estratégia do jogador i com as escolhas dos outros jogadores fixas.

Definição. Para um jogador $i \in P$ e $s_{-i} \in S_{-i}$, dizemos que uma estratégia $s_i \in S_i$ é uma *resposta ótima* para s_{-i} se $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$ para todo $s'_i \in S_i$.

Se o jogador i souber com certeza que os outros jogadores irão escolher s_{-i} , então o melhor resultado que ele pode obter é jogando uma resposta ótima. No Dilema do Prisioneiro, C é resposta ótima para C e também para N e por isso C é chamada de estratégia dominante.

Definição. Para um jogador $i \in P$, dizemos que uma estratégia $s_i \in S_i$ é *dominante* se, para todo $s_{-i} \in S_{-i}$, temos que s_i é uma resposta ótima para s_{-i} .

Se um jogador tem uma estratégia dominante, então ele pode jogar tal estratégia sem se preocupar com a estratégia escolhida pelos outros jogadores, já que para qualquer escolha que eles façam, tal estratégia será uma resposta ótima.

No caso do Dilema do Prisioneiro, C é uma estratégia dominante para ambos jogadores, já que trocar da estratégia N para a estratégia C sempre diminui o número de anos que o suspeito ficará preso, independente da estratégia escolhida pelo outro suspeito. Desta forma, podemos prever que, em tal situação, ambos os jogadores optariam por confessar, ou seja, escolheriam a estratégia C .

A seguir, introduzimos outro jogo que irá exemplificar as definições apresentadas. Começamos com uma definição informal do jogo.

No *Jogo de Congestionamento* dois jogadores desejam transmitir dados à partir de um ponto de origem O comum a ambos para um destino D , o que pode ser feito através de dois canais diferentes A e B . O canal A é um pouco mais rápido do que o canal B e, portanto, tem um custo menor se utilizado, mas a velocidade de ambos os canais cai drasticamente se ambos os jogadores escolherem o mesmo canal. A Figura 1.1 ilustra essa situação.

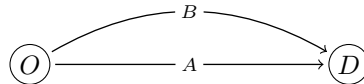


Figura 1.1: Representação gráfica do Jogo de Congestionamento onde os jogadores 1 e 2 desejam enviar pacotes de O para D e têm como opção dois canais: A (o mais rápido) e B (um pouco mais lento do que A).

Assim como no caso do Dilema do Prisioneiro, podemos considerar também a representação matricial do Jogo de Congestionamento que resume os detalhes de tal jogo. Essa representação é apresentada na Tabela 1.2.

	2		
1		A	B
A		-5	-2
		-5	-1
B		-1	-6
		-2	-6

Tabela 1.2: Representação matricial do Jogo de Congestionamento. O jogador 1 escolhe a linha e o jogador 2 escolhe a coluna. Note como os jogadores preferem o perfil de estratégias (A, B) ou (B, A) em vez de (A, A) ou (B, B) .

Note que no Jogo de Congestionamento, nenhum jogador tem uma estratégia dominante. Assim, é difícil prever quais serão as escolhas dos jogadores. A seguir introduzimos um novo conceito para tentar prever quais serão tais escolhas.

Definição. Um vetor de estratégias $s \in S$ é um *equilíbrio de Nash* se, para todo jogador $i \in P$, s_i é uma resposta ótima para s_{-i} , isto é, para todo $s'_i \in S_i$ temos que $u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$.

Um equilíbrio de Nash é uma situação onde nenhum jogador pode aumentar sua utilidade trocando de estratégia individualmente. O conceito de equilíbrio de Nash tem uma grande importância na área de Teoria dos Jogos pois é uma das formas de tentar prever qual será o comportamento dos jogadores em uma situação econômica real. Apesar de não ser claro quando os jogadores atingem ou não tal equilíbrio, tal conceito foi fundamental na história do desenvolvimento da Teoria dos Jogos e é muito importante em várias situações reais.

Note que (C, C) é um equilíbrio de Nash no Dilema do Prisioneiro. Isso ocorre porque nenhum dos suspeitos ganha ao mudar individualmente da estratégia C para a estratégia N nesta situação. De fato, se todos os jogadores têm uma estratégia dominante então o perfil de estratégias onde cada

um escolhe uma de suas estratégias dominantes é sempre um equilíbrio de Nash.

Já o Jogo de Congestionamento tem dois equilíbrios de Nash diferentes (a saber (A, B) e (B, A)), portanto é difícil prever qual será o perfil de estratégias a não ser que haja uma forma de coordenação das estratégias escolhidas pelos jogadores.

No caso do Dilema do Prisioneiro, as utilidades dos jogadores nesse jogo foram escolhidas de forma que o (único) equilíbrio de Nash ocorra quando ambos os prisioneiros confessem. Isso é algo recorrente na Teoria dos Jogos, já que muitas vezes queremos definir regras para um jogo de forma que os jogadores se comportem de uma forma específica.

Infelizmente, nem todo jogo tem um equilíbrio de Nash. Considere, por exemplo, o jogo de *Par-ou-Ímpar* onde temos dois jogadores e cada um escolhe um número par ou ímpar. O jogador 1 ganha se ambos os jogadores escolherem a mesma estratégia e o jogador 2 ganha se os jogadores escolherem estratégias diferentes. A Tabela 1.3 representa formalmente o jogo.

1	2	Par	Ímpar
Par	-1	1	-1
Ímpar	1	-1	1

Tabela 1.3: Representação matricial do jogo Par-ou-Ímpar. O jogador 1 escolhe a linha e o jogador 2 escolhe a coluna.

Podemos estender um jogo para permitir que os jogadores, em vez de escolherem uma estratégia deterministicamente, possam escolher uma estratégia de maneira não determinística, ou seja, usando aleatoriedade. Nesta extensão, as estratégias possíveis para um jogador são todas as distribuições de probabilidade sobre o conjunto de estratégias de tal jogador no jogo original e a utilidade de um jogador é a utilidade esperada no jogo original. Abaixo formalizamos tal conceito.

Definição. Dado um jogo $(P, \mathcal{S}, \mathcal{U})$, a *extensão mista* de $(P, \mathcal{S}, \mathcal{U})$ é um jogo $(P, \mathcal{S}', \mathcal{U}')$ onde $\mathcal{S}' = \{S'_1, \dots, S'_n\}$ e $\mathcal{U}' = \{u'_1, \dots, u'_n\}$ tal que,

para cada jogador $i \in P$, S'_i é um espaço de distribuições de probabilidade sobre S_i e, para cada vetor $\sigma = (\sigma_1, \dots, \sigma_n) \in S'$, onde $S' = S'_1 \times \dots \times S'_n$ temos que $u'_i(\sigma) = \mathbb{E}_{s \in \sigma}[u_i(s)]$.

Tal extensão captura a ideia de que um jogador pode querer tomar uma decisão usando aleatoriedade em um jogo arbitrário. Por exemplo, no caso do jogo Par-ou-Ímpar, os jogadores podem escolher entre par ou ímpar aleatoriamente.

Definição. Sejam (P, S, \mathcal{U}) um jogo, com $S = \{S_1, \dots, S_n\}$, e $(\sigma_1, \dots, \sigma_n)$ um vetor onde σ_i é uma distribuição de probabilidades sobre S_i . Dizemos que $(\sigma_1, \dots, \sigma_n)$ é um *equilíbrio misto de Nash* para (P, S, \mathcal{U}) se $(\sigma_1, \dots, \sigma_n)$ é um equilíbrio de Nash para a extensão mista de (P, S, \mathcal{U}) .

Um jogo nem sempre tem um equilíbrio de Nash, como é o caso do jogo Par-ou-Ímpar, mas a extensão mista de um jogo com um número finito de jogadores onde cada jogador tem um conjunto de estratégias finito tem sempre pelo menos um equilíbrio de Nash.

Teorema 1.1.2 (Nash [Nas50]). Todo jogo finito com os conjuntos de estratégias finitos tem um equilíbrio misto de Nash.

No caso do Par-ou-Ímpar, um equilíbrio misto ocorre quando ambos os jogadores escolherem par com probabilidade $1/2$ (e, conseqüentemente, ímpar com probabilidade $1/2$). De fato, esse é o único equilíbrio misto do jogo.

Já no caso do Jogo de Congestionamento, o cenário onde ambos os jogadores escolhem o caminho A com probabilidade $5/8$ e o caminho B com probabilidade $3/8$ é um equilíbrio misto. De fato, note que, se o jogador 2 joga tal estratégia e o jogador 1 escolhe A com probabilidade p e B com probabilidade $1 - p$ temos que o custo esperado do jogador 1 é

$$p \left(5 \cdot \frac{5}{8} + 1 \cdot \frac{3}{8} \right) + (1 - p) \left(2 \cdot \frac{5}{8} + 6 \cdot \frac{3}{8} \right) = p \frac{7}{2} + (1 - p) \frac{7}{2} = \frac{7}{2}.$$

Assim, com tal escolha do jogador 2, qualquer estratégia mista do jogador 1 é uma resposta ótima, em particular, escolher $p = 5/8$ é uma resposta ótima. Por simetria, o mesmo é válido para o jogador 2 e temos que o perfil de estratégias onde ambos escolhem A com probabilidade $5/8$ é um equilíbrio misto de Nash. De fato, o Jogo do Congestionamento tem apenas três equilíbrios mistos: seus dois equilíbrios puros e o equilíbrio misto

apresentado acima. Ou seja, um jogo pode ter outros equilíbrios mistos além dos seus equilíbrios puros.

Note que, no caso do Dilema do Prisioneiro, se ambos os jogadores cooperarem e escolherem N então eles ficam presos por 4 anos no total. Porém, o único equilíbrio desse jogo ocorre quando ambos confessam e, assim, ambos ficam presos por um total de 8 anos.

Considere então o *Jogo da Poluição*, onde temos n jogadores cada um com duas estratégias P (poluir) e N (não poluir). O custo de um jogador i que escolhe P é o número de jogadores que escolheram P e o custo de um jogador i que escolhe N é o número de jogadores que escolheram P mais um custo adicional de 3 para a implementação de medidas contra a poluição. Note que poluir é uma estratégia dominante já que se tiverem k outros jogadores poluindo então é melhor poluir e pagar $k + 1$ do que não poluir e pagar $k + 3$. Assim, tal jogo tem apenas um equilíbrio onde todos os jogadores poluem, resultando em um custo total de n^2 . Se todos os jogadores cooperassem e ninguém poluísse, então teríamos um custo total de $3n$.

Com essa discrepância em mente, estamos interessados em analisar em um determinado jogo qual é a desvantagem que obtemos por considerar jogadores racionais se comparado com o melhor resultado que poderia ocorrer para a sociedade. Baseado neste tipo de comparação, definimos a seguir o *preço da anarquia* e o *preço da estabilidade*.

Seja J um jogo com conjunto de vetores de estratégias S e $f: S \rightarrow \mathbb{R}$ uma função de bem-estar social. Denotamos por $\mathcal{E}(J)$ o conjunto de perfis de estratégias de J em equilíbrio e $\text{OPT}(J) = \max\{f(s) : s \in S\}$. Com isso, s é um perfil de estratégias *ótimo social* se $f(s) = \text{OPT}(J)$. Usualmente, consideramos f como a soma das utilidades dos jogadores para um determinado perfil de estratégias.

O objetivo, para o sistema como um todo, é que o perfil de estratégias em equilíbrio atingido pelos jogadores esteja próximo do bem-estar social máximo (de preferência, que tenha o mesmo valor).

O *preço da anarquia*, PA, de um jogo J é a maior razão entre o valor de um perfil de estratégias em equilíbrio e o valor de um perfil de estratégias ótimo social [KP99]. Mais formalmente, temos que

$$\text{PA} = \max \left\{ \frac{\text{OPT}(J)}{f(s)} : s \in \mathcal{E}(J) \right\}.$$

O *preço da estabilidade*, PE, de um jogo J é a menor razão entre o valor de um perfil de estratégias em equilíbrio e o valor de um perfil de estratégias ótimo social [ADK⁺08]. Formalmente, temos

$$PE = \min \left\{ \frac{\text{OPT}(J)}{f(s)} : s \in \mathcal{E}(J) \right\}.$$

Note que, sempre temos que $PA \geq PE \geq 1$.

O preço da anarquia, nos diz o quão longe um equilíbrio pode estar do bem-estar social máximo e o preço da estabilidade diz o quão perto ele pode estar deste valor.

Em jogos com função de custo, ao invés de utilidade, podemos definir PA e PE similarmente invertendo as frações. No exemplo do Jogo da Poluição, por exemplo temos $PA = PE = n^2/(3n) = n/3$ o que indica que tanto o melhor quando o pior equilíbrio é $n/3$ vezes mais custoso para a sociedade.

Para ilustrar, considere o Jogo do Congestionamento. O bem-estar social máximo ocorre quando um jogador escolhe o caminho A e o outro jogador escolhe o caminho B e tem custo 3. Note que nesta situação também temos um equilíbrio puro, e portanto o preço da estabilidade para tal jogo é 1. Porém, como vimos anteriormente, além desses dois equilíbrios puros, existe também um equilíbrio misto onde cada jogador tem custo $7/2$ e portanto o preço da anarquia para tal jogo é $7/6$.

Jogos de Roteamento

O interesse por jogos de roteamento nos dias atuais é facilmente justificado pela grande quantidade de situações que tal modelo captura. Uma situação típica do cotidiano que pode ser modelada por este tipo de jogo é a seguinte: suponha que os jogadores sejam motoristas em uma grande metrópole e que cada um deseja se locomover do seu local de origem até seu destino, minimizando o tempo de viagem. Outro exemplo importante é aquele em que cada jogador é um servidor que deseja enviar (o mais rapidamente possível) uma certa quantidade de informação através de uma grande rede de comunicação (como a Internet) de um certo endereço origem para um endereço destino. Uma característica fundamental que é comum a ambos exemplos é fato da escolha de cada jogador depender da escolha dos demais jogadores, uma vez que um dos fatores mais relevantes no tempo de percurso é a quantidade de tráfego (número de jogadores) em cada trecho de uma rota (ruas no primeiro exemplo e cabos de fibra ótica no segundo).

Jogos de roteamento são tradicionalmente divididos em duas categorias: jogos atômicos e jogos não-atômicos. Cada um desses tem sua própria noção de equilíbrio: nos jogos atômicos considere-se o tradicional equilíbrio de Nash, enquanto o segundo trata do equilíbrio de Wardrop. Neste capítulo consideramos as duas versões e apresentamos resultados importantes sobre ambas. Em particular, para cada uma delas, exploramos as seguintes questões:

- (a) Sempre existe um equilíbrio? Caso contrário, em que condições existe um equilíbrio?

- (b) Quando um equilíbrio existe, o quão ineficiente ele é? Ou seja, qual é o preço da anarquia?

Mostramos também que jogos de roteamento podem ser estudados dentro do modelo mais geral de jogos potenciais. Em particular, o método da função potencial é amplamente usado neste capítulo para provar a existência de equilíbrios.

2.1 Notação e definições

Um *jogo de roteamento* é realizado em um grafo orientado $G = (V, E)$. Representamos os jogadores pelo conjunto $\mathcal{I} = \{1, 2, \dots, k\}$. Cada jogador está associado a um par (s_i, t_i) de vértices de G , chamado de par origem-destino (OD). Note que jogadores distintos podem estar associados a um mesmo par OD. Cada jogador tem uma *demanda* $r_i > 0$ que representa a quantidade de tráfego que ele quer mandar da sua origem ao seu destino. Representamos todas as demandas de forma compacta por um *vetor de demandas* r indexado por \mathcal{I} . Um caso particular importante, que chamamos de *jogo de roteamento com demanda uniforme*, é aquele em que todas as demandas são iguais a uma constante.

Cada aresta $e \in E$ está associada a uma função custo $c_e: \mathbb{R} \rightarrow \mathbb{R}_+$ que supomos ser não-negativa, contínua e não-decrescente. Intuitivamente, essas características capturam o fato de que quanto maior o tráfego x em uma aresta e , maior é o custo $c_e(x)$ (isto é intuitivo se pensarmos que e representa uma estrada e c_e é o *tempo de percurso*). Usamos o vetor c indexado por E para denotar todas as funções custo da rede.

Para definir o conjunto de estratégias de cada jogador é necessário introduzir o conceito de *fluxo*. Considere um jogador i . Seja \mathcal{P}^i o conjunto de todos os $s_i t_i$ -caminhos em G . Usamos também a notação $\mathcal{P} = \bigcup_{i \in \mathcal{I}} \mathcal{P}^i$. Um $s_i t_i$ -*fluxo* (ou simplesmente *fluxo*) é uma função $f^i: \mathcal{P}^i \rightarrow \mathbb{R}_+$. Informalmente, para um caminho $P \in \mathcal{P}$, o valor $f_P^i = f^i(P)$ é a quantidade de fluxo que o jogador i manda através de P . Dizemos que o $s_i t_i$ -fluxo f^i é *viável* para i se $\sum_{P \in \mathcal{P}^i} f_P^i = r_i$, ou seja, se envia toda a demanda de s_i a t_i . Uma estratégia para um jogador i é um $s_i t_i$ -fluxo viável. Assim, um perfil de estratégias é um *fluxo* $f = (f^1, \dots, f^k)$. Dizemos que f é *viável*

se cada f^i for viável para $i = 1, \dots, k$. Usamos a notação

$$f_e^i = \sum_{P \in \mathcal{P}^i : e \in P} f_P^i \quad (2.1)$$

para $i = 1, \dots, k$ para representar a quantidade de fluxo do jogador i que passa por uma aresta e . Além disso, usamos

$$f_e = \sum_{i \in \mathcal{I}} f_e^i = \sum_{i=1}^k f_e^i \quad (2.2)$$

para indicar a soma dos fluxos que passam por e , considerando todos os jogadores.

Definimos o *custo* de um caminho P em relação a um fluxo f como sendo a soma dos custos de suas arestas, ou seja,

$$c_P(f) = \sum_{e \in P} c_e(f_e). \quad (2.3)$$

O *custo individual* de um jogador i em relação a um fluxo f é definido como:

$$C^i(f) = \sum_{P \in \mathcal{P}^i} c_P(f) f_P^i. \quad (2.4)$$

Usando (2.3) podemos reescrever a definição como:

$$C^i(f) = \sum_{e \in E} c_e(f_e) f_e^i. \quad (2.5)$$

Assim, um fluxo é um equilíbrio de Nash se nenhum jogador consegue modificar sua estratégia (fluxo) e reduzir seu custo individual.

Isto completa a descrição do conceito de um *jogo de roteamento atômico*. Usamos as mesmas notações para *jogos de roteamento não-atômicos*, mas postergamos a explicação sobre a diferença entre esses modelos até a Seção 2.4.

Uma instância de um jogo de roteamento (atômico ou não) é denotada por (G, r, c) deixando implícito os demais elementos.

Exemplo 2.1.1. Ilustramos as definições com o exemplo da Figura 2.1 (retirado de [Cur13]). O conjunto dos jogadores é $\mathcal{I} = \{1, 2\}$ com vetor de

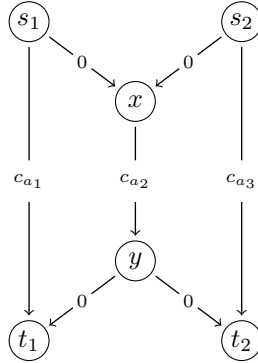


Figura 2.1: Exemplo de instância de um jogo de roteamento

demandas $r = (r_1, r_2)$. Os pares OD dos jogadores são (s_1, t_1) e (s_2, t_2) , respectivamente.

Um exemplo de fluxo viável é $f = (f^1, f^2)$ em que o jogador 1 envia toda sua demanda pela aresta $a_1 = (s_1, t_1)$ e o jogador 2 envia toda sua demanda pela aresta $a_3 = (s_2, t_2)$. Assim, ignorando-se as arestas de custo zero, temos que $f^1 = (f_{a_1}^1, f_{a_2}^1, f_{a_3}^1) = (r_1, 0, 0)$ e $f^2 = (0, 0, r_2)$. O custo individual do jogador 1 é $C^1(f) = c_{a_1}(f_{a_1})r_1 = c_{a_1}(r_1)r_1$ e do jogador 2 é $C^2(f) = c_{a_3}(f_{a_3})r_2 = c_{a_3}(r_2)r_2$.

Dentro do modelo de jogos atômicos temos uma divisão entre jogos *divisíveis* e *indivisíveis*. No primeiro, a estratégia de cada jogador i é um fluxo f^i (como definido anteriormente) enquanto no segundo, o conjunto das estratégias de um jogador i é \mathcal{P}^i . Ou seja, no jogo indivisível, cada jogador tem que mandar sua demanda ao longo de um único caminho. Neste capítulo nos restringimos ao estudo de jogos indivisíveis por terem sido melhor estudados e compreendidos. O leitor interessado em jogos atômicos divisíveis pode consultar o trabalho de Curi [Cur13]. Formalmente, no jogo de roteamento atômico (indivisível), um $s_i t_i$ -fluxo f^i é viável se $f_P^i = r_i$ para exatamente um caminho $P \in \mathcal{P}^i$ e $f_Q = 0$ para todo $Q \in \mathcal{P}^i \setminus \{P\}$. Um fluxo $f = (f^1, \dots, f^k)$ é viável se f^i for viável para todo $i \in \mathcal{I}$.

Em outras seções estudamos a ineficiência do equilíbrio de jogos de roteamento. Para tal, usamos como custo social a função utilitária que consiste no custo total do tráfego. Mais precisamente, se f é um fluxo viável

para (G, r, c) , então o custo de f é dado por:

$$C(f) = \sum_{i=1}^k C^i(f) \quad (2.6)$$

Usando (2.3) podemos reescrever (2.6) como:

$$C(f) = \sum_{e \in E} c_e(f_e) f_e. \quad (2.7)$$

Dizemos que um fluxo viável f é *ótimo* para (G, r, c) se ele for um fluxo de custo mínimo entre todos os fluxos viáveis.

Definimos o *preço da anarquia* para jogos de roteamento como sendo a razão entre o custo do pior equilíbrio e o custo de uma solução ótima. Em muitos exemplos que estudamos, restringimos a classe das funções custo (por exemplo, funções polinomiais). Neste caso, fica-se subentendido que ao falar de preço de anarquia consideramos apenas os equilíbrios e as soluções ótimas de instâncias com essa restrição de custo.

2.2 Jogos potenciais

Vários casos particulares de jogos de roteamento pertencem a uma classe mais geral de jogos, chamados *jogos potenciais*. Um jogo deste tipo possui uma *função potencial* definida sobre os perfis de estratégia do jogo tal que um ótimo global desta corresponde a um equilíbrio de Nash do jogo. No que segue apresentamos formalmente este conceito e os principais resultados.

Lembre-se que denotamos um jogo na forma normal como uma tripla $(\mathcal{I}, \{\mathcal{S}_i\}_{i \in \mathcal{I}}, \{C_i\}_{i \in \mathcal{I}})$ onde \mathcal{I} é o conjunto de jogadores, \mathcal{S}_i é o conjunto de estratégias do jogador i e C_i é a função custo do jogador i . Denotamos também o conjunto de todos os perfis do jogo por \mathcal{S} .

Definição. Uma função $\Phi: \mathcal{S} \rightarrow \mathbb{R}$ é uma *função potencial* para um jogo $(\mathcal{I}, (\mathcal{S}_i)_{i \in \mathcal{I}}, (C_i)_{i \in \mathcal{I}})$ se, para todo $i \in \mathcal{I}$ e para todo $s \in \mathcal{S}$, temos que:

$$C_i(s'_i, s_{-i}) - C_i(s_i, s_{-i}) \geq 0 \Leftrightarrow \Phi(s'_i, s_{-i}) - \Phi(s_i, s_{-i}) \geq 0 \quad (2.8)$$

para todo $s'_i \in \mathcal{S}_i$.

Uma função potencial é *exata* se

$$\Phi(s'_i, s_{-i}) - \Phi(s_i, s_{-i}) = C_i(s'_i, s_{-i}) - C_i(s_i, s_{-i}). \quad (2.9)$$

Em outras palavras, a variação de custo do jogador i ao trocar a estratégia s_i por s'_i é positiva se, e somente se, a variação de potencial nos pontos correspondentes é positiva. Em particular, para uma função potencial exata, a variação no custo é igual à variação na função potencial.

Um jogo finito que possui uma função potencial (exata) é chamado *jogo potencial (exato)*. O próximo resultado mostra a relevância de jogos potenciais.

Teorema 2.2.1. Seja $(\mathcal{I}, (\mathcal{S}_i)_{i \in \mathcal{I}}, (C_i)_{i \in \mathcal{I}})$ um jogo potencial finito. O perfil de estratégias que minimiza sua função potencial é um equilíbrio de Nash.

Demonstração. Seja Φ uma função potencial para o jogo e seja s um perfil de estratégias que minimiza Φ . Seja i um jogador qualquer e seja s'_i uma estratégia para i . Pela escolha de s , temos que $\Phi(s'_i, s_{-i}) \geq \Phi(s_i, s_{-i})$. Pela definição de função potencial, temos que $C_i(s'_i, s_{-i}) - C_i(s_i, s_{-i}) \geq 0$. Assim, o jogador i não tem incentivo para se desviar de sua estratégia s_i para s'_i . Isto implica que s é um equilíbrio do jogo. \square

Um exemplo importante de jogo potencial é o *jogo de congestionamento* introduzida por Rosenthal [Ros73]. Um jogo de congestionamento é composto de k jogadores e m recursos. O conjunto de estratégias \mathcal{S}_i de um jogador i é uma coleção de subconjuntos de recursos. Cada recurso j está associada a uma função custo $c_j(x)$ que depende da quantidade x de jogadores cujas estratégias contém o recurso j . O custo total que o jogador i paga quando escolhe a estratégia s_i é a soma dos custos dos recursos em s_i . Em outras palavras, se x_j denota a quantidade de jogadores que usam o recurso j , então i paga $\sum_{j \in s_i} c_j(x_j)$. Por exemplo, jogos de roteamento atômico (com demanda uniforme) são uma classe particular de jogos de congestionamento em que cada recurso é uma aresta do grafo e uma estratégia para um jogador i é um subconjunto de arestas que corresponde a um caminho ligando seu par OD (s_i, t_i) .

Rosenthal [Ros73] provou que todo jogo de congestionamento é um jogo potencial. Em 1996, Monderer e Shapley [MS96] mostraram uma espécie de recíproca: para todo jogo potencial, existe um jogo de congestionamento com a mesma função potencial.

2.3 Roteamento atômico

Apresentamos a seguir uma caracterização de equilíbrio de Nash em jogos de roteamento atômicos.

Suponha que $f = (f_1, \dots, f_k)$ seja um equilíbrio de Nash de um jogo atômico (indivisível) (G, r, c) . Lembre-se que a estratégia de um jogador i é um fluxo f^i tal que $f_P^i = r_i$ para algum caminho $P \in \mathcal{P}^i$ e $f_Q^i = 0$ para todo $Q \in \mathcal{P}^i \setminus \{P\}$. Para um jogador i , denotamos

$$f^{(-i)} = (f^1, \dots, f^{i-1}, f^{i+1}, \dots, f^k)$$

e para toda estratégia \tilde{f}^i de i denotamos

$$(\tilde{f}^i, f^{(-i)}) = (f^1, \dots, f^{i-1}, \tilde{f}^i, f^{i+1}, \dots, f^k).$$

De acordo com a definição de equilíbrio de Nash, um fluxo f é um equilíbrio se para todo jogador i , temos que

$$C^i(f) \leq C^i(\tilde{f}^i, f^{(-i)}),$$

para toda estratégia \tilde{f}^i de i . Seja \tilde{f}^i uma estratégia qualquer de i e seja \tilde{P} o caminho correspondente a esta estratégia. Usando a equação (2.4), segue que

$$C^i(f) - C^i(\tilde{f}^i, f^{(-i)}) = r_i(c_P(f) - c_{\tilde{P}}(\tilde{f})).$$

Logo, o lado esquerdo da equação acima é não-positivo se e somente se $c_P(f) \leq c_{\tilde{P}}(\tilde{f})$. Assim, podemos definir equilíbrio de Nash para jogos atômicos (indivisíveis) da seguinte forma.

Definição. Seja f um fluxo viável para (G, r, c) . Então $f = (f_1, \dots, f_k)$ é um *equilíbrio* se, e somente se, para todo jogador $i \in \{1, \dots, k\}$ e todo par $P, \tilde{P} \in \mathcal{P}^i$ com $f_P^i > 0$ temos que

$$c_P(f) \leq c_{\tilde{P}}(\tilde{f}), \quad (2.10)$$

onde \tilde{f} é o fluxo idêntico a f exceto que $\tilde{f}_P^i = 0$ e $\tilde{f}_{\tilde{P}}^i = r_i$.

Esta definição é mais conveniente para provar os resultados de jogos atômicos deste capítulo. Note que a desigualdade (2.10) pode ser escrita como

$$c_P(f) \leq \sum_{e \in P \cap \tilde{P}} c_e(f_e) + \sum_{e \in \tilde{P} \setminus P} c_e(f_e + r_i), \quad (2.11)$$

sem a necessidade de definir \tilde{f} .

2.3.1 Existência de equilíbrio em jogos atômicos

Primeiramente apresentamos uma instância de um jogo atômico devido a Goemans, Mirrokni e Vetta [GMV05] em que os custos são funções polinomiais e que não possui nenhum equilíbrio.

Exemplo 2.3.1. Considere dois jogadores 1 e 2, ambos com par OD igual a (s, t) e demandas $r_1 = 1$ e $r_2 = 2$. Considere os caminhos $P_1 = (s, t)$, $P_2 = (s, v, t)$, $P_3 = (s, w, t)$ e $P_4 = (s, v, w, t)$ como na Figura 2.2.

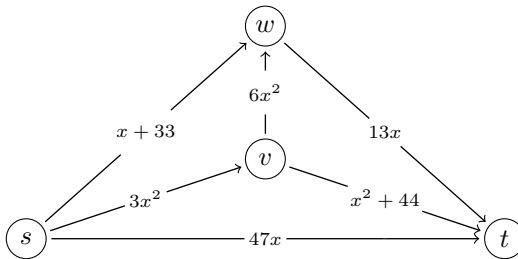


Figura 2.2: Instância atômica do jogo de roteamento que não possui equilíbrio.

As seguintes afirmações implicam que não existe equilíbrio nesta rede.

- (1) Se o jogador 2 escolhe P_1 ou P_2 , então a única escolha do jogador 1 que minimiza seu custo é P_4 .
- (2) Se o jogador 2 escolhe P_3 ou P_4 , então a única escolha do jogador 1 que minimiza seu custo é P_1 .
- (3) Se o jogador 1 escolhe P_4 , então a única escolha do jogador 2 que minimiza seu custo é P_3 .
- (4) Se o jogador 1 escolhe P_1 , então a única escolha do jogador 2 que minimiza seu custo é P_2 .

Deixamos a cargo do leitor a verificação das afirmações (1)-(4).

Em vista do Exemplo 2.3.1, vários pesquisadores estudaram vários casos particulares, restringindo a classe de funções custo ou as demandas.

Para vários desses jogos particulares, é possível mostrar a existência de um equilíbrio e determinar exatamente o preço da anarquia. A seguir apresentamos alguns desses resultados.

O primeiro resultado é sobre jogos de roteamento atômico com demanda uniforme. Sem perda de generalidade, podemos supor que a demanda comum é igual a 1.

Teorema 2.3.2. Seja (G, r, c) uma instância atômica na qual $r_i = 1$ para todo $i = 1, \dots, k$. Então (G, r, c) possui um equilíbrio.

Demonstração. Considere a função

$$\Phi(f) = \sum_{e \in E} \sum_{j=1}^{f_e} c_e(j)$$

definida sobre o conjunto dos fluxos viáveis. A seguir, mostramos que Φ é uma função potencial (exata) para este jogo atômico. Segue então do Teorema 2.2.1 que o jogo possui um equilíbrio.

Sejam f um fluxo viável e i um jogador qualquer. Suponha que $f_P^i = 1$ para algum $P \in \mathcal{P}^i$ e $f_Q^i = 0$ para qualquer outro caminho $Q \in \mathcal{P}^i$. Seja $\tilde{P} \in \mathcal{P}^i$ e seja \tilde{f} o fluxo idêntico a f exceto que $\tilde{f}_P^i = 0$ e $\tilde{f}_{\tilde{P}}^i = 1$.

Antes de prosseguir, faremos algumas observações simples. Seja $P \circ \tilde{P} = (E \setminus (P \cup \tilde{P})) \cup (P \cap \tilde{P})$. Note que para $e \in P \circ \tilde{P}$, temos que $f_e = \tilde{f}_e$, para toda aresta $e \in \tilde{P} \setminus P$, temos que $\tilde{f}_e = f_e + 1$ e para toda aresta $e \in P \setminus \tilde{P}$, temos que $\tilde{f}_e = f_e - 1$. Assim, os custo dos caminhos P e \tilde{P} podem ser escritos como:

$$c_P(f) = \sum_{e \in (P \cap \tilde{P})} c_e(f_e) + \sum_{e \in P \setminus \tilde{P}} c_e(f_e), \quad (2.12)$$

$$c_{\tilde{P}}(\tilde{f}) = \sum_{e \in (P \cap \tilde{P})} c_e(f_e) + \sum_{e \in \tilde{P} \setminus P} c_e(f_e + 1). \quad (2.13)$$

Também das observações acima, temos que:

$$\Phi(f) = \sum_{e \in P \circ \tilde{P}} \sum_{j=1}^{f_e} c_e(j) + \sum_{e \in \tilde{P} \setminus P} \sum_{j=1}^{f_e} c_e(j) + \sum_{e \in P \setminus \tilde{P}} \sum_{j=1}^{f_e} c_e(j), \quad (2.14)$$

$$\Phi(\tilde{f}) = \sum_{e \in P \circ \tilde{P}} \sum_{j=1}^{f_e} c_e(j) + \sum_{e \in \tilde{P} \setminus P} \sum_{j=1}^{f_e+1} c_e(j) + \sum_{e \in P \setminus \tilde{P}} \sum_{j=1}^{f_e-1} c_e(j). \quad (2.15)$$

Subtraindo (2.12) de (2.13) e (2.14) de (2.15) concluímos que

$$c_{\tilde{P}}(\tilde{f}) - c_P(f) = \Phi(\tilde{f}) - \Phi(f),$$

e portanto, Φ é uma função potencial exata. Isto conclui a prova. \square

Vale a pena ressaltar que a prova do Teorema 2.3.2 não pressupõe nenhuma hipótese sobre as funções custo.

Mostramos agora que jogos de roteamento atômicos com custos afins (e demanda arbitrária) possuem um equilíbrio. Antes disso, apresentamos o seguinte lema técnico.

Lema 2.3.3. Seja c uma função real afim dada por $c(x) = ax + b$ com $a, b \in \mathbb{R}$. Então para todo $r \in \mathbb{R}$ temos que

$$c(x+r)(x+r) + c(r)r - c(x)x = 2r \cdot c(x+r), \quad (2.16)$$

$$c(x-r)(x-r) - c(r)r - c(x)x = -2r \cdot c(x). \quad (2.17)$$

Demonstração. Por meio de simples cálculos, temos que

$$\begin{aligned} c(x+r)(x+r) + c(r)r - c(x)x &= \\ &= (a(x+r) + b)(x+r) + (ar + b)r - (ax + b)x \\ &= (a(x+r) + b)(x+r) + ar^2 + br - ax^2 - bx \\ &= (a(x+r) + b)(x+r) + (a(x+r) + b)(r-x) \\ &= 2r(a(x+r) + b) = 2r \cdot c(x+r), \end{aligned}$$

e portanto (2.16) vale.

A demonstração de (2.17) é análoga e a omitimos. \square

A seguir, apresentamos o resultado principal para existência de equilíbrio em jogos com custos afins.

Teorema 2.3.4. Seja (G, r, c) uma instância de um jogo atômico onde para cada $e \in E$, a função custo c_e é afim com coeficientes não-negativos. Então (G, r, c) possui um equilíbrio.

Demonstração. Suponha que $c_e(f) = a_e(f) + b_e$ com $a_e, b_e \geq 0$ para todo $e \in E$.

Para cada $e \in E$, defina a função:

$$h_e(f_e) = c_e(f_e)f_e + \sum_{j \in \mathcal{I}_e} c_e(r_j)r_j = (a_e(f_e)f_e + b_e)f_e + \sum_{j \in \mathcal{I}_e} (a_e r_j + b_e)r_j,$$

onde \mathcal{I}_e é o conjunto dos jogadores que usam um caminho passando por e . Considere a função

$$\Phi(f) = \sum_{e \in E} h_e(f_e) = \sum_{e \in E} \left(c_e(f_e)f_e + \sum_{j \in \mathcal{I}_e} c_e(r_j)r_j \right). \quad (2.18)$$

A seguir, mostramos que Φ é uma função potencial para este jogo.

Sejam f um fluxo viável e i um jogador qualquer. Suponha que $f_P^i = 1$ para algum $P \in \mathcal{P}^i$ e $f_Q^i = 0$ para qualquer outro caminho $Q \in \mathcal{P}^i$. Seja $\tilde{P} \in \mathcal{P}^i$ e seja \tilde{f} o fluxo idêntico a f exceto que $\tilde{f}_P^i = 0$ e $\tilde{f}_{\tilde{P}}^i = 1$. Como na prova do Teorema 2.3.2 seja $P \circ \tilde{P} = (E \setminus (P \cup \tilde{P})) \cup (P \cap \tilde{P})$. Note que para $e \in P \circ \tilde{P}$, temos que $f_e = \tilde{f}_e$, para toda aresta $e \in \tilde{P} \setminus P$, temos que $\tilde{f}_e = f_e + r_i$ e para toda aresta $e \in P \setminus \tilde{P}$, temos que $\tilde{f}_e = f_e - r_i$. Denote por $\tilde{\mathcal{I}}_e$ o conjunto dos jogadores que usam um caminho passando por e no fluxo \tilde{f} . Note que para toda aresta $e \in P \circ \tilde{P}$, temos que $\mathcal{I}_e = \tilde{\mathcal{I}}_e$, para toda aresta $e \in \tilde{P} \setminus P$, temos que $\tilde{\mathcal{I}}_e = \mathcal{I}_e \cup \{i\}$ e para toda aresta $e \in P \setminus \tilde{P}$, temos que $\tilde{\mathcal{I}}_e = \mathcal{I}_e - \{i\}$.

Assim,

$$c_P(f) = \sum_{e \in (P \cap \tilde{P})} c_e(f_e) + \sum_{e \in P \setminus \tilde{P}} c_e(f_e), \quad (2.19)$$

$$c_{\tilde{P}}(\tilde{f}) = \sum_{e \in (P \cap \tilde{P})} c_e(f_e) + \sum_{e \in \tilde{P} \setminus P} c_e(f_e + r_i). \quad (2.20)$$

Logo,

$$c_{\tilde{P}}(\tilde{f}) - c_P(f) = \sum_{e \in \tilde{P} \setminus P} c_e(f_e + r_i) - \sum_{e \in P \setminus \tilde{P}} c_e(f_e). \quad (2.21)$$

Também temos que:

$$\Phi(f) = \sum_{e \in P \circ \tilde{P}} h_e(f_e) + \sum_{e \in \tilde{P} \setminus P} h_e(f_e) + \sum_{e \in P \setminus \tilde{P}} h_e(f_e), \quad (2.22)$$

$$\Phi(\tilde{f}) = \sum_{e \in P \circ \tilde{P}} h_e(\tilde{f}_e) + \sum_{e \in \tilde{P} \setminus P} h_e(\tilde{f}_e) + \sum_{e \in P \setminus \tilde{P}} h_e(\tilde{f}_e). \quad (2.23)$$

Lembrando que $\tilde{f}_e = f_e$ para $e \in P \circ \tilde{P}$, temos que:

$$\begin{aligned} \Phi(\tilde{f}) - \Phi(f) &= \sum_{e \in \tilde{P} \setminus P} \left(h_e(\tilde{f}_e) - h_e(f_e) \right) \\ &+ \sum_{e \in P \setminus \tilde{P}} \left(h_e(\tilde{f}_e) - h_e(f_e) \right). \end{aligned} \quad (2.24)$$

Calculamos agora o valor de cada termo de cada uma das somatórias acima. Considere primeiro uma aresta $e \in \tilde{P} \setminus P$. Como $\tilde{f}_e = f_e + r_i$ e $\tilde{\mathcal{I}}_e = \mathcal{I}_e \cup \{i\}$, temos que

$$\begin{aligned} h_e(\tilde{f}_e) - h_e(f_e) &= c_e(f_e + r_i)(f_e + r_i) + \sum_{j \in \mathcal{I}_e \cup \{i\}} c_e(r_j)r_j \\ &- c_e(f_e)f_e - \sum_{j \in \mathcal{I}_e} c_e(r_j)r_j \\ &= c_e(f_e + r_i)(f_e + r_i) + c_e(r_i)r_i - c_e(f_e)f_e \\ &= 2r_i \cdot c_e(f_e + r_i), \end{aligned} \quad (2.25)$$

onde a última igualdade segue de (2.16).

Agora suponha que $e \in P \setminus \tilde{P}$. Como $\tilde{f}_e = f_e - r_i$ e $\tilde{\mathcal{I}}_e = \mathcal{I}_e \setminus \{i\}$, temos que

$$\begin{aligned} h_e(\tilde{f}_e) - h_e(f_e) &= c_e(f_e - r_i)(f_e - r_i) + \sum_{j \in \mathcal{I}_e \setminus \{i\}} c_e(r_j)r_j \\ &- c_e(f_e)f_e - \sum_{j \in \mathcal{I}_e} c_e(r_j)r_j \\ &= c_e(f_e - r_i)(f_e - r_i) - c_e(r_i)r_i - c_e(f_e)f_e \\ &= 2r_i \cdot c_e(f_e), \end{aligned} \quad (2.26)$$

onde a última igualdade segue de (2.17). Logo, substituindo (2.25) e (2.26) em (2.24) temos que

$$\Phi(\tilde{f}) - \Phi(f) = 2r_i \left(\sum_{e \in \tilde{P} \setminus P} c(f_e + r_i) - \sum_{e \in P \setminus \tilde{P}} c(f_e) \right), \quad (2.27)$$

que combinado com (2.21) implica que Φ é uma função potencial. Do Teorema 2.2.1 segue que o jogo possui um equilíbrio. \square

2.3.2 Preço da anarquia de jogos atômicos

Na subseção anterior mostramos que jogos atômicos com demanda uniforme sempre possuem um equilíbrio, mesmo para funções de custo arbitrárias. Entretanto, isso não garante necessariamente que o preço da anarquia seja limitado. Roughgarden e Tardos [RT02] apresentaram uma família infinita de instâncias de jogos atômicos com demanda uniforme em que o preço da anarquia tende a infinito. As funções custo dessas instâncias são da forma $1/l(x)$ onde l é uma função afim. Nesta seção, concentramos-nos apenas em instâncias atômicas em que os custos são funções afins.

Começamos com o caso de demanda uniforme. Antes precisamos do seguinte resultado cuja prova omitimos.

Lema 2.3.5. Para quaisquer reais $\alpha, \beta \geq 0$, temos que

$$\alpha(\beta + 1) \leq \frac{5}{3}\alpha^2 + \frac{1}{3}\beta^2.$$

O próximo resultado foi obtido independentemente por Awerbuch, Azar e Epstein [AAE05] e Christodoulou e Koutsoupias [CK05]

Teorema 2.3.6. Seja (G, r, c) uma instância atômica na qual $r_i = 1$ para todo $i = 1, \dots, k$ e cada aresta e tem custo afim $c_e(x) = a_e(x) + b_e$ com $a_e, b_e \geq 0$. Então o preço da anarquia de (G, r, c) é no máximo $5/2$.

Demonstração. Sejam f um equilíbrio e f^* uma solução ótima. Para cada jogador i , suponha que P_i é a estratégia de i em f e P_i^* é a estratégia de i em f^* . Pela definição de equilíbrio, um jogador i não tem incentivo para trocar sua estratégia P_i por P_i^* . Logo, para cada jogador i temos que

$$\begin{aligned} c_{P_i}(f) &\leq \sum_{e \in P_i \cap P_i^*} c_e(f_e) + \sum_{e \in P_i^* \setminus P_i} c_e(f_e + 1) \\ &\leq \sum_{e \in P_i^*} c_e(f_e + 1), \end{aligned} \tag{2.28}$$

onde a primeira desigualdade segue de (2.11) e a última desigualdade segue do fato de c_e ser não-decrescente.

Assim,

$$\begin{aligned}
C(f) &= \sum_{i=1}^k \sum_{e \in P_i} c_e(f_e) \\
&\leq \sum_{i=1}^k \sum_{e \in E} c_e(f_e + 1) \\
&= \sum_{e \in P_i^*} \sum_{i: e \in P_i^*} c_e(f_e + 1) \\
&= \sum_{e \in E} f_e^* c_e(f_e + 1) \\
&= \sum_{e \in E} [a_e f_e^* (f_e + 1) + b_e f_e^*] \\
&\leq \sum_{e \in E} [a_e \left(\frac{5}{3} (f_e^*)^2 + \frac{1}{3} f_e^2 \right) + b_e f_e^*] \tag{2.29} \\
&\leq \frac{5}{3} \left(\sum_{e \in E} f_e^* (a_e f_e^* + b_e) \right) + \frac{1}{3} \sum_{e \in E} a_e (f_e)^2 \\
&= \frac{5}{3} C(f^*) + \frac{1}{3} C(f),
\end{aligned}$$

onde (2.29) segue do Lema 2.3.5 (tomando $\alpha = f_e^*$ e $\beta = f_e$). Portanto,

$$C(f) \leq \frac{5}{2} C(f^*),$$

e isto conclui a prova. \square

O limitante do Teorema 2.3.6 é justo. Veja o exemplo a seguir, devido a Awerbuch, Azar e Epstein [AAE05], que mostra que o preço da anarquia desse jogo é pelo menos $5/2$.

Exemplo 2.3.7. Considere quatro jogadores, cada um com demanda unitária. Na Figura 2.3 indicamos os pares OD e as funções custo de cada aresta.

Cada jogador i tem duas possíveis estratégias: escolher a aresta (s_i, t_i) ou um caminho de comprimento dois. O fluxo f em que cada jogador escolhe a correspondente aresta é um equilíbrio. Este também é um fluxo ótimo

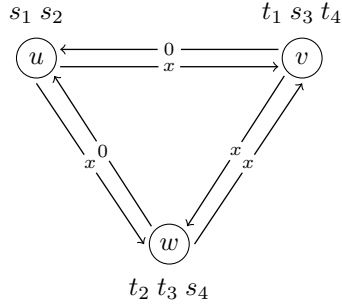


Figura 2.3: Instância atômica com demanda uniforme e custos afins que possui preço de anarquia $5/2$.

e tem custo 4. Por outro lado, se todos os jogadores escolhem o correspondente caminho de comprimento dois, obtemos um segundo equilíbrio de custo 10. Esses são os dois únicos equilíbrios. Assim, o preço da anarquia desta instância é $10/4 = 5/2$.

Mostramos a seguir um limitante para o preço da anarquia de jogos atômicos com demanda genérica e custos afins, devido a Awerbuch, Azar e Epstein [AAE05].

Teorema 2.3.8. Seja (G, r, c) uma instância de um jogo atômico na qual cada aresta e tem custo afim $c_e(x) = a_e(x) + b_e$ com $a_e, b_e \geq 0$. Então o preço da anarquia de (G, r, c) é no máximo $1 + \phi \approx 2.618$ onde $\phi = (1 + \sqrt{5})/2$ é a razão áurea.

Demonstração. Sejam f um equilíbrio e f^* uma solução ótima de (G, r, c) . Para cada jogador i suponha que P_i é a estratégia de i em f e P_i^* é a estratégia de i em f^* . O custo que o jogador i pagaria por P_i^* se mudasse sua estratégia para este é

$$\begin{aligned} \sum_{e \in P \cap P_i^*} c_e(f_e) + \sum_{e \in P_i^* \setminus P} c_e(f_e + r_i) &\leq \sum_{e \in P_i^*} c_e(f_e + r_i) \\ &= \sum_{e \in P_i^*} (a_e(f_e + r_i) + b_e). \end{aligned}$$

Como f é um equilíbrio, segue que

$$\begin{aligned}
C(f) &= \sum_{i=1}^k r_i \sum_{e \in P_i} c_e(f_e) \\
&\leq \sum_{i=1}^k r_i \sum_{e \in P_i^*} c_e(f_e + r_i) \\
&= \sum_{i=1}^k r_i \left(\sum_{e \in P_i^*} (a_e(f_e + r_i) + b_e) \right) \\
&\leq \sum_{i=1}^k r_i \left(\sum_{e \in P_i^*} [a_e(f_e + f_e^*) + b_e] \right) \\
&= \sum_{e \in E} \sum_{i: e \in P_i^*} r_i [a_e(f_e + f_e^*) + b_e] \\
&= \sum_{e \in E} [a_e(f_e + f_e^*) + b_e] f_e^* \\
&\leq C(f^*) + \sum_{e \in E} a_e f_e f_e^*. \tag{2.30}
\end{aligned}$$

Aplicando a desigualdade de Cauchy-Schwarz aos vetores $\{\sqrt{a_e} f_e\}_{e \in E}$ e $\{\sqrt{a_e} f_e^*\}_{e \in E}$ obtemos

$$\sum_{e \in E} a_e f_e f_e^* \leq \sqrt{\sum_{e \in E} a_e f_e^2} \cdot \sqrt{\sum_{e \in E} a_e (f_e^*)^2} \leq \sqrt{C(f)} \cdot \sqrt{C(f^*)}.$$

Combinando esta desigualdade com (2.30), segue que

$$\frac{C(f)}{C(f^*)} - 1 \leq \sqrt{\frac{C(f)}{C(f^*)}}.$$

Elevando os dois lados ao quadrado e resolvendo a equação quadrática $x^2 - 3x + 1 \leq 0$, obtemos

$$\frac{C(f)}{C(f^*)} \leq \frac{3 + \sqrt{5}}{2} \approx 2.618,$$

e o resultado segue. \square

O limitante do Teorema 2.3.8 é justo como mostrado pelo exemplo a seguir.

Exemplo 2.3.9. Considere a seguinte modificação das demandas do Exemplo 2.3.7 apresentada na Figura 2.4 onde $r_1 = r_2 = \phi$ e $r_3 = r_4 = 1$.

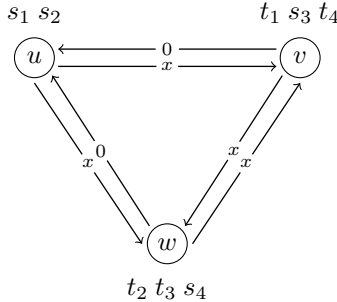


Figura 2.4: Instância atômica com custos afins que possui preço de anarquia $1 + \phi$.

Cada jogador i tem duas possíveis estratégias: escolher a aresta (s_i, t_i) ou um caminho de comprimento dois. O fluxo em que cada jogador escolhe a correspondente aresta é uma solução ótima com custo total $2\phi^2 + 2$. A solução em que cada jogador escolhe o caminho de comprimento dois é um equilíbrio de custo total $4\phi^2 + 4\phi + 2$. Assim, o preço da anarquia deste jogo é $(4\phi^2 + 4\phi + 2)/(2\phi^2 + 2) = 1 + \phi$.

2.4 Roteamento não-atômico

O modelo não-atômico de roteamento é conveniente para modelar redes grandes nas quais cada jogador tem uma demanda infinitesimal em relação à demanda total. Informalmente, cada jogador individualmente tem influência negligível no tráfego total da rede. Um exemplo típico é a Internet ou o tráfego de pedestres em uma grande metrópole.

Informalmente, a diferença entre jogos atômicos e não-atômicos é a seguinte. Em um jogo não-atômico, para cada par OD, há um número infinito (ou muito grande) de jogadores associados a este e a demanda de cada um

destes jogadores é infinitesimal se comparada com a demanda total. Assim, nenhum jogador sozinho consegue afetar o fluxo (tráfego) na rede. Já em jogos atômicos, ao menos um jogador tem demanda não desprezível e consegue influenciar o fluxo na rede¹.

Usamos $\mathcal{I} = \{1, \dots, k\}$ para denotar os diferentes pares OD. Cada par OD está associado a um número infinito de jogadores cuja demanda individual é infinitesimal e com demanda total r_i (não desprezível). Com esta convenção, f^i representa o $s_i t_i$ -fluxo que atende à demanda r_i , ou seja, ele representa o fluxo do conglomerado de jogadores associados a (s_i, t_i) , em vez de ser o fluxo do jogador i . Note que no modelo não-atômico o custo individual de cada jogador dado por (2.4) tende a zero e assim, a definição usual de equilíbrio de Nash não é apropriada.

Antes de definir formalmente uma noção mais apropriada de equilíbrio, apresentamos um exemplo que apesar de sua aparente simplicidade, é fundamental para o entendimento de jogos não-atômicos.

Exemplo 2.4.1 (Pigou [Pig20]). Considere um grafo com dois vértices s e t e duas arestas ligando s a t . A aresta superior tem custo 1 enquanto a aresta inferior tem custo igual à quantidade x de fluxo na aresta. Há um número infinito de jogadores com par OD (s, t) e a demanda total na rede é 1. Veja a Figura 2.5.

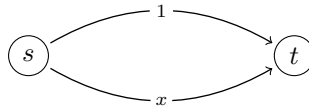


Figura 2.5: Exemplo de Pigou.

Note que a aresta inferior é mais barata se e somente se a quantidade de fluxo nela é menor que uma unidade. No (único) equilíbrio todos os jogadores escolhem a aresta inferior, com custo total 1. A solução ótima consiste em dividir a demanda total igualmente nas duas arestas, resultando em um custo de $(1/2) \cdot 1 + (1/2)(1/2) = 3/4$. Logo, o preço da anarquia deste jogo é $4/3$.

Uma pequena modificação do Exemplo de Pigou feita por Roughgarden

¹É comum dizer que um tal jogador tem *poder de mercado*.

e Tardos [RT02] mostra que o preço da anarquia pode ser arbitrariamente grande, mesmo para instâncias pequenas.

Exemplo 2.4.2. Considere um grafo com dois vértices s e t e duas arestas ligando s a t . A aresta superior tem custo 1 enquanto a aresta inferior tem custo igual a x^p com p grande. Há um número infinito de jogadores com par OD (s, t) e a demanda total na rede é 1. Veja a Figura 2.6.

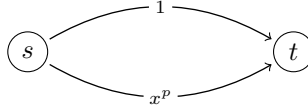


Figura 2.6: Exemplo Não-Linear de Pigou.

Como no Exemplo de Pigou, o custo do equilíbrio é 1. O fluxo ótimo envia uma quantidade $\varepsilon < 1$ pela aresta superior e tem custo total $\varepsilon + (1 - \varepsilon)^p$, onde ε tende a zero à medida que p tende a infinito. O Corolário 2.4.4, apresentado posteriormente, mostra que $\varepsilon = 1 - (p + 1)^{1/p}$. Quando p tende a infinito, o custo do fluxo ótimo tende a zero. Logo, o preço da anarquia é ilimitado.

Posteriormente, mostraremos que para certas classes de funções custo é possível limitar (ou mesmo calcular exatamente) o preço da anarquia.

2.4.1 Equilíbrio de Wardrop

A definição usual de equilíbrio de Nash não é adequada para o modelo de jogos de roteamento não-atômicos. Neles, a demanda de cada jogador é infinitesimalmente pequena e não é claro o que deveria ser seu *custo individual*, uma vez que tende a zero. Assim, em jogos deste tipo usamos o chamado *equilíbrio de Wardrop*, introduzido por Wardrop [War52].

Definição. Seja f um fluxo viável em uma instância (G, r, c) de um jogo não-atômico. Um fluxo f é um *equilíbrio* (de Wardrop) em (G, r, c) , se, para cada jogador $i \in \{1, \dots, k\}$ e cada par $P, \hat{P} \in \mathcal{P}^i$ com $f_P^i > 0$, temos que

$$c_P(f) \leq c_{\hat{P}}(f). \quad (2.31)$$

Note que em um equilíbrio de Wardrop f , todos os caminhos usados por um jogador i têm custo mínimo. Em particular, todos os caminhos usados pelo jogador têm o mesmo custo. Comparando esta definição com a de equilíbrio de Nash, notamos que o equilíbrio de Wardrop tenta capturar o fato de que a demanda de um único jogador é muito pequena. Assim, um jogador, quando pondera em uma troca de estratégia, enxerga os custos das arestas na rede como sendo fixos. Deste modo, o jogador sempre tenta escolher caminhos mínimos nesta rede.

Na próxima subseção, mostramos que todo jogo de roteamento não-atômico possui um equilíbrio de Wardrop que é *único* em um certo sentido. Daqui em diante, em qualquer lugar desta seção onde mencionamos *preço da anarquia*, referimo-nos à razão entre o custo do equilíbrio de Wardrop e o custo de uma solução ótima.

Lembre-se que supomos que as funções custo são não-negativas, contínuas e não-decrescentes.

2.4.2 Existência de equilíbrio

O objetivo desta subseção é mostrar que todo jogo de roteamento não-atômico possui um equilíbrio e que todos os equilíbrios têm o mesmo custo. A ideia básica é a mesma usada nas seções anteriores para o caso atômico: exibimos uma função potencial definida sobre os perfis de estratégias tal que o mínimo da função corresponde a um equilíbrio de Wardrop.

Inicialmente apresentamos uma caracterização de fluxos ótimos em jogos não-atômicos para uma classe particular de funções custo. Para evitar confusão, nesta caracterização usamos os símbolos l e l_e em vez de c e c_e . Suponha que cada aresta e está associada a uma função custo l_e tal que $x \cdot l_e(x)$ é diferenciável e convexa (não precisamos desta hipótese sobre c para provar o resultado principal, mas a caracterização requer isto). Note que $x \cdot l_e(x)$ é a contribuição da aresta e no custo total de um fluxo f (quando $x = f_e$). Defina o *custo marginal* de uma aresta e (em relação a l_e) como sendo

$$l_e^*(x) = (x \cdot l_e(x))' = l_e(x) + x \cdot l_e'(x).$$

Para um caminho P e um fluxo f seja

$$l_P^*(f) = \sum_{e \in P} l_e^*(f_e), \quad (2.32)$$

ou seja, a soma dos custos marginais das arestas do caminho P .

Proposição 2.4.3. Seja (G, r, l) uma instância de um jogo não-atômico na qual para toda aresta e , a função $x \cdot l_e(x)$ é diferenciável e convexa. Então f^* é um fluxo ótimo de (G, r, l) se e somente se, para todo jogador $i \in \mathcal{I}$ e todo par $P, \tilde{P} \in \mathcal{P}^i$ com $f_P^* > 0$, temos que

$$l_P^*(f^*) \leq l_{\tilde{P}}^*(f^*). \quad (2.33)$$

A demonstração segue facilmente da caracterização de soluções ótimas de problemas de otimização convexa e a omitimos. O leitor interessado pode consultar Curi [Cur13]. Note a semelhança entre a desigualdade (2.33) e a desigualdade (2.31) na definição de equilíbrio de Wardrop. A desigualdade (2.33) caracteriza fluxos ótimos em termos de custos marginais de caminhos enquanto a desigualdade (2.31) caracteriza equilíbrios de Wardrop em termos de custos de caminhos. O próximo resultado segue facilmente da Proposição 2.4.3.

Corolário 2.4.4. Seja (G, r, l) uma instância de um jogo não-atômico na qual para toda aresta e , a função $x \cdot l_e(x)$ é diferenciável e convexa. Então um fluxo viável f^* é um fluxo ótimo de (G, r, l) se e somente se é um equilíbrio em (G, r, l^*) .

Consideremos novamente a rede original (G, r, c) . Para obter uma caracterização de um equilíbrio de uma instância (G, r, c) precisamos encontrar uma função l_e tal que $c_e(x) = l_e^*(x)$ para cada $e \in E$ e aplicar o resultado anterior. Para cada $e \in E$ defina

$$l_e(x) = \frac{1}{x} \int_0^x c_e(y) dy \quad (2.34)$$

para $y > 0$ e estenda a definição para $y = 0$ por continuidade tomando $l_e(0) = c_e(0)$. Como c_e é contínua e não-decrescente, segue que $x \cdot l_e(x)$ é diferenciável e convexa. Além disso, é fácil ver que l_e é não-decrescente ($l_e(x)$ é uma média de c_e no intervalo $[0, x]$). Assim, (G, r, l) é uma instância de um jogo não-atômico. Note que o custo marginal de e (em relação a l) é $l_e^*(x) = (x \cdot l_e(x))' = c_e(x)$. Pela Proposição 2.4.3, temos que f é um equilíbrio em (G, r, c) se e somente se é um fluxo ótimo de (G, r, l) onde l é definido como em (2.34). Considere agora a função

potencial para (G, r, c) :

$$\Phi(f) = \sum_{e \in E} \int_0^{f_e} c_e(x) dx \quad (2.35)$$

definida sobre o conjunto dos fluxos viáveis de (G, r, c) . Note que $\Phi(f)$ é exatamente o custo social de f na rede (G, r, l) . Das observações anteriores, segue o próximo resultado.

Proposição 2.4.5. Seja (G, r, c) uma instância de um jogo não-atômico. Então um fluxo viável f é um equilíbrio em (G, r, c) se e somente se é o mínimo global de Φ dada por (2.35).

Note que Φ é uma função convexa, pois para cada $e \in E$, a função $c_e(x)$ é não-decrescente e portanto, cada termo $\int_0^{f_e} c_e(x) dx$ é uma função convexa.

Apresentamos agora o principal resultado desta subseção.

Teorema 2.4.6. Seja (G, r, c) uma instância de um jogo não-atômico. Então

- (a) (G, r, c) possui um equilíbrio, e
- (b) se f e \tilde{f} são equilíbrios de (G, r, c) , então $c_e(f_e) = c_e(\tilde{f}_e)$ para todo $e \in E$.

Além disso, todos os equilíbrios de um jogo não-atômico têm o mesmo custo, e portanto, o preço da anarquia é igual ao preço da estabilidade.

Demonstração. Começamos provando (a). Como cada função c_e é contínua, a função potencial Φ também é contínua nesse domínio. Logo, existe um fluxo f que minimiza $\Phi(f)$. Pela Proposição 2.4.5, segue que f é um equilíbrio de (G, r, c) .

Para verificar (b), sejam f e \tilde{f} equilíbrios em (G, r, c) . Pela Proposição 2.4.5, ambos f e \tilde{f} minimizam a função potencial Φ . Considere uma combinação convexa arbitrária $g = \lambda f + (1 - \lambda)\tilde{f}$ com $\lambda \in [0, 1]$ de f e \tilde{f} . Pela convexidade de Φ temos que

$$\Phi(\lambda f + (1 - \lambda)\tilde{f}) \leq \lambda\Phi(f) + (1 - \lambda)\Phi(\tilde{f}),$$

para todo $\lambda \in [0, 1]$. Como f e \tilde{f} são mínimos de Φ , a desigualdade acima tem que valer com igualdade para todo $\lambda \in [0, 1]$. Como cada termo

$\int_0^{f_e} c_e(x) dx$ de Φ é convexo, isto só é possível se for linear entre f_e e \tilde{f}_e . Isto implica que c_e é constante entre f_e e \tilde{f}_e . Logo, $c_e(f) = c_e(\tilde{f})$.

Por fim, note que a última afirmação no enunciado do teorema segue diretamente de (b). \square

2.4.3 Preço da anarquia de jogos não-atômicos

Para formalizar alguns dos resultados desta subseção, denotamos por \mathcal{C} o conjunto das funções custo não-negativas, contínuas e não-decrescentes. Apresentamos um limitante bastante geral para o preço da anarquia de jogos não-atômicos, devido a Correa, Schulz e Moses [CSM04]. Para entender de onde veio o limitante, veja o exemplo a seguir.

Exemplo 2.4.7. Considere um grafo com dois vértices s e t e duas arestas ligando s a t . Há um número infinito de jogadores com par OD (s, t) e a demanda total na rede é r . A aresta superior tem custo $c(r)$ (fixo) e a aresta inferior tem custo igual a $c(x)$ onde $c \in \mathcal{C}$. Veja a Figura 2.7.

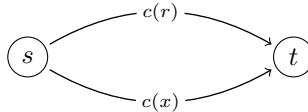


Figura 2.7: Exemplo Genérico de Pigou.

Note que para $x \leq r$ temos que $c(x) \leq c(r)$. Suponha que exista um equilíbrio de Wardrop que envia um fluxo positivo ε pela aresta superior. Neste caso, Como o custo da aresta inferior é $c(r - \varepsilon)$, devemos ter $c(r) = c(r - \varepsilon)$. Assim, o custo deste equilíbrio é $\varepsilon c(r) + (r - \varepsilon)c(r - \varepsilon) = rc(r)$. Este também é o custo do equilíbrio de Wardrop em que toda a demanda passa pela aresta inferior. Isto mostra que todos os equilíbrios têm o mesmo custo $rc(r)$. Por outro lado, o fluxo ótimo envia uma certa quantidade x pela aresta inferior e tem custo total

$$\min_{0 \leq x \leq r} \{xc(x) + (r - x)c(r)\} = rc(r) - \max_{0 \leq x \leq r} \{x(c(r) - c(x))\}.$$

Logo, o preço da anarquia desta instância é

$$\left(1 - \frac{\max_{x \in \mathbb{R}_+} \{x(c(r) - c(x))\}}{rc(r)}\right)^{-1}.$$

Para toda função $c \in \mathcal{C}$ e todo $r \in \mathbb{R}_+$, seja

$$\beta(r, c) = \frac{1}{rc(r)} \max_{x \in \mathbb{R}_+} \{x(c(r) - c(x))\},$$

onde, por convenção, $0/0 = 0$. Defina o *valor de anarquia* β como

$$\beta(\mathcal{C}) = \sup_{c \in \mathcal{C}} \sup_{r \in \mathbb{R}_+} \beta(r, c). \quad (2.36)$$

Note que devido ao Exemplo 2.4.7, segue que $(1 - \beta(\mathcal{C}))^{-1}$ é um limitante inferior para o preço da anarquia para jogos não-atômicos com funções custo em \mathcal{C} . Mostramos a seguir que $(1 - \beta(\mathcal{C}))^{-1}$ é de fato o preço da anarquia desses jogos. É interessante notar que o Exemplo de Pigou e este resultado mostram que o preço de anarquia não depende da topologia da rede mas apenas da classe de funções custo consideradas.

Antes precisamos do seguinte resultado de Roughgarden [Rou03].

Proposição 2.4.8. Seja (G, r, c) uma instância de um jogo não-atômico. Então um fluxo viável f em (G, r, c) é um equilíbrio de Wardrop 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 viável \tilde{f} em (G, r, c) .

Demonstração. Seja f um fluxo viável. Considere a função:

$$H_f(\tilde{f}) = \sum_{i=1}^k \sum_{P \in \mathcal{P}^i} c_P(f) \tilde{f}_P = \sum_{e \in E} c_e(f_e) \tilde{f}_e \quad (2.37)$$

definida sobre o conjunto dos fluxos viáveis de (G, r, c) . Considere para cada aresta e a função custo constante $l_e(x) = c_e(f_e)$. Considerando a segunda igualdade em (2.37), o valor $H_f(\tilde{f})$ é o custo total de um fluxo \tilde{f} em (G, r, l) . Assim, provar a proposição é equivalente a mostrar que f é um equilíbrio de Wardrop em (G, r, c) se e somente se minimiza H_f .

Lembre-se que na definição de equilíbrio de Wardrop, se f é um equilíbrio, cada jogador escolhe um caminho de custo mínimo onde cada aresta tem custo $c_e(f_e)$ nesta rede. Por (2.37), um fluxo \tilde{f} minimiza $H_{\tilde{f}}$ se e somente se, para todo i , temos que $\tilde{f}_P > 0$ apenas para caminhos P em \mathcal{P}^i que minimizem $c_P(f)$, ou seja tenha custo mínimo na rede. Esta é exatamente a condição de equilíbrio de Wardrop e o resultado segue. \square

Teorema 2.4.9. Seja (G, r, c) uma instância de um jogo não-atômico com custos em \mathcal{C} . Então o preço de anarquia de (G, r, c) é no máximo $(1 - \beta(\mathcal{C}))^{-1}$.

Demonstração. Por simplicidade, denote $\beta = \beta(\mathcal{C})$. Sejam f um equilíbrio e f^* uma solução ótima. A seguir, mostramos que

$$C(f) \leq C(f^*) + \beta C(f), \quad (2.38)$$

o que implica o resultado.

Pela Proposição 2.4.8, temos que

$$\begin{aligned} \sum_{e \in E} c_e(f_e) f_e &\leq \sum_{e \in E} c_e(f_e) f_e^* \\ &= \sum_{e \in E} c_e(f_e^*) f_e^* + \sum_{e \in E} (c_e(f_e) - c_e(f_e^*)) f_e^* \\ &\leq \sum_{e \in E} c_e(f_e^*) f_e^* + \sum_{e \in E} \left[\max_{x \in \mathbb{R}_+} \frac{(c_e(f_e) - c_e(x))x}{c_e(f_e) f_e} \right] \cdot c_e(f_e) f_e \\ &= C(f^*) + \sum_{e \in E} \beta(f_e, c_e) \cdot c_e(f_e) f_e \\ &\leq C(f^*) + \sum_{e \in E} \left[\sup_{c \in \mathcal{C}} \sup_{x \in \mathbb{R}_+} \beta(x, c) \right] \cdot c_e(f_e) f_e \\ &= C(f^*) + \beta(\mathcal{C}) \sum_{e \in E} c_e(f_e) f_e \\ &= C(f^*) + \beta(\mathcal{C}) C(f), \end{aligned}$$

o que conclui a prova. \square

A seguir descrevemos duas consequências do Teorema 2.4.9. Omitimos as provas destes resultados.

Teorema 2.4.10. Seja (G, r, c) um jogo não-atômico com funções de custo afim. Então o preço da anarquia de (G, r, c) é $4/3$.

Teorema 2.4.11. Seja (G, r, c) um jogo não-atômico com funções de custo que são polinômios de grau máximo d . Então o preço da anarquia de (G, r, c) é $\Theta\left(\frac{d}{\ln d}\right)$.

Balanceamento de Carga

O problema de balanceamento de carga é um problema que ocorre em sistemas computacionais e foi bastante investigado na literatura como um problema de otimização em escalonamento de tarefas. Em tais problemas, temos um conjunto de máquinas $M = \{1, \dots, m\}$ e um conjunto de tarefas $N = \{1, \dots, n\}$ que devem ser atribuídas às máquinas. Cada tarefa $i \in N$ tem um peso w_{ij} associado a cada máquina $j \in M$, que pode ser visto como o tempo para se processar ou executar a tarefa i na máquina j .

Uma atribuição $A: N \rightarrow M$ é uma alocação de cada tarefa para uma das máquinas. Dada atribuição A , denotamos o conjunto de tarefas atribuídas à uma máquina $j \in M$ como A_j . A *carga* de uma máquina j é a soma dos pesos das tarefas atribuídas a ela, que denotamos por

$$\ell_j(A) = \sum_{i \in A_j} w_{ij}.$$

Quando a atribuição A estiver clara pelo contexto, escrevemos apenas ℓ_j para denotar $\ell_j(A)$. O custo da atribuição A , chamado de *makespan* e que denotamos por $c(A)$, é dado pela maior carga em uma máquina, i.e.,

$$c(A) = \max_{j \in M} \ell_j(A).$$

O custo de uma tarefa i é dado pela carga da máquina $A(i)$. Assim, cada jogador tem como custo a carga da máquina onde sua tarefa foi alocada.

Em sua forma mais geral, o problema de balanceamento de carga lida com *máquinas não-relacionadas*. Neste caso, não impomos nenhuma

restrição em relação aos valores w_{ij} . Porém, neste capítulo, estamos interessados em dois casos particulares. Em ambos os casos, cada tarefa i é representada por um único valor w_i que representa o seu peso. No caso de *máquinas uniformes*, todas as máquinas são idênticas. Assim, para toda tarefa i , temos que $w_{ij} = w_i$ para qualquer máquina j . Já no caso de *máquinas relacionadas*, as máquinas distinguem-se apenas por sua velocidade de processamento. Cada máquina j tem velocidade s_j , de forma que, para toda tarefa i e máquina j , temos que $w_{ij} = w_i/s_j$. Note que o caso de máquinas uniformes é um caso particular de máquinas relacionadas onde toda máquina j tem velocidade $s_j = 1$.

O problema de balanceamento de carga na abordagem de otimização faz parte dos problemas de escalonamentos, que estão entre os problemas mais investigados na área de Computação [KSW97]. Trata-se de um problema NP-difícil mesmo para o caso onde há apenas duas máquinas uniformes [GJ79].

Quando consideramos o balanceamento de carga dentro da ótica da teoria de jogos, podemos ter jogadores que controlam tarefas e/ou máquinas. Para este capítulo, damos enfoque à versão onde cada tarefa é controlada por um jogador, as máquinas são uniformes ou relacionadas e o custo social de uma atribuição será dado pelo seu *makespan*. Desta forma, consideramos que a carga de uma tarefa $i \in N$ é dada apenas por seu peso w_i e cada máquina $j \in M$ possui uma velocidade s_j . Com isso, uma entrada para este jogo pode ser dado por uma tupla $J = (n, m, w, s)$. Assim, o conjunto de estratégias de um jogador é dado pelo conjunto de máquinas, onde cada jogador procura minimizar seu custo. Ademais, dizemos que uma atribuição é ótima se seu *makespan* é mínimo. Aplicações deste jogo ocorrem na transferência de dados na Internet e em sistemas dinâmicos [EDKM07, ORS93].

3.1 Preliminares

Nesta seção, apresentamos algumas definições e resultados que valem tanto para o caso de máquinas uniformes quanto máquinas relacionadas.

Um perfil de estratégias está em equilíbrio de Nash se não há incentivo para nenhum jogador mudar a máquina onde sua tarefa está alocada.

Proposição 3.1.1. Uma atribuição A é um equilíbrio de Nash se e somente

se para toda tarefa $i \in N$, temos que

$$\ell_{A(i)} \leq \ell_k + \frac{w_i}{s_j},$$

para toda máquina $k \in M$.

A seguir apresentamos um exemplo retirado de Nisan et al. [NRTV07] onde o *makespan* de um equilíbrio é pior do que o *makespan* de uma atribuição ótima.

Exemplo 3.1.2. Considere um jogo de balanceamento de carga onde temos duas máquinas $M = \{1, 2\}$ uniformes e quatro tarefas $N = \{1, 2, 3, 4\}$, duas com peso 1 e duas com peso 2. A Figura 3.1 apresenta as únicas duas atribuições deste jogo em equilíbrio de Nash, a menos de simetrias.

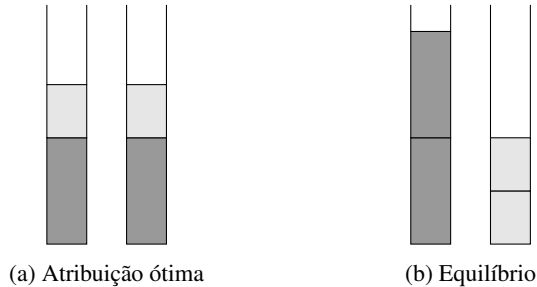


Figura 3.1: Uma instância do jogo de balanceamento de carga com duas máquinas e quatro tarefas, sendo que as duas tarefas mais claras têm peso 1 e as duas tarefas mais escuras têm peso 2.

A configuração (a) é uma atribuição ótima em equilíbrio, onde cada máquina tem uma tarefa de peso 2 e outra de peso 1. A configuração (b), que apresenta a primeira máquina com duas tarefas de peso 2 e a segunda com duas tarefas de peso 1, também é uma configuração em equilíbrio de Nash. Nesta configuração, não há incentivo para migrar a tarefa de peso 1 para a primeira máquina, pois esta tem carga pior que sua atual máquina. E não há incentivo para um jogador migrar uma tarefa de peso 2, da primeira máquina para a segunda máquina, pois caso migrasse ficaria com o mesmo custo. Além disso, qualquer atribuição com custo maior que 4 não está em equilíbrio de Nash. Assim, o preço da anarquia deste jogo é $4/3$.

Dizemos que um jogador (ou tarefa) está *satisfeito* se ele não pode reduzir seu custo migrando sua tarefa de uma máquina para outra. Dada uma atribuição inicial, um jogador insatisfeito com a atribuição pode migrar sua tarefa de uma máquina para outra onde obterá um custo menor. O seguinte teorema mostra que o jogo de balanceamento de carga em máquinas uniformes sempre converge para um equilíbrio de Nash se jogadores insatisfeitos migram para máquinas melhores.

Teorema 3.1.3. O jogo de balanceamento de carga com máquinas relacionadas sempre converge para um equilíbrio de Nash, independente das políticas usadas para a escolha de tarefas e máquinas.

Demonstração. Considere uma atribuição A qualquer e seja $\lambda(A) = (\lambda_1, \dots, \lambda_m)$ o vetor de cargas das máquinas da atribuição A , em ordem não-crescente de carga ($\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$). Considere um jogador i insatisfeito. Vamos mostrar que, após a migração de i , a nova atribuição A' é tal que $\lambda(A')$ é lexicograficamente menor que $\lambda(A)$.

Por simplicidade, renomeie as máquinas de acordo com a ordem lexicográfica, isto é, a carga da máquina $j \in M$ é precisamente λ_j . Suponha que i está em uma máquina j e migra para uma máquina k . Assim, a carga de k com a tarefa i é menor do que a carga de j , isto é, $\lambda_k + w_i/s_k < \lambda_j$. A migração diminui a carga de j e aumenta a carga de k , mas a nova carga de k é ainda estritamente menor que a carga antiga de i . Ademais, $j < k$, já que independente da velocidade das máquinas, nunca é vantajoso migrar para uma máquina com carga maior do que a carga de j . Note que apenas as máquinas j e k terão suas cargas modificadas após o movimento do jogador i . Como o número de máquinas com carga pelo menos λ_j continua o mesmo e a carga λ_j diminui, temos que $\lambda(A')$ é lexicograficamente menor do que $\lambda(A)$.

Para concluir a prova, note que, como o número de atribuições possíveis é limitado (por m^n), existe uma atribuição A^* lexicograficamente mínima. Ademais, A^* é um equilíbrio de Nash, já que não existe jogador insatisfeito. Portanto, o jogo converge a um equilíbrio de Nash em uma quantidade finita de passos. \square

Note que, ao migrar uma tarefa de uma máquina para outra, a carga na máquina de origem diminui e na de destino aumenta, mas esta não passa a carga da máquina de origem antes da migração. Ou seja, a migração de um jogador insatisfeito para uma máquina melhor nunca aumenta o *makespan*.

Assim, se a atribuição inicial for ótima, o jogo converge para um equilíbrio também de custo mínimo. O seguinte resultado é uma consequência direta deste fato.

Corolário 3.1.4. O preço da estabilidade do jogo de balanceamento de carga é 1.

Assim, existe uma atribuição de custo social mínimo que também é um equilíbrio de Nash. Caso tal equilíbrio fosse apresentado para os jogadores, eles não teriam motivo para desviar da atribuição sugerida. Assim, tal atribuição não só é benéfica para os jogadores individualmente mas também para a sociedade como um todo.

Porém, como encontrar uma atribuição ótima na versão de otimização do balanceamento de carga é um problema NP-difícil mesmo para duas máquinas, temos o seguinte resultado.

Teorema 3.1.5. Encontrar um equilíbrio de Nash de custo social mínimo para o jogo de balanceamento de carga com duas máquinas uniformes é um problema NP-difícil.

Apesar deste jogo sempre convergir para um equilíbrio de Nash, o número de rodadas necessárias para isso pode ser exponencialmente maior que o tamanho da entrada [EDKM07, FKK⁺09]. Por exemplo, a prova do Teorema 3.1.3 apenas garante que a convergência se dá em $O(m^n)$ passos. Por outro lado, existem algumas políticas simples que podem limitar consideravelmente o número de rodadas necessárias para a convergência. A quantidade de rodadas necessárias se atingir um equilíbrio de Nash é chamado de *tempo de convergência* do jogo.

A seguir analisamos o tempo de convergência do jogo e o preço da anarquia, considerando primeiro o jogo com máquinas uniformes e posteriormente para máquinas relacionadas.

3.2 Jogo com máquinas uniformes

Nessa seção, abordamos jogos de balanceamento de carga com máquinas uniformes. A seguir, analisamos o tempo de convergência para este jogo e limitantes para o preço da anarquia, mostrando que o custo de qualquer equilíbrio de Nash neste jogo é menor que 2, e há jogos onde o preço da anarquia pode chegar tão próximo de 2 quanto se queira.

3.2.1 Tempo de convergência

A seguir, consideramos duas políticas de migração de tarefas insatisfeitas. Em ambas, uma tarefa insatisfeita sempre migra para uma máquina onde obterá o menor custo, isto é, o jogador muda de estratégia para uma resposta ótima. Assim, chamamos tais políticas de *políticas de resposta ótima*. Na *política de resposta ótima das tarefas mais leves*, a cada passo, uma das tarefas mais leves migra para uma máquina que seja uma resposta ótima. De forma similar, definimos a *política de resposta ótima das tarefas mais pesadas*.

Even-dar, Kesselman e Mansour [EDKM07], mostram que mesmo quando as máquinas são uniformes, a política de resposta ótima das tarefas mais leves pode precisar de um número exponencial de passos para se atingir um equilíbrio, como descrito no teorema seguinte.

Teorema 3.2.1. Existe uma instância do jogo de balanceamento de cargas com n tarefas e m máquinas uniformes onde a política de resposta ótima das tarefas mais leves requer pelo menos $\left(n/(m-1)^2\right)^{m-1}$ passos para atingir um equilíbrio de Nash.

Porém, se considerarmos a política de resposta ótima das tarefas mais pesadas, para qualquer atribuição inicial, é possível chegar a um equilíbrio no jogo de balanceamento de carga rapidamente [FH79].

Teorema 3.2.2. Para qualquer atribuição inicial do jogo de balanceamento de carga com n tarefas e m máquinas, a política de resposta ótima das tarefas mais pesadas converge para um equilíbrio de Nash em no máximo n passos.

Demonstração. Seja ℓ_j^t a carga da máquina j no tempo t e considere que uma tarefa i migra de j para j^* no tempo t .

Considere uma tarefa k que estava satisfeita e se tornou insatisfeita após a migração de i no tempo t . Note que k se tornou insatisfeita ou porque a carga da máquina j diminuiu (e, portanto, k não está em j) ou porque a carga da máquina j^* aumentou (e, portanto, k está na máquina j^*).

Se k está em uma máquina j' que não sofreu alteração, então

$$\ell_{j'}^t = \ell_{j'}^{t+1} \quad (3.1)$$

$$> \ell_j^{t+1} + w_k \quad (3.2)$$

$$= \ell_j^t - w_i + w_k \quad (3.3)$$

$$> \ell_{j^*}^t + w_k. \quad (3.4)$$

A igualdade (3.1) vale pois a carga da máquina j' não mudou do tempo t para o tempo $t + 1$. A desigualdade (3.2) vale pois k prefere a máquina j no tempo $t + 1$ (pois sua carga foi reduzida). A igualdade (3.3) vale pois i migrou de j no tempo t . Por fim, a desigualdade (3.4) vale pois i prefere a máquina j^* no tempo t . Concluimos que k já estava insatisfeita no tempo t .

Por outro lado, se k está em j^* e deseja migrar para a máquina j' no tempo $t + 1$, então

$$\ell_{j'}^{t+1} + w_k < \ell_{j^*}^{t+1} = \ell_{j^*}^t + w_i.$$

Se $j' = j$, então, como i prefere a máquina j^* e i migrou de j no tempo t , temos que

$$\ell_{j^*}^t + w_i < \ell_{j'}^t = \ell_{j'}^{t+1} + w_i.$$

Se $j' \neq j$, então, como a máquina j^* é uma resposta ótima para i e i não estava em j' no tempo t , temos que

$$\ell_{j^*}^t + w_i \leq \ell_{j'}^t + w_i = \ell_{j'}^{t+1} + w_i.$$

Em ambos os casos, concluímos que $w_k < w_i$.

Com isso, concluímos que quando uma tarefa migra, torna insatisfeitas apenas tarefas com peso menor. Isto é, após migrar uma tarefa, ela nunca mais fica insatisfeita. Portanto, cada tarefa migra no máximo uma vez e o equilíbrio neste caso é atingido em no máximo n passos. \square

3.2.2 Preço da anarquia

Agora, vamos considerar o preço da anarquia para jogos de balanceamento de carga com máquinas uniformes.

O jogo de balanceamento de carga foi investigado por Finn e Horowitz [FH79] como um problema de escalonamento no qual uma atribuição é iterativamente melhorada movendo-se uma tarefa por vez. Da

análise, obtém-se que o preço da anarquia do jogo de balanceamento de carga é menor que 2, como apresentado no teorema seguinte.

Teorema 3.2.3. O preço da anarquia do jogo de balanceamento com n tarefas e m máquinas é no máximo $2 - \frac{2}{m+1}$.

Demonstração. Seja A uma atribuição em equilíbrio e $c(A)$ o custo social desta atribuição. Seja j^* uma máquina com carga máxima, isto é, $c(A) = \ell_{j^*}$. Se j^* tem apenas uma tarefa, o resultado é claramente válido. Assim, vamos considerar que j^* tem pelo menos duas tarefas. Seja i uma tarefa de menor peso na máquina j^* . Como há pelo menos duas tarefas em j^* e i é de peso mínimo, temos que $w_i \leq \ell_{j^*}/2$.

Agora, considere uma máquina j diferente de j^* . Como a atribuição A está em equilíbrio, temos que $\ell_j \geq \ell_{j^*} - w_i$ para toda máquina j (caso contrário, a tarefa i poderia migrar para j). Com isso, temos que

$$\ell_j \geq \ell_{j^*} - w_i \geq \ell_{j^*} - \frac{\ell_{j^*}}{2} = \frac{1}{2}\ell_{j^*} = \frac{1}{2}c(A).$$

Seja então A^* uma atribuição de *makespan* mínimo. Usando o fato que $c(A^*) \geq \sum_{i=1}^n w_i/m$, temos

$$\begin{aligned} c(A^*) &\geq \frac{\sum_{i=1}^n w_i}{m} \\ &= \frac{\sum_{j=1}^m \ell_j}{m} = \frac{\ell_{j^*} + \sum_{j \in M \setminus \{j^*\}} \ell_j}{m} \\ &\geq \frac{c(A) + \sum_{j \in M \setminus \{j^*\}} \frac{1}{2}c(A)}{m} \\ &= \frac{(m+1)c(A)}{2m}. \end{aligned}$$

Isolando $c(A)$, temos o limitante desejado para o preço da anarquia. \square

Observamos também que o preço da anarquia apresentado no Teorema 3.2.3 é justo [NRTV07, Cap. 20].

Proposição 3.2.4. Para cada valor de m , existe uma instância onde o preço da anarquia no jogo de balanceamento de carga é exatamente $2 - \frac{2}{m+1}$.

3.3 Jogos com máquinas relacionadas

Nessa seção, abordamos jogos de balanceamento de carga com máquinas relacionadas. Infelizmente, para tais jogos, o preço da anarquia é substancialmente maior do que o preço da anarquia para o caso com máquinas uniformes. Ademais, não se sabe se existe uma política de migração de tarefas insatisfeita que sempre convirja para um equilíbrio em um número polinomial de migrações. Assim, tais jogos são mais complicados do que a versão com máquinas uniformes. De fato, lembre-se que a versão com máquinas uniformes é um caso particular de máquinas relacionadas, todas com a mesma velocidade.

3.3.1 Tempo de convergência

Como mencionado anteriormente, não é claro se sempre existe uma sequência de migrações de jogadores insatisfeitos para máquinas melhores que leve a um equilíbrio de Nash em um número polinomial (em $n + m$) de passos. O que se sabe, é que podemos construir uma atribuição em equilíbrio, através do algoritmo LPT (*Largest Processing Time*). Este é um algoritmo para escalonamento de tarefas que atribui as tarefas em ordem não-crescente de peso. Em cada atribuição, a tarefa é alocada na máquina que minimiza seu custo, no momento da inserção. O seguinte teorema mostra que a atribuição produzida pelo algoritmo LPT é um equilíbrio de Nash [FKK⁺09].

Teorema 3.3.1. O algoritmo LPT produz uma atribuição em equilíbrio de Nash para máquinas relacionadas.

Demonstração. Considere que as tarefas já estão ordenadas na ordem em que serão alocadas. Vamos provar o resultado por indução no número de tarefas alocadas. O resultado é trivialmente válido para a base. Assim, considere a iteração t , logo após a atribuição da tarefa t . Pela hipótese de indução, as tarefas $1, \dots, t - 1$ estavam satisfeitas no tempo $t - 1$. Seja j^* a máquina onde t foi atribuída pelo algoritmo LPT. Note que apenas tarefas alocadas a j^* podem se tornar insatisfeitas, já que tarefas em outras máquinas estavam previamente satisfeitas e após a atribuição de t , houve aumento na carga apenas de j^* .

Seja $i < t$ uma tarefa alocada a j^* e, para toda máquina j , seja ℓ_j^t a carga da máquina j após a atribuição da tarefa t . Para toda máquina j ,

como t prefere a máquina j^* à máquina j , $w_t \leq w_i$ e $A(t) = j^*$, temos que

$$\ell_{j^*}^t \leq \ell_j^t + \frac{w_t}{s_j} \leq \ell_j^t + \frac{w_i}{s_j}.$$

Logo, i está satisfeito e a atribuição está em equilíbrio. \square

3.3.2 Preço da anarquia

Diferente da versão com máquinas uniformes, o jogo de balanceamento de carga com máquinas relacionadas não apresenta preço da anarquia constante. Porém, o mesmo é limitado por uma função assintoticamente menor que a logarítmica no número de máquinas. De fato, o preço da anarquia do jogo de balanceamento de carga com máquinas relacionadas é $\Theta(\lg m / \lg \lg m)$, onde m é o número de máquinas [CV02].

Teorema 3.3.2. Para todo jogo de balanceamento de cargas com máquinas relacionadas $J = (n, m, w, s)$ e toda atribuição $E: N \rightarrow M$ em equilíbrio, vale que

$$c(E) = O\left(\frac{\lg m}{\lg \lg m}\right) \text{OPT}(J).$$

Ademais, existe um jogo com m máquinas relacionadas e uma atribuição A em equilíbrio tal que

$$c(A) = \Omega\left(\frac{\lg m}{\lg \lg m}\right) \text{OPT}(J).$$

3.4 Estratégias mistas

No caso do jogo de balanceamento de carga, podemos considerar também estratégias mistas, onde cada estratégia mista é uma distribuição de probabilidade no conjunto de máquinas.

Uma estratégia mista para um jogador $i \in N$ é um vetor p_i onde p_i^j é a probabilidade da tarefa i ser alocada na máquina j . Um perfil de estratégias $P = (p_i^j)_{i \in N, j \in M}$ é dado pelas probabilidades de cada jogador para cada máquina e induz uma atribuição aleatória. Denote por x_i^j a variável aleatória binária que indica se a tarefa i é alocada na máquina j e A esta atribuição aleatória. Assim, $p_i^j = \mathbb{P}[x_i^j = 1]$, ou equivalentemente,

$p_i^j = \mathbb{P}[A(i) = j]$. Com isso, a carga esperada, da máquina j dentro do perfil de estratégia P é dado por

$$\mathbb{E}[\ell_j] = \mathbb{E}\left[\sum_{i \in N} \frac{w_i x_i^j}{s_j}\right] = \sum_{i \in N} \frac{w_i \mathbb{E}[x_i^j]}{s_j} = \sum_{i \in N} \frac{w_i p_i^j}{s_j}.$$

O custo social do perfil de estratégias P é definido como o *makespan* esperado, dado por

$$c(P) = \mathbb{E}[c(A)] = \mathbb{E}\left[\max_{j \in M}(\ell_j)\right].$$

Uma vez que cada jogador procura minimizar seu custo, o custo esperado calculado pelo usuário i ao atribuir sua tarefa na máquina j é dado por $c^j(i)$, onde

$$c^j(i) = \frac{w_i + \sum_{k \neq i} w_k p_k^j}{s_j} = \mathbb{E}[\ell_j] + (1 - p_i^j) \frac{w_i}{s_j}.$$

Como nos jogos mistos o equilíbrio de Nash é definido através do valor esperado, a seguinte proposição é válida.

Proposição 3.4.1. Um perfil de estratégias P está em equilíbrio de Nash se e somente se para toda tarefa $i \in N$ e toda máquina $j \in M$ tal que $p_i^j > 0$, segue que $c^j(i) \leq c^k(i)$ para toda máquina $k \in M$.

Como todo equilíbrio puro é um equilíbrio misto, temos que o custo social do pior equilíbrio misto é maior ou igual ao custo social do pior equilíbrio puro. De fato, mesmo para máquinas uniformes, o custo social do pior equilíbrio misto pode ser muito pior do que o custo social do pior equilíbrio puro, como mostrado por Czumaj e Vöcking [CV02].

Teorema 3.4.2. Para todo jogo de balanceamento de cargas com máquinas uniformes $J = (n, m, w)$ e toda atribuição E em equilíbrio, vale que

$$c(E) = O\left(\frac{\lg m}{\lg \lg m}\right) \text{OPT}(J).$$

Ademais, Koutsoupias e Papadimitriou [KP99] mostraram que existe um jogo com m máquinas relacionadas e uma atribuição A tal que

$$c(A) = \Omega\left(\frac{\lg m}{\lg \lg m}\right) \text{OPT}(J).$$

A piora no preço da anarquia quando consideramos estratégias mistas também para o caso de máquinas relacionadas [CV02].

Teorema 3.4.3. Para todo jogo de balanceamento de cargas com máquinas relacionadas $J = (n, m, w, s)$ e toda atribuição E em equilíbrio, vale que

$$c(E) = O\left(\frac{\lg m}{\lg \lg \lg m}\right) \text{OPT}(J).$$

Ademais, existe um jogo com m máquinas relacionadas e e uma atribuição A tal que

$$c(A) = \Omega\left(\frac{\lg m}{\lg \lg \lg m}\right) \text{OPT}(J).$$

3.5 Mecanismos de coordenação

Nas seções anteriores, notamos o quanto o sistema se deteriora na presença de jogadores racionais, através da análise do preço da anarquia. Esta medida captura a falta de coordenação entre jogadores, que agem conforme o seu próprio interesse em um sistema descentralizado (onde não há um sistema que estipule qual tarefa será alocada em qual máquina).

Nesta seção, mostramos que é possível projetar mecanismos que incentivam a coordenação dos jogadores de maneira a obter equilíbrios com bons valores de preço da anarquia. A seção é baseada nos trabalhos de Christodoulou, Koutsoupias e Nanavati [CKN04] que introduziram a noção de mecanismos de coordenação.

Abordamos este modelo para o jogo de balanceamento de carga com máquinas uniformes e, posteriormente, apresentamos alguns resultados para o caso de máquinas relacionadas.

O jogo é semelhante ao apresentado nas seções anteriores, porém, neste jogo, os custos para jogadores com tarefas em uma mesma máquina pode diferir. Ademais, cada máquina tem uma política de atribuição de custos às tarefas e cada jogador tem conhecimento das políticas de atribuição de custos aplicadas por cada máquina. Neste modelo, a política de atribuição de cada máquina considera apenas as informações das tarefas atribuídas a ela.

O conjunto de políticas de atribuição das máquinas formam um mecanismo de coordenação do jogo, cujo objetivo é incentivar a coordenação

por parte dos jogadores buscando limitar o custo social da atribuição dentro de valores próximos do ótimo social. De fato, apresentamos mecanismos de coordenação com preço da anarquia limitados por constantes.

Consideramos que o custo de uma tarefa é exatamente o tempo em que a tarefa termina sua execução na máquina em que foi atribuída. Com isso, um mecanismo pode priorizar tarefas atribuídas a uma máquina, em detrimento de outras, ou mesmo atrasar seu tempo de finalização, mesmo que a máquina esteja livre. É justamente a capacidade de dar tratamento diferenciado às tarefas em uma mesma máquina, que exploramos nos mecanismos de coordenação.

A política de atribuição usada por uma máquina j é dada pelo tempo de término de cada tarefa atribuída a j . Denotamos por $c^j : N \rightarrow \mathbb{R}$, a função de custo da máquina j , para cada tarefa de uma máquina. Assim, $c^j(i)$ é o tempo de término da tarefa i , caso esta seja atribuída à máquina j , e igual a 0, caso contrário. Podemos definir um *mecanismo de coordenação* como uma sequência de funções de custo $\mathcal{C} = (c^1, \dots, c^m)$, uma função para cada máquina. Ademais, consideramos que, para toda atribuição A , temos

$$\max\{c^j(i) : i \in S\} \geq \sum_{i \in S} w_{ij}, \quad (3.5)$$

para toda máquina $j \in M$ e todo conjunto $S \subseteq A_j$.

Note que a condição acima permite que uma máquina atrase o tempo de término de uma tarefa ou mesmo possua tempo ocioso entre duas tarefas consecutivas atribuídas a ela.

Exemplo 3.5.1. Considere uma máquina j e duas tarefas i e i' atribuídas a j . Pela restrição (3.5), temos que $c^j(i) \geq w_{ij}$ e $c^j(i') \geq w_{i'j}$, isto é, cada tarefa tem um custo de pelo menos o seu peso. Ademais, temos que $\max\{c^j(i), c^j(i')\} \geq w_{ij} + w_{i'j}$. Assim, uma opção seria escolher $c^j(i) = w_{ij}$ e $c^j(i') = w_{ij} + w_{i'j}$. Isto é equivalente a processar primeiro a tarefa i (de forma que ela termina no tempo w_{ij}) e, imediatamente depois, processar a tarefa i' (de forma que ela termina no tempo $w_{i'j}$).

Em particular, anteriormente consideramos a função de custo $c^j(i) = \sum_{i \in S} w_{ij}$ que satisfaz a restrição (3.5). Porém, esta política não dá tratamento diferenciado às tarefas de uma mesma máquina. De fato, note que em tal função de custo, todas as tarefas terminam ao mesmo tempo, como se fossem executadas em paralelo.

Um ponto importante nos mecanismos de coordenação que consideramos, é que eles são determinísticos. Para que haja distinção entre duas tarefas de mesmo peso, consideramos que a entrada define uma ordem nas tarefas, razão pela qual utilizamos uma n -upla como entrada para as funções de custo, e com isso as tarefas são desempatadas pela ordem lexicográfica de seus identificadores.

Dado um mecanismo de coordenação $\mathcal{C} = (c^1, \dots, c^m)$, como parte do jogo de balanceamento de cargas para cálculo dos custos das tarefas, denote por $\mathcal{E}(\mathcal{C}, w)$ o conjunto de atribuições do jogo em equilíbrio de Nash, utilizando o mecanismo de coordenação \mathcal{C} . O custo de uma atribuição A para o mecanismo \mathcal{C} , denotado por $c(A)$ é o custo máximo de um jogador, isto é, $c(A) = \max_{i \in N} \{c^j(i) : A(i) = j\}$. Definimos o preço da anarquia de um mecanismo de coordenação $\mathcal{C} = (c^1, \dots, c^m)$, como

$$PA(\mathcal{C}) = \max_w \max_{A \in \mathcal{E}(\mathcal{C}, w)} \frac{c(A)}{OPT(w)},$$

onde $OPT(w)$ é o *makespan* de uma atribuição ótima para w . Note que $OPT(w)$ independe do mecanismo de coordenação.

Dizemos que um mecanismo de coordenação é simétrico se todas as máquinas utilizam a mesma política de alocação. Infelizmente, todo mecanismo de coordenação simétrico tem preço da anarquia (considerando equilíbrios mistos) pelo menos $\Omega(\lg m / \lg \lg m)$ [CKN04], assim, estamos interessado em mecanismos de coordenação assimétricos.

Para o próximo exemplo e na definição de um mecanismo de coordenação para m máquinas e n jogadores, utilizamos uma ordenação L , das tarefas de N , em ordem não-decrescente dos pesos, desempatando pela ordem lexicográfica dos identificadores das tarefas, e a relação de ordem $i' \prec i''$ (ou de maneira equivalente, $i'' \succ i'$) se i' aparece antes de i'' em L . Por simplicidade, dizemos que as tarefas estão em ordem crescente (resp. decrescente) dos pesos se satisfazem a ordem dada por \prec (resp. \succ). O seguinte exemplo, apresentado por Christodoulou et al. [CKN04], apresenta um mecanismo de coordenação com preço da anarquia pelo menos $4/3$ quando temos pelo menos quatro jogadores.

Exemplo 3.5.2. Considere duas máquinas uniformes, com o seguinte mecanismo de coordenação: cada máquina ordena as tarefas que lhe foram atribuídas e executa uma após a outra, na ordem dada pela ordenação. O custo de cada tarefa é o tempo do término de sua execução.

A primeira (resp. segunda) máquina ordena as tarefas que lhe foram atribuídas, colocando-as na ordem decrescente em relação a \succ (resp. crescente em relação a \prec) dos pesos.

Este mecanismo de coordenação tem preço da anarquia 1 quando $n \leq 3$ e pelo menos $4/3$ quando $n \geq 4$ como mostramos a seguir. Para isto, vamos dividir em casos.

É direto verificar que o resultado é válido quando $n = 1$. Quando $n = 2$, digamos com pesos $w_1 \succ w_2$ o primeiro jogador irá colocar na primeira máquina, que ordena os pesos de maneira decrescente (priorizando o primeiro jogador), e o segundo jogador na segunda máquina, que ordena os pesos de maneira crescente (priorizando o segundo jogador), obtendo uma atribuição ótima. Note que caso o primeiro jogador também coloque sua tarefa na segunda máquina, seu tempo de término seria $w_1 + w_2$. O mesmo é válido para o caso do segundo jogador colocar sua tarefa na primeira máquina.

Considere $n = 3$ tarefas, com pesos $w_1 \succ w_2 \succ w_3$. Com isso, o primeiro jogador colocará sua tarefa na primeira máquina, se assegurando que terá o melhor tempo de execução neste caso. Já o terceiro jogador, colocará sua tarefa na segunda máquina obtendo o menor custo possível. Analisando desta forma, o segundo jogador coloca sua tarefa na segunda máquina, que não será pior que colocar na primeira máquina. Assim, esta atribuição é ótima.

Para mostrar o limitante inferior do preço da anarquia para $n \geq 4$, considere quatro tarefas com pesos $w_1 = 2 + \varepsilon$, $w_2 = 2$, $w_3 = 1 - \varepsilon$ e $w_4 = 1$, para um valor arbitrariamente pequeno de ε . Analisando da mesma forma, há apenas uma atribuição em equilíbrio, com a primeira tarefa na primeira máquina e as demais tarefas na segunda máquina, dando um custo social de $4 - \varepsilon$. Na atribuição ótima, a primeira e terceira tarefas ficam em uma máquina e as demais na outra, dando custo de 3, e portanto com preço da anarquia $(4 - \varepsilon)/3$, que pode ficar tão próximo de $4/3$ quanto se queira, a medida que ε tende a zero.

Considere o caso geral onde temos m máquinas e n tarefas. Para isso, vamos usar o seguinte mecanismo para quebrar as simetrias:

- (a) Cada máquina escalona as tarefas em ordem não-crescente do seu peso (usando a ordem lexicográfica para quebrar eventuais empates).
- (b) Cada máquina j acrescenta um atraso, se necessário, em cada tarefa

atribuída a ela, de maneira que esta termine na próxima unidade de tempo $t = j \pmod{m+1}$.

A ordem definida na parte (a) do mecanismo nos dá uma ordenação única das tarefas. Na parte (b), apesar do mecanismo piorar uma solução com atrasos, é possível limitar o atraso total de uma máquina para valores bem pequenos. Para isto, basta usar unidades de tempo tão menores quanto se queira. Note que para o atraso total em uma máquina ser no máximo uma constante ε , basta usar unidades de tempo da forma $\varepsilon/(mn)$. Considerando que o maior número de tarefas em uma máquina é n e cada uma pode aumentar o tempo total em no máximo m unidades de tempo, temos um atraso total de no máximo nm unidades de tempo, e consequentemente um atraso de tempo de no máximo ε .

Com este mecanismo de coordenação, ganhamos uma importante propriedade, de limitar o jogo para ter exatamente um equilíbrio de Nash. Para tanto, basta ver que, em um equilíbrio de Nash, a i -ésima tarefa sempre é processada antes de qualquer tarefa $j > i$ que esteja na mesma máquina que i . Assim, a máquina que dá custo mínimo é unicamente determinada pelos pesos das tarefas anteriores a i já que todas as máquinas têm cargas distintas (por causa do atraso).

Lema 3.5.3. O mecanismo de coordenação acima define exatamente um equilíbrio de Nash no jogo de balanceamento de cargas com máquinas uniformes.

Note que a menos do atraso em cada tarefa, o mecanismo acima nos dá um equilíbrio de Nash que é exatamente o obtido pelo algoritmo LPT, visto anteriormente. Assim, vamos denominar este mecanismo por LPT*. Com isso, temos um preço da anarquia menor que $4/3$, como pode ser visto no próximo teorema. Uma melhora significativa em relação ao limitante de $2 - 2/(m+1)$ apresentado no Teorema 3.2.3. A prova é a mesma da análise do fator de aproximação do algoritmo LPT [Gra66], que incluímos por ser elegante. Usamos o seguinte lema, cuja prova deixamos para o leitor interessado.

Lema 3.5.4. Se o mecanismo LPT* para máquinas uniformes obtém uma atribuição onde todas as máquinas têm no máximo duas tarefas, então seu preço da anarquia nestas condições é 1.

Teorema 3.5.5. O mecanismo LPT*, para máquinas uniformes, tem preço da anarquia $4/3 - 1/3m$.

Demonstração. Como o acréscimo devido ao atraso das tarefas pode ser tão pequeno quanto se queira, vamos ignorar este valor, sabendo que podemos obter um preço da anarquia tão próximo do obtido sem atrasos, quanto se queira.

A prova será por contradição. Vamos supor que exista jogo com atribuição A , obtida pelo mecanismo LPT*, e uma atribuição \mathcal{O} de custo social ótimo para este jogo, onde é válido que

$$\frac{c(A)}{c(\mathcal{O})} > \frac{4}{3} - \frac{1}{3m}.$$

A escolha da máquina onde a tarefa n foi atribuída ocorreu por ser a máquina menos carregada naquele momento. Com isso, todas as máquinas estão com carga de pelo menos $c(A) - w_n$,

$$c(A) - w_n \leq \frac{\sum_{i=1}^{n-1} w_i}{m},$$

o que nos dá que

$$\begin{aligned} c(A) &\leq \frac{\sum_{i=1}^n w_i}{m} + \frac{(m-1)w_n}{m} \\ &\leq c(\mathcal{O}) + \frac{(m-1)w_n}{m}. \end{aligned}$$

Dividindo a desigualdade anterior e usando a hipótese, temos

$$\frac{4}{3} - \frac{1}{3m} < \frac{c(A)}{c(\mathcal{O})} < 1 + \frac{m-1}{m} \frac{w_n}{c(\mathcal{O})},$$

que comparando os extremos e simplificando, temos

$$c(\mathcal{O}) < 3w_n.$$

Como w_n é a menor tarefa, concluímos que todas as máquinas apresentam no máximo duas tarefas e pelo Lema 3.5.4 anterior, o preço da anarquia é 1, contrariando a suposição inicial.

Por fim, o preço da anarquia de $4/3 - 1/3m$ é justo, também advindo da análise do fator de aproximação do algoritmo LPT. O jogo definido por esta instância possui três jogadores com tarefas de peso m e para cada $k = m+1, \dots, 2m-1$, há dois jogadores com tarefas de peso k . O custo social ótimo é $3m$, mas a atribuição obtida pelo mecanismo LPT* tem custo $4m-1$. A Figura 3.2 apresenta uma tal instância para $m=3$. \square

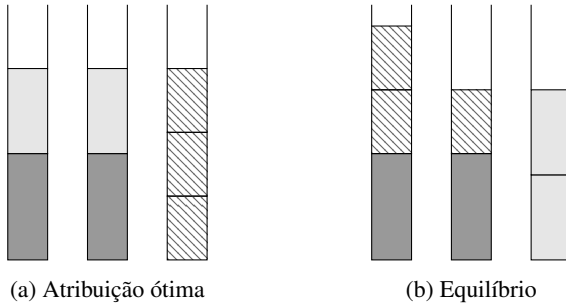


Figura 3.2: Uma instância do jogo de balanceamento de carga com três máquinas e sete tarefas. Duas tarefas (mais escuras) têm peso 5, duas tarefas (mais claras) têm peso 4 e três tarefas (hachuradas) têm peso 3. Na Figura (a) temos uma atribuição ótima (com *makespan* 9) e na Figura (b) temos uma atribuição em equilíbrio de Nash encontrada pelo mecanismo LPT* (com *makespan* 11).

Quando consideramos máquinas relacionadas, é possível aplicar o mesmo mecanismo LPT*, dando prioridade para as tarefas maiores e atribuindo-as, nesta ordem, na máquina que a finalizar no menor tempo.

O seguinte teorema, provado por Kovács [Kov10], apresenta limitantes inferiores e superiores para o preço da anarquia do mecanismo LPT*.

Teorema 3.5.6. O mecanismo de coordenação LPT* para o caso de máquinas relacionadas, possui preço da anarquia entre $1,54$ e $1 + \sqrt{3}/3 \approx 1,5773$.

Note que essa é uma melhora expressiva em relação ao preço da anarquia de $\Theta(\lg m / \lg \lg m)$ visto anteriormente para máquinas relacionadas.

Considere agora que os jogadores possam, além de escolher a máquina onde irá alocar sua tarefa, também mentir qual o real peso da sua tarefa, com o objetivo de enganar o mecanismo de coordenação [Kou03]. Note que um jogador i não terá vantagens em mentir o peso de sua tarefa para um valor menor que w_i , uma vez que a máquina terminará antes de completá-la. Por outro lado, o jogador pode ser beneficiado ao declarar um peso maior. Em particular, um jogador pode se beneficiar ao mentir o peso de sua tarefa no mecanismo LPT*, como podemos ver no exemplo seguinte.

Exemplo 3.5.7. Considere a seguinte instância com duas máquinas idênticas e três tarefas. As tarefas têm pesos $w_1 = 1$, $w_2 = 2$ e $w_3 = 3$. Caso todos os jogadores digam a verdade, o mecanismo LPT* atribuiria a tarefa 3 em uma máquina e as tarefas 1 e 2 em outra máquina, dando um custo social ótimo. Nesta atribuição as tarefas 3 e 2 estariam nas melhores posições, pois ambas seriam as primeiras a serem atendidas por suas respectivas máquinas. Porém, conhecendo o mecanismo, o jogador 1 poderia mentir, anunciando um peso um pouco maior que 2. Desta maneira, o mecanismo colocaria a tarefa do jogador 1 na frente da tarefa do jogador 2, obtendo vantagens.

Agora, considere o mecanismo baseado no algoritmo de escalonamento SPT (*Shortest Processing Time*), cuja diferença com o algoritmo LPT está na ordenação das tarefas, onde tarefas menores são atribuídas primeiro. Da mesma forma, vamos considerar o mecanismo SPT*, igual ao mecanismo LPT*, com a diferença que tarefas menores são atribuídas primeiro e têm mais prioridade. Consideramos as mesmas formas para desempatar tarefas de mesmo tamanho, pela ordenação lexicográfica de seus identificadores, e com o acréscimo de atrasos para que a cada momento exista apenas uma máquina de menor carga.

Teorema 3.5.8. O mecanismo SPT*, para máquinas uniformes, tem preço da anarquia $2 - 1/m$. Ademais, nenhum jogador se beneficia ao mentir (individualmente) o peso de sua tarefa.

Embora tenha um preço da anarquia pior que o do mecanismo LPT*, a garantia de que os jogadores não obtêm benefício ao mentir torna o SPT um mecanismo importante nesta abordagem.

No próximo capítulo, consideramos em mais detalhes situações onde desejamos os que os jogadores reportem suas reais informações para um determinado sistema.

Leilões e Mecanismos

Leilões são usados desde a antiguidade para diversos tipos de vendas. Eles são utilizados para transferir bens públicos para empresas privadas, conceder direitos de utilização de recursos naturais de um governo e na venda das mais variadas *commodities* e objetos de arte. Porém, em tais leilões os compradores têm interesses próprios e independentes e, por isso, podem agir de maneira estratégica buscando obter um resultado mais favorável para si.

Este tipo de situação ocorre não apenas em leilões, mas em vários problemas onde há vários participantes, como eleições e escolhas de líderes, definição de prioridades, alocação e pareamento por preferências, etc. Em tais situações, em geral é interessante estipular um conjunto de regras que levem os jogadores a terem um determinado comportamento mesmo agindo de forma egoísta.

Nesse capítulo, consideramos leilões, o impacto do pensamento estratégico dos compradores em algumas situações econômicas e o projeto de mecanismos que levem os jogadores a terem um “bom” comportamento.

4.1 Leilões de um único item

Para introduzir alguns conceitos fundamentais da Teoria de Leilões, começamos abordando a venda de um único item.

Considere um leilão de um único item onde os potenciais compradores fazem lances com o objetivo de comprar o item. Consideramos que cada

comprador b tem um *valor* $v_b \in \mathbb{R}_+$ (conhecido apenas por b) que representa o preço máximo que b está disposto a pagar pelo item. Ademais, consideramos que a utilidade de um comprador b é $v_b - p$ se o comprador b ganha o item e paga o preço p e 0 caso o comprador b não ganhe o item¹. A escolha do preço p depende do leilão considerado, como apresentamos a seguir. Do ponto de vista da Teoria dos Jogos, é de se esperar que os compradores irão se comportar de forma estratégica nos leilões em que participam em busca de maximizar sua utilidade.

Consideramos leilões onde o leiloeiro gostaria de entregar o item para o comprador com maior v_b , independentemente do preço cobrado pelo item. Isto é, a motivação de tal leiloeiro não é maximizar o seu lucro, mas sim maximizar o *bem-estar social*. Maximizar o bem-estar social pode ser interessante, por exemplo, em casos como a distribuição de itens do governo para a sociedade.

Existem várias formas de leiloar um único item. Por exemplo, podemos considerar os *leilões de carta fechada*, onde os compradores fazem um único lance pelo item simultaneamente e de forma que apenas o leiloeiro conhece o valor de tal lance. Para evitar empates, assumimos que o comprador de maior lance com o menor índice (para alguma ordenação fixa dos compradores) é o ganhador.

No *Leilão de Primeiro Preço*, após cada comprador enviar o seu lance ao leiloeiro, o leiloeiro entregará o item ao comprador que fizer o maior lance e cobrará esse valor desse comprador. Do ponto de vista estratégico, cada comprador escolhe um lance $\ell_b \in \mathbb{R}_+$ e obtém utilidade 0 se não recebe o item e $v_b - \ell_b$ se recebe o item.

Um dos problemas deste tipo de leilão, é que os compradores tendem a mentir sobre o valor a ser declarado no envelope. Note que, no leilão de primeiro preço, se o comprador b recebe o item e seu lance é $\ell_b = v_b$ então sua utilidade é 0, a mesma utilidade que ele obteria se não recebesse o item. Mas se o comprador fizer um lance estritamente menor que v_b e ele for o ganhador, seu lucro será positivo. Dessa forma, há um estímulo para que o lance não reflita o real valor do comprador, e assim é difícil garantir que o ganhador é o que dá maior valor ao item. Ou seja, o leiloeiro pode não estar maximizando o bem estar social como gostaria.

Já no *Leilão de Segundo Preço* (também chamado de *Leilão de Vickrey* [Vic61]), ao invés de cobrarmos do ganhador o seu próprio lance

¹Dizemos que tal utilidade é *quasi-linear* já que $v_b - p$ é linear em p para v_b constante.

(como no Leilão de Primeiro Preço), o preço cobrado do ganhador é o segundo maior lance dado pelos compradores. Como mostramos a seguir, esse leilão tem a interessante propriedade de ser *compatível com incentivo* (ou *à prova de estratégia*), isto é, um comprador nunca obtém uma utilidade menor ao escolher como lance o seu real valor para o item.

Proposição 4.1.1. No Leilão de Vickrey, declarar o valor $\ell_b = v_b$ é uma estratégia dominante para todo comprador b .

Demonstração. Considere um comprador b e um vetor ℓ_{-b} representando o lance dos outros jogadores. Seja p o maior valor dos outros compradores, isto é, $p = \max\{\ell_{b'} : b' \neq b\}$. Vamos mostrar que v_b é uma resposta ótima para ℓ_{-b} .

Lembre-se que se b não recebe o item reportando ℓ_b , então sua utilidade é 0 e se ele ganha o item sua utilidade é $v_b - p$.

Se $v_b \leq p$, então para qualquer lance ℓ_b , a utilidade de b é não-positiva e se $v_b > p$, então a utilidade de b é no máximo $v_b - p$. Portanto, v_b é uma resposta ótima para ℓ_{-b} .

Assim, como v_b é uma resposta ótima para qualquer vetor ℓ_{-b} , temos que v_b é uma estratégia dominante. \square

Além dos leilões de carta fechada, existem outros tipos de leilões para a venda de um único item. A forma mais tradicional de vender um único item é através do *Leilão Inglês*, um leilão aberto onde o leiloeiro inicia o leilão com um preço mínimo, que é aumentado em pequenos incrementos até que haja apenas um comprador interessado em adquirir o item por aquele preço [Kri09].

É interessante notar a semelhança entre o Leilão de Segundo Preço e o Leilão Inglês. Suponha que o leiloeiro inicie um Leilão Inglês com um preço mínimo igual a zero, que os incrementos sejam de $\varepsilon > 0$. Sejam v e w , respectivamente, o maior e o segundo maior valor dos compradores e suponha que $v \geq w + \varepsilon$. Enquanto o lance atual é menor ou igual a w , pelo menos dois compradores ainda estão interessados no item, porém quando o lance atual for maior do que w , apenas um comprador estará interessado no item. Assim, ao final de tal leilão, o ganhador recebe o item pagando um preço entre w e $w + \varepsilon$, um resultado muito próximo do resultado obtido com o Leilão de Segundo Preço.

Note que no leilão de Vickrey, pudemos fazer os compradores declararem seus valores privados por termos definido o pagamento do vencedor

baseado nos valores declarados pelos outros compradores. De fato esta ideia é fundamental no mecanismo VCG, apresentado a seguir.

4.2 Mecanismos

Considere que desejemos escolher uma *alternativa* a de um conjunto de alternativas A e que temos n jogadores onde cada jogador j tem um valor privado $v_j(a) \in \mathbb{R}$ para cada $a \in A$. Dizemos que $v_j: A \rightarrow \mathbb{R}$ é uma *valoração* e consideramos que cada jogador j tem um conjunto V_j de possíveis valorações. Assim como no Capítulo 1, denotamos $V_1 \times V_2 \times \dots \times V_n$ por V e $V_1 \times \dots \times V_{j-1} \times V_{j+1} \times \dots \times V_n$ por V_{-j} .

Idealmente, cada jogador reporta ao sistema qual é a sua valoração. Porém, como as valorações dos jogadores são informações privadas e os jogadores agem de forma estratégica, consideramos que o jogador reporta um elemento ℓ_j de V_j não necessariamente igual a v_j . Assim, estipulamos um preço p_j a ser pago por cada jogador j . Se a alternativa a é escolhida e o jogador j precisa pagar p_j então sua utilidade é $v_j(a) - p_j$.

Definição. Um mecanismo (de revelação direta) é dado por uma função de escolha social $f: V_1 \times \dots \times V_n \rightarrow A$ e funções de pagamento p_1, \dots, p_n onde $p_j: V_1 \times \dots \times V_n \rightarrow \mathbb{R}$ é o valor que o jogador j paga.

Baseado nas estratégias dos jogadores, um mecanismo deve escolher uma alternativa de um conjunto A (chamada de *escolha social*) e deve definir quanto cada comprador deve pagar por esta escolha. Estamos interessados em mecanismos para os quais cada comprador tem como estratégia dominante declarar a sua informação privada para cada resultado (mentir não o leva a ter mais vantagens).

Definição. Um mecanismo (f, p_1, \dots, p_n) é dito ser *compatível com incentivo* se para todo jogador j , para todo $v_j \in V_j$ e para todo $\ell \in V$ temos

$$v_j(a) - p_j(v_j, \ell_{-j}) \geq v_j(a') - p_j(\ell_j, \ell_{-j}),$$

onde $a = f(v_j, \ell_{-j})$ e $a' = f(\ell_j, \ell_{-j})$.

No que se segue, desejamos escolher uma alternativa $a \in A$ à partir das valorações dos jogadores que maximize $\sum_{j=1}^n v_j(a)$ (o *bem-estar social*). Chamamos as alternativas que maximizam $\sum_{j=1}^n \ell_j(a)$ de *socialmente eficiente*. No caso de mecanismos compatíveis com incentivo, temos

que alternativas socialmente eficientes maximizam o bem-estar social se considerarmos que os jogadores reportam a verdade (já que não há incentivo para mentir). Assim, podemos maximizar $\sum_{j=1}^n \ell_j(a)$ para maximizar $\sum_{j=1}^n v_j(a)$. Porém, se o mecanismo não for compatível com incentivo então escolher uma alternativa que maximiza $\sum_{j=1}^n \ell_j(a)$ não necessariamente é o mesmo que maximizar $\sum_{j=1}^n v_j(a)$.

Uma das principais abordagens para projetar mecanismos é aplicar o método VCG [Vic61, Cla71, Gro73], que permite definir mecanismos compatíveis com incentivo que encontram alternativas socialmente eficientes.

Definição. Um mecanismo (f, p_1, \dots, p_n) é dito ser um *mecanismo VCG* (Vickrey-Clarke-Groves) se, para todo $\ell \in V$,

$$f(\ell_1, \dots, \ell_n) \in \arg \max \left\{ \sum_j \ell_j(a) : a \in A \right\}. \quad (4.1)$$

Ademais, existem funções h_1, \dots, h_n , onde $h_j: V_{-j} \rightarrow \mathbb{R}$ tais que, para todo $\ell \in V$,

$$p_j(\ell_1, \dots, \ell_n) = h_j(\ell_{-j}) - \sum_{k \neq j} \ell_k(f(\ell_1, \dots, \ell_n)). \quad (4.2)$$

Na restrição (4.1) temos que o mecanismo deve escolher uma alternativa socialmente eficiente a dadas as valorações submetidas pelos jogadores. Já na restrição (4.2) temos que a função de pagamento de um jogador j depende de uma função $h_j(\ell_{-j})$ e da soma $\sum_{k \neq j} \ell_k(a)$ que é a soma total dos valores dos outros jogadores para a alternativa a . Note que o valor pago ao jogador j não depende de ℓ_j . Na visão do jogador j , o termo h_j é basicamente uma constante, pois só depende dos valores declarados pelos outros jogadores.

Teorema 4.2.1. Todo mecanismo VCG é compatível com incentivo.

Demonstração. Considere um jogador j . Vamos mostrar que declarar v_j é uma estratégia dominante para o jogador j . Seja ℓ_{-j} as valorações declaradas pelos outros jogadores. Assim, considere $\ell_j \in V_j$ e sejam $a = f(v_j, \ell_{-j})$ e $a' = f(\ell)$. A utilidade de j , ao declarar v_j é

$$v_j(a) - p_j(v_j, \ell_{-j}) = v_j(a) - h_j(\ell_{-j}) + \sum_{k \neq j} \ell_k(a).$$

Ao declarar ℓ_j , a utilidade de j é

$$v_j(a') - p_j(\ell) = v_j(a') - h_j(\ell_{-j}) + \sum_{k \neq j} \ell_k(a').$$

Note que, como a é uma alocação socialmente eficiente para (v_j, ℓ_{-j}) , temos que $v_j(a) + \sum_{k \neq j} \ell_k(a) \geq v_j(a') + \sum_{k \neq j} \ell_k(a')$ e, assim, temos que a utilidade do jogador j não é melhor ao declarar ℓ_j . \square

Note que, apesar de dizermos que p_j é o preço pago pelo jogador j , a função p_j não é necessariamente não-negativa. Assim, dependendo da situação e das funções h_1, \dots, h_n , é possível que alguns (ou até mesmo todos) os jogadores tenham preços negativos. Assim, tais jogadores recebem dinheiro ao invés de pagar, o que pode não ser interessante em certos casos.

Definição. Dizemos que um mecanismo *não tem transferências positivas* se nenhum jogador recebe do mecanismo (em vez de pagar). Isto é, se para todo $\ell \in V$ e todo jogador j , temos $p_j(\ell) \geq 0$.

Em geral, também é interessante que em um mecanismo, um jogador não pague por uma alternativa mais do que ela vale para o mesmo se ele reportar sua real valoração. Assim, as utilidades do jogadores são sempre não-negativas e portanto nenhum jogador tem prejuízo ao participar do mecanismo.

Definição. Um mecanismo é dito ser *individualmente racional* se os jogadores sempre obtêm utilidade não-negativa ao reportar sua real valoração. Isto é, para todo $\ell \in V$, se $a = f(\ell)$ então $\ell_j(a) - p_j(\ell) \geq 0$.

Note que o jogador pode vir a obter utilidade negativa em um mecanismo individualmente racional ao mentir, informando $\ell_j(a) > v_j(a)$.

É possível obter mecanismos VCG que atendam a estas duas condições utilizando a regra de pivotação de Clarke [Cla71]: para cada jogador j , definimos $h_j(\ell_{-j}) = \max\{\sum_{k \neq j} \ell_k(b) : b \in A\}$. Assim, o preço pago por um jogador j é

$$p_j(\ell) = \max \left\{ \sum_{k \neq j} \ell_k(b) : b \in A \right\} - \sum_{k \neq j} \ell_k(a),$$

isto é, o prejuízo (ou *externalidade*) que ele causa aos outros jogadores por participar do mecanismo.

Lema 4.2.2. Um mecanismo VCG com regra de pivotação de Clarke não faz transferências positivas. Além disso, se $v_j(a) \geq 0$ para todo $v_j \in V_j$ e $a \in A$, então o mecanismo é individualmente racional.

Demonstração. Seja $a = f(\ell_1, \dots, \ell_n)$ uma alternativa que maximiza $\sum_k \ell_k(a')$ para $a' \in A$ e $b \in A$ uma alternativa que maximiza $\sum_{k \neq j} \ell_k(b')$ para $b' \in A$. Note que

$$p_j(\ell) = \sum_{k \neq j} \ell_k(b) - \sum_{k \neq j} \ell_k(a) \geq 0,$$

já que b maximiza $\{\sum_{k \neq j} \ell_k(b') : b' \in A\}$ e, portanto, não há transferências positivas.

Para mostrar que o mecanismo é individualmente racional considere que $\ell_j = v_j$. A utilidade do comprador j é dada por

$$\begin{aligned} v_j(a) - p_j(\ell) &= v_j(a) + \sum_{k \neq j} \ell_k(a) - \sum_{k \neq j} \ell_k(b) \\ &\geq \sum_k \ell_k(a) - \sum_k \ell_k(b) \end{aligned} \quad (4.3)$$

$$\geq 0, \quad (4.4)$$

onde a desigualdade (4.3) é válida pois $\ell_j(b) \geq 0$ e $\ell_j = v_j$ e a desigualdade (4.4) é válida pois a foi escolhida como uma alternativa que maximiza $\sum_k \ell_k(a')$ para $a' \in A$. \square

4.2.1 Leilões de segundo preço e mecanismo VCG

Considere novamente o Leilão de Segundo Preço. A seguir, definimos um mecanismo equivalente a tal leilão e mostramos que o mesmo é um mecanismo VCG com a regra de pivotação de Clarke. Denotamos o valor privado de um comprador b por w_b ao invés de v_b como fizemos anteriormente.

Uma alternativa a indica quem é o ganhador. Assim, se o conjunto de compradores é $B = \{1, \dots, n\}$ então o conjunto de alternativas A é igual a B . O valor da alternativa a para o comprador b é dado por $v_b(a)$ que é 0 se $a \neq b$ ou w_b se $a = b$. Assim, V_b é o conjunto de todas as valorações ℓ_b que satisfazem $\ell_b(a) = 0$ se $a \neq b$ e $\ell_b(a) \geq 0$ caso contrário. Note que o lance de um comprador b é o valor $\ell_b(b)$.

Aplicando o mecanismo VCG para tal cenário, escolhemos uma alternativa a que maximiza $\sum_{b \in B} \ell_b(a)$. Como existe apenas um termo não nulo em $\sum_{b \in B} \ell_b(a)$, equivalentemente, escolhemos o comprador que dá o maior lance.

Seja a o vencedor, isto é, $a \in \arg \max \{ \sum_{b \in B} \ell_b(a') : a' \in A \}$. Utilizando a regra de pivotação de Clarke temos que, para um comprador b ,

$$h_b(\ell_{-b}) = \max \left\{ \sum_{c \neq b} \ell_c(a') : a' \in A \right\}.$$

Note que, se $b = a$, então $h_b(\ell_{-b})$ é o segundo maior lance e se $b \neq a$ então $h_b(\ell_{-b})$ é o maior lance. Assim, um comprador b paga o preço

$$p_b = h_b(\ell_{-b}) - \sum_{c \neq b} \ell_c(a).$$

Se $b = a$, então $\sum_{c \neq b} \ell_c(a) = 0$ e, portanto, p_b é o segundo maior lance. Se $b \neq a$, então $\sum_{c \neq b} \ell_c(a)$ é igual ao maior lance e, portanto, p_b é 0.

4.2.2 Jogo de caminho mínimo

Muitos projetos de mecanismos podem ser resolvidos usando o mecanismo VCG. Um exemplo é o jogo do caminho mínimo [NR99], que definimos a seguir.

Considere um grafo $G = (V, E)$ onde cada jogador é capaz de construir uma única aresta $e \in E$ e, caso o faça, terá um custo de valor $c_e \geq 0$. Por simplicidade, usamos e para denotar tanto a aresta como o jogador. Ademais, consideramos que G é 2-aresta conexo (isto é, a remoção de uma aresta não desconecta o grafo).

Suponha que desejemos construir um caminho de um vértice s a um vértice t em G através da construção de algumas das arestas de G . Assim, temos um leilão reverso: desejamos comprar arestas dos jogadores para construir o caminho e cada jogador informa o seu custo de construção. Desejamos escolher um caminho de custo social mínimo, isto é, um caminho P que minimize $\sum_{e \in P} c_e$. Ademais, gostaríamos de compensar os jogadores financeiramente pela construção da aresta através de um pagamento p_e (isto é, um preço não positivo).

Este problema tem aplicações na Internet, onde várias sub-redes pertencem a diferentes agentes e busca-se uma rota para fazer uma transmissão de dados pela rede. A ideia é construir um mecanismo que permita construir um caminho barato de s a t e defina pagamentos para os compradores de maneira que eles não tenham vantagens ao declarar valores diferentes dos custos verdadeiros das construções de suas arestas.

Como os jogadores podem agir de maneira estratégica, utilizamos um mecanismo VCG para garantir que os jogadores reportem seus custos reais para as construções das arestas.

O conjunto de alternativas A é composto por todos os caminhos de s a t em G . Assim, chamamos uma alternativa apenas por caminho. Denotamos o custo $\sum_{e \in P} c_e$ de um caminho P simplesmente por $c(P)$. O valor $v_e(P)$ de um jogador e para um caminho P é 0 se P não contém a aresta e e $-c_e$ se P contém a aresta e . Assim, um caminho P é socialmente eficiente se $P \in \arg \min \{c(P') : P' \in A\}$, ou seja, coincide com o nosso objetivo de construir um caminho de custo social mínimo.

Basta então definir as funções h_1, \dots, h_n usadas na definição dos preços. Porém, como estamos lidando com custos e os preços serão negativos, consideramos *pagamentos* onde o pagamento que um jogador recebe é o oposto do preço que ele paga. Seja \mathcal{P}_{G-e} o conjunto dos caminhos de s a t que não usam a aresta e . Para um jogador e , escolhemos $h_e = \min \{c(P') : P' \in \mathcal{P}_{G-e}\}$. Assim, quando o caminho de custo mínimo escolhido é P^* , o pagamento do agente e é dado por

$$p_e = \min \{c(P') : P' \in \mathcal{P}_{G-e}\} - \sum_{f \in P^*-e} c_f.$$

Note que, se $e \notin P^*$ então $p_e = 0$ e se $e \in P^*$ então

$$p_e = \min \{c(P') : P' \in \mathcal{P}_{G-e}\} - c(P^*) + c_e,$$

como $c(P^*) \leq \min \{c(P') : P' \in \mathcal{P}_{G-e}\}$, temos que $p_e \geq c_e$. Portanto, todo jogador que tem sua aresta construída recebe um valor de pelo menos c_e e todo jogador que não tem sua aresta construída recebe 0, assim o nosso mecanismo é individualmente racional. Ademais, como $c_e \geq 0$, nosso mecanismo não tem transferências negativas, isto é, $p_e \geq 0$ para todo jogador e .

A definição do resultado e dos pagamentos pode ser feita através de no máximo n execuções de um algoritmo de caminho mínimo, sendo uma

execução para a obtenção do caminho P^* e no máximo $n - 1$ execuções para os pagamentos, uma para cada aresta $e \in P^*$. Para obter um caminho mínimo P' de s a t que não passa por e , basta executar um algoritmo de caminho mínimo no grafo $G - e$. Como o grafo é 2-aresta conexo, tal caminho sempre existe.

Assim, o mecanismo VCG para este jogo pode ser realizado com complexidade de tempo polinomial, uma vez que a obtenção de um caminho mínimo pode ser feita em tempo polinomial [Dij59]. É interessante observar que em vez de fazer $n - 1$ execuções do algoritmo de caminho mínimo para computar os pagamentos dos agentes no caminho ótimo, é possível obter todos os pagamentos computando apenas duas árvores de caminhos mínimos [HS01].

Exemplo 4.2.3. Considere o grafo da Figura 4.1.

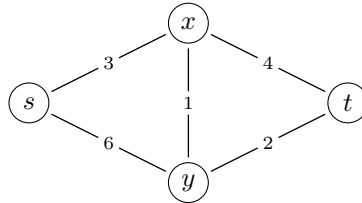


Figura 4.1: Construção de caminho mínimo.

O caminho mínimo de s a t é dado pela sequência de vértices $P^* = (s, x, y, t)$ e tem custo 6. Para calcular o pagamento da aresta sx , obtemos um caminho de s a t sem a aresta sx , o que nos dá o caminho $P_{-sx} = (s, y, t)$ de custo 8. Assim, o pagamento p_{sx} da aresta sx é dado por

$$p_{sx} = c(P_{-sx}) - (c(P^*) - c_{sx}) = 8 - (6 - 3) = 5.$$

Note que se o custo da aresta sx fosse maior que 5, o custo do caminho P^* seria maior que 8 e com isso seria melhor tomar o caminho (s, y, t) que tem custo 8. Analogamente, o pagamento das arestas xy e yt são dados por

$$\begin{aligned} p_{xy} &= c(P_{-xy}) - (c(P^*) - c_{xy}) = 7 - (6 - 1) = 2, \\ p_{yt} &= c(P_{-yt}) - (c(P^*) - c_{yt}) = 7 - (6 - 2) = 3. \end{aligned}$$

Quando o problema de otimização subjacente para encontrar o resultado ótimo e os pagamentos é resolvido de maneira eficiente, podemos ter uma implementação do mecanismo VCG de maneira eficiente. Em particular, podemos ter mecanismos eficientes para o correspondente jogo que busca (constrói) uma árvore geradora de peso mínimo [CLRS01] e o jogo que busca um emparelhamento de peso máximo [LP86].

4.3 Leilões combinatórios

Nesta seção, consideramos leilões onde vários itens estão sendo leiloados e cada comprador pode dar valores diferentes para cada combinação de itens. Tais leilões são chamados de leilões combinatórios e foram introduzidos por Rassenti [Ras81] numa primeira tentativa de criar um mercado apoiado por computadores [RSB82].

Em tais leilões temos um conjunto finito de itens $I = \{1, \dots, n\}$ e um conjunto finito de compradores $B = \{1, \dots, m\}$. No caso do leilão de um único item, temos apenas um ganhador. Porém, no caso da venda de vários itens, podemos distribuir os itens entre os compradores. Uma *alocação* de itens a compradores é uma família A de conjuntos de itens indexada por B tal que para dois compradores diferentes b_1 e b_2 , temos que $A_{b_1} \cap A_{b_2} = \emptyset$. Isto é, cada item é alocado para no máximo um comprador e cada comprador recebe um conjunto (possivelmente vazio) de itens.

Consideramos que, para cada comprador b , existe uma função $v_b: 2^I \rightarrow \mathbb{R}_+$ onde, para $S \subseteq I$, $v_b(S)$ indica o valor do conjunto S para o comprador b . Chamamos v_b de *valoração do comprador b* e denotamos por V o conjunto de valorações válidas para cada comprador. Chamamos o vetor (v_1, \dots, v_m) simplesmente de *valoração*.

Usualmente, consideramos que, para todo comprador b , $v_b(\emptyset) = 0$ e que v_b é uma função monótona, isto é, $v_b(T) \geq v_b(S)$ para todo S e todo T tal que $S \subseteq T \subseteq I$. A restrição de v_b ser monótona é chamada de *livre disposição* pois assume que, se necessário, o comprador b pode se dispor dos itens do conjunto $T \setminus S$ sem custos.

Note que, como temos n itens, para representar a valoração de cada comprador precisamos de $2^n - 1$ valores. Assim, a própria quantidade de informação para representar uma instância de um leilão combinatório pode ser proibitiva. Porém, existem situações onde podemos representar as valorações dos compradores utilizando uma quantidade polinomial em

$|I \times B|$ de números. Por exemplo, o comprador b pode desejar obter um único conjunto S dando valor v para o mesmo. Assim, para cada conjunto T , se $S \subseteq T$, então $v_b(T) = v$ e, caso contrário, $v_b(T) = 0$. Neste caso, basta armazenar S e v para tal comprador.

Assim como no caso do leilão de um único item, consideramos que a valoração do comprador é de conhecimento exclusivo do comprador e que cada comprador submete um lance $\ell_b \in V$ que não é necessariamente igual a sua real valoração.

Estamos interessados em leilões combinatórios onde encontramos uma alocação A de itens a compradores que seja socialmente eficiente, isto é, que maximize $\sum_{b \in B} v_b(A_b)$. Tal problema, também é chamado de Problema da Determinação dos Vencedores (*Winner Determination Problem*).

Quando desejamos vender vários itens para vários compradores, podemos realizar vários leilões de único item. Porém, isso não retrata situações importantes como, por exemplo, quando um comprador está interessado em comprar um determinado conjunto de itens mas sim uma parte do mesmo. Neste caso, dizemos que os itens deste conjunto se *complementam*. Outra situação é quando um certo conjunto de itens pode ser *substituído* por outro. Isto é, um comprador poderia estar interessado em pelo menos um dos conjuntos, mas se tiver os dois seu valor não será maior que ter apenas um deles. Assim, vender vários itens através de leilões individuais pode levar a um bem-estar social menor do que se considerássemos todos os itens em um único leilão.

Exemplo 4.3.1. Considere um leilão onde dois itens $\{i_1, i_2\}$ estão sendo leiloados e há três potenciais compradores b_1, b_2 e b_3 . A Tabela 4.1 dá os pesos atribuídos por cada comprador a cada possível conjunto.

	$\{i_1\}$	$\{i_2\}$	$\{i_1, i_2\}$
b_1	3	6	6
b_2	5	4	15
b_3	2	8	10

Tabela 4.1: Exemplo de instância para leilão combinatório.

Nesse caso, o comprador b_1 prefere o item i_2 ao item i_1 , porém ele dá o mesmo valor para $\{i_1, i_2\}$ e $\{i_2\}$. Assim, i_1 e i_2 são substitutos para b_1 . Já o comprador b_2 prefere o item i_1 ao item i_2 , porém os dois itens se

complementam já que $v_{b_2}(\{i_1, i_2\}) > v_{b_2}(\{i_1\}) + v_{b_2}(\{i_2\})$. Finalmente, para o comprador b_3 , os itens são independentes e o valor dos dois itens juntos é exatamente a soma dos dois itens em separado.

Note que, se fizermos dois leilões separados, um para i_1 e outro para i_2 , então venderíamos o item i_1 para o comprador b_2 e o item i_2 para o comprador b_3 , obtendo um bem-estar social de 13. Porém, através de um leilão combinatório podemos vender ambos os itens para o comprador b_2 e obter um bem-estar social de 15.

O Problema da Determinação dos Vencedores pode ser formulado como um programa inteiro [BM97]. Seja x um vetor binário de variáveis indexados por $B \times 2^I$ onde $x_{b,S}$ é igual a 1 se e somente se o comprador b recebe o conjunto S na alocação encontrada. O objetivo é determinar x que

$$\begin{aligned} \text{(WDP)} \quad \max \quad & \sum_{b \in B} \sum_{S \subseteq I} v_b(S) x_{b,S} \\ \text{sujeito a} \quad & \sum_{S \subseteq I} x_{b,S} \leq 1 \quad \forall b \in B, \end{aligned} \quad (4.5)$$

$$\sum_{b \in B} \sum_{S \subseteq I: i \in S} x_{b,S} \leq 1 \quad \forall i \in I, \quad (4.6)$$

$$x_{b,S} \in \{0, 1\} \quad \forall b \in B, \forall S \subseteq I, \quad (4.7)$$

onde a restrição (4.5) garante que apenas um conjunto $S \subseteq I$ seja atribuído ao comprador b e a restrição (4.6) garante que o item i esteja em no máximo um conjunto S alocado a um comprador b . Ademais, note que a função objetivo maximiza o valor da alocação encontrada, isto é, a soma dos valores dos conjuntos atribuídos.

Podemos obter uma relaxação linear do programa inteiro (WDP) acima se removermos a condição de integralidade (4.5) para x , considerando apenas que $x_{b,S} \geq 0$ para todo comprador b e todo $S \subseteq I$.

Considere um vetor racional u de variáveis indexado por B e um vetor racional p de variáveis indexado por I . O programa dual da relaxação linear

de (WDP) consiste de encontrar vetores u e p que

$$\begin{aligned} \min \quad & \sum_{b \in B} u_b + \sum_{i \in I} p_i \\ \text{sujeito a} \quad & u_b + \sum_{i \in S} p_i \geq v_b(S) \quad \forall b \in B, \forall S \subseteq I, \\ & u_b \geq 0 \quad \forall b \in B, \\ & p_i \geq 0 \quad \forall i \in I. \end{aligned}$$

Os nomes das variáveis u e p foram escolhidos de forma proposital como explicamos a seguir. Porém, antes precisamos de introduzir algumas definições.

Uma *precificação* (de itens) é um vetor não-negativo p onde p_i indica o preço do item i . Dada uma precificação p e uma alocação A , se um comprador b recebe o conjunto S , então sua utilidade é $v_b(S) - \sum_{i \in A_b} p_i$. Ademais, dada uma precificação p para os itens, um conjunto T é uma *demanda* para o comprador b se, para todo conjunto $S \subseteq I$, temos que $v_b(S) - \sum_{i \in S} p_i \leq v_b(T) - \sum_{i \in T} p_i$. Assim, nenhum outro conjunto T dá utilidade maior para o comprador b do que o conjunto S . Por fim, dadas uma alocação A e uma precificação p , (A, p) é um *equilíbrio walrasiano* se, para cada comprador b , A_b é uma demanda para b considerando p e para qualquer item i não vendido, $p_i = 0$.

Estamos prontos para apresentar dois teoremas que relacionam a existência de um equilíbrio walrasiano com a existência de uma solução ótima inteira para a relaxação linear da formulação (WDP) [BM97]. No que segue, consideramos uma alocação A como solução da formulação (WDP). Para tanto, basta considerar a solução x correspondente a A onde, para todo comprador b , $x_{b, A_b} = 1$ e $x_{b, S} = 0$ para todo $S \neq A_b$.

Teorema 4.3.2 (Primeiro Teorema do Bem-Estar Social). Seja (A, p) um equilíbrio walrasiano, então A maximiza a relaxação linear da formulação (WDP). Em particular, A é socialmente eficiente.

Demonstração. Considere uma solução fracionária $x_{b, S}^*$ da relaxação linear. Como cada comprador recebe uma demanda, temos que $v_b(A_b) - \sum_{i \in A_b} p_i \geq v_b(S_b) - \sum_{i \in S_b} p_i$ para todo conjunto $S_b \subseteq I$.

Ademais, temos que para todo b , $\sum_{S \subseteq I} x_{b,S}^* \leq 1$. Assim,

$$v_b(A_b) - \sum_{i \in A_b} p_j \geq \sum_{S \subseteq I} x_{b,S}^* \left(v_{b,S} - \sum_{i \in S} p_i \right). \quad (4.8)$$

Note então que

$$\sum_{b \in B} \sum_{i \in A_b} p_i = \sum_{i \in I} p_i,$$

já que itens vendidos têm preço zero e um item é vendido a no máximo um comprador. Note também que

$$\sum_{b \in B} \sum_{S \subseteq I} x_{b,S}^* \sum_{i \in S} p_i \leq \sum_{i \in I} p_i,$$

já que o coeficiente de cada p_i é no máximo 1. Somando (4.8) sobre todos compradores, temos que

$$\sum_{b \in B} \left(v_b(A_b) - \sum_{i \in A_b} p_j \right) \geq \sum_{b \in B} \left(\sum_{S \subseteq I} x_{b,S}^* \left(v_{b,S} - \sum_{i \in S} p_i \right) \right).$$

Assim, como

$$\sum_{b \in B} \sum_{i \in A_b} p_i \geq \sum_{b \in B} \sum_{S \subseteq I} x_{b,S}^* \sum_{i \in S} p_i,$$

segue que

$$\sum_{b \in B} v_b(A_b) \geq \sum_{b \in B} \sum_{S \subseteq I} x_{b,S}^* v_{b,S}.$$

Em particular, como A é solução ótima inteira da relaxação de (WDP) é também solução ótima de (WDP) e, portanto, é socialmente eficiente. \square

Assim, se um equilíbrio walrasiano existir, então ele é solução ótima de (WDP). Apresentamos a seguir o resultado na outra direção, mostrando que toda solução inteira ótima forma um equilíbrio walrasiano.

Teorema 4.3.3 (Segundo Teorema do Bem-Estar Social). Se existe uma solução inteira ótima para (WDP) então existe um equilíbrio walrasiano para v .

Demonstração. Considere a alocação que A_1, \dots, A_m que pode ser obtida de tal solução ótima. Considere também uma solução dual ótima (u, p) que satisfaz folgas complementares em relação a solução inteira ótima considerada. Assim, como $x_{b, A_b} > 0$ então $u_b + \sum_{i \in A_b} p_i = v_b(A_b)$ e portanto, para todo conjunto T , temos que

$$v_b(A_b) - \sum_{i \in A_b} p_i \geq v_b(T) - \sum_{i \in T} p_i,$$

de onde concluímos que b recebe uma demanda. Além disso, se um item i não é vendido, então

$$\sum_{b \in B} \sum_{S \subseteq I: i \in S} x_{b, S} = 0 < 1$$

e, portanto, $p_i = 0$. Assim, (A, p) é um equilíbrio walrasiano. \square

Suponha que possamos consultar os compradores sobre suas preferências. Mais especificamente, considere que possamos fazer uma *consulta de demanda* onde perguntamos para um comprador b qual é uma de suas demandas para uma precificação p .

Se considerarmos que consultas de demanda são possíveis e que podem ser executadas em tempo polinomial em $|I \times B|$ (ou através de um oráculo que não consome tempo para ser consultado), então podemos resolver a relaxação linear de (WDP), mesmo tendo um número exponencial em $|I \times B|$ de variáveis. Ademais, se existir um equilíbrio walrasiano, então existe uma solução ótima da relaxação de (WDP) que pode ser encontrada em tempo polinomial em $|I \times B|$.

Porém, como (WDP) nem sempre tem uma solução ótima inteira, temos que um equilíbrio walrasiano nem sempre existe. De fato, considere o exemplo a seguir, retirado de Nisan et al. [NRTV07].

Exemplo 4.3.4. Considere dois itens i_1 e i_2 , dois compradores b_1 e b_2 e a valoração apresentada na Tabela 4.2.

Note que a alocação que maximiza o bem-estar social é $A_{b_1} = \emptyset$ e $A_{b_2} = \{i_1, i_2\}$. Porém, a solução fracionária x onde $x_{b_1, \{i_1\}}, x_{b_1, \{i_2\}},$ e $x_{b_2, \{i_1, i_2\}}$ são iguais a $1/2$ e $x_{b, S}$ é igual a 0 para os outros valores de b e S tem valor $7/2$ e é uma solução viável para a relaxação linear de (WDP). Isto é, a alocação A não é solução ótima da relaxação linear de (WDP) e, conseqüentemente, não é um equilíbrio walrasiano.

	$\{i_1\}$	$\{i_2\}$	$\{i_1, i_2\}$
b_1	2	2	2
b_2	0	0	3

Tabela 4.2: Exemplo de instância sem equilíbrio walrasiano.

Assim, nem sempre é possível utilizar a relaxação linear de (WDP) juntamente com consultas de demanda para resolver o Problema da Determinação dos Vencedores em tempo polinomial.

De fato, o Problema da Determinação dos Vencedores é NP-difícil [San02]. Um caso particular que pode ser resolvido em tempo polinomial, é quando cada comprador b está interessado em apenas um conjunto S_b de cardinalidade máxima 2 [RHP98].

Teorema 4.3.5. O Problema da Determinação dos Vencedores pode ser resolvido em tempo polinomial se, para cada comprador b , existe um conjunto S_b e um valor w_b tal que $|S_b| \leq 2$ e, para todo conjunto S , $v_b(S) = w_b$ se $S_b \subseteq S$ e $v_b(S) = 0$, caso contrário.

Demonstração. Considere um grafo G com um vértice para cada item e um vértice para cada comprador com $|S_b| = 1$. Para cada comprador b com $|S_b| = 1$, existe uma aresta em G com peso w_b entre o vértice que representa o único item em S_b e o vértice que representa o comprador b . Ademais, para cada comprador b com $|S_b| = 2$, existe uma aresta entre os vértices que representam os dois itens de S_b com valor w_b .

Note que existe uma bijeção entre alocações de itens a compradores e emparelhamentos de G (um emparelhamento é um conjunto de arestas que não têm extremos em comum). De fato, basta considerar o conjunto de vencedores de uma alocação com o conjunto de arestas correspondentes em G . Ademais, tal bijeção preserva o valor da alocação. Como o problema de se obter um emparelhamento de peso máximo pode ser resolvido em tempo polinomial [LP86], concluímos que o Problema da Determinação dos Vencedores pode ser resolvido em tempo polinomial neste caso. \square

Note que a implementação de um mecanismo VCG pede que uma solução ótima do Problema da Determinação dos Vencedores seja encontrada. Assim, em alguns casos é possível projetar um mecanismo VCG que

execute em tempo polinomial em $|I \times B|$ como no caso descrito acima ou como no caso do Teorema abaixo de Rothkopf et al. [RHP98].

Teorema 4.3.6. O mecanismo VCG pode ser implementado em tempo polinomial para leilões combinatórios onde $n \leq \log m$ ou $m \leq \log n$.

Porém, de forma geral, a menos que $P = NP$, não podemos ter um mecanismo VCG que possa ser computado em tempo polinomial em $|I \times B|$ para leilões combinatórios. Uma possibilidade abordada na literatura é considerar mecanismos compatíveis com incentivo que não utilizam uma solução ótima do Problema da Determinação dos Vencedores, mas sim uma solução aproximada. Isto é, apesar do mecanismo ser compatível com incentivo, ele não encontra uma alocação socialmente eficiente.

Outra opção é utilizar um algoritmo exato para resolver o Problema da Determinação dos Vencedores. Neste caso, há um algoritmo por programação dinâmica que executa em tempo $O(3^{|I|})$ [RHP98]. Porém, só é razoável utilizar esse algoritmo para instâncias pequenas.

Problemas de Precificação

Nesse capítulo abordamos problemas onde desejamos vender itens a vários compradores assim como no caso de leilões combinatórios. Porém, aqui estamos interessados em atribuir preços a cada um dos itens com o objetivo de maximizar o lucro do vendedor. Tais problemas são chamados de *problemas de precificação*. Em tais problemas é necessário decidir os preços dos itens conforme as preferências dos usuários já que se os itens forem caros demais, então poucas vendas serão realizadas. Por outro lado, se os itens forem muito baratos, então é possível que o lucro do vendedor seja baixo.

Decidir como precificar itens é um problema econômico muito interessante, bastante estudado e com diversos modelos e abordagens diferentes. Nesse capítulo, apresentamos modelagens não-paramétricas [RRG06] para esse problema. Consideramos modelos de *demanda unitária*, onde cada comprador deseja comprar no máximo um item. Assim, não é necessário considerar que os compradores dão valores para cada um dos subconjuntos de itens, como ocorre em leilões combinatórios. Basta considerar qual é o maior valor v_{ib} que um comprador b está disposto a pagar pelo item i . Ademais, nesses modelos não estamos preocupados com o comportamento estratégico dos compradores e, portanto, assumimos que v_{ib} é de conhecimento do vendedor.

No que segue, denotamos por $I = \{1, \dots, n\}$ o conjunto de itens e por $B = \{1, \dots, m\}$ o conjunto de compradores. A seguir, redefinimos os conceitos de valoração, precificação e alocação para o contexto de demanda unitária.

Uma *valoração* é uma matriz inteira não-negativa v indexada por $I \times B$ em que v_{ib} representa o maior valor que o comprador b está disposto a pagar pelo item i . Dizemos que v_{ib} é o *valor* do item i para o comprador b . Uma *precificação* é um vetor racional não-negativo p indexado por I , onde p_i indica o preço do item i . Uma *alocação* é um vetor x indexado por B onde x_b é o item alocado para o comprador b . Por vezes, para um comprador b e o item i , se $x_b = i$, então dizemos que i é *vendido* para b e que b *recebe* o item i . Dizemos que um item i é *viável* para o comprador b se $p_i \leq v_{ib}$. De forma geral, consideramos que cada item tem infinitas cópias e portanto pode ser vendido para tantos compradores quanto desejado, o que chamamos de *oferta ilimitada*. O *lucro* de um par (x, p) , onde x é uma alocação e p é uma precificação é definido como a soma do preço do item vendido, se existir, para cada comprador b .

5.1 O problema da compra máxima

Começamos apresentando o problema da compra máxima onde cada comprador b recebe um item viável.

Definição. No *problema da compra máxima*, dada uma valoração v , desejamos encontrar uma precificação p e uma alocação x que maximiza o lucro de forma que cada comprador ou recebe um item viável ou não recebe nenhum item.

Note que desejamos encontrar não somente a precificação p mas também uma alocação x . De fato, no caso do problema da compra máxima assim como em outros problemas descritos nesse capítulo, dada uma precificação p , podem existir várias alocações viáveis de itens a compradores. Assim, consideramos que o vendedor pode escolher a alocação x além da precificação.

Além disso, em uma solução ótima do problema da compra máxima (com oferta ilimitada), cada comprador b ou recebe um item viável i de preço máximo ou então não existe item viável para b . Porém, isso não é necessariamente verdade quando restringimos o número de vezes que um item pode ser vendido (oferta limitada).

Exemplo 5.1.1. A Figura 5.1 mostra graficamente uma valoração juntamente com uma alocação e uma precificação onde os compradores recebem apenas itens viáveis.

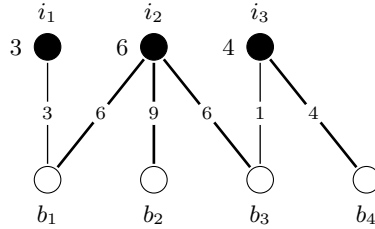


Figura 5.1: Uma instância para o problema da compra máxima.

O k -ésimo item é representado por um vértice de cor preta com rótulo i_k e o k -ésimo comprador é representado por um vértice de cor branca com rótulo b_k . Omitimos os pares $(i, b) \in I \times B$ tais que $v_{ib} = 0$. As arestas grossas indicam uma alocação e o valor ao lado esquerdo dos itens representam uma precificação. Como o nosso objetivo é maximizar o lucro, não há motivos para vender o item i_1 para o comprador b_1 , a não ser que não possamos vender o item i_2 mais do que duas vezes.

O nome do problema pode ser um pouco confuso porém foi mantido conforme introduzido por Aggarwal et al. [AFMZ04].

5.1.1 Oferta ilimitada

Nessa seção apresentamos um algoritmo de aproximação para o problema da compra máxima apresentado por Aggarwal et al. [AFMZ04].

Seja $P = \{v_{ib} : i \in I, b \in B\}$, isto é, P é o conjunto de valores distintos de v . Note que, sem perda de generalidade, podemos nos restringir a escolher os preços dos itens como elementos de P . De fato, para toda solução (x, p) tal que existe um item i com preço $p_i \notin P$, podemos encontrar uma solução com lucro não menor tal que todos os itens têm preços em P .

Começamos apresentando uma formulação de programação linear inteira para o problema. No que segue, temos um vetor binário w de variáveis indexado por $I \times P$ e um vetor binário ℓ de variáveis indexado por $B \times P$. Para todo item i e preço p , w_{ip} é igual a 1 se e somente o preço do item i é p e para todo item b e preço p , ℓ_{bp} é igual a 1 se e somente se obtemos lucro p a partir do comprador b . O objetivo é determinar w e ℓ que seja

solução ótima do seguinte programa linear inteiro.

$$(CM) \quad \max \sum_{b \in B} \sum_{p \in P} p \cdot \ell_{bp}$$

$$\text{sujeito a} \quad \sum_{p \in P} w_{ip} = 1 \quad \forall i \in I, \quad (5.1)$$

$$\sum_{p \in P} \ell_{bp} \leq 1 \quad \forall b \in B, \quad (5.2)$$

$$\ell_{bp} \leq \sum_{i \in I: p \leq v_{ib}} w_{ip} \quad \forall b \in B, \forall p \in P, \quad (5.3)$$

$$w_{ip} \in \{0, 1\} \quad \forall i \in I, \forall p \in P, \quad (5.4)$$

$$\ell_{bp} \in \{0, 1\} \quad \forall b \in B, \forall p \in P. \quad (5.5)$$

A restrição (5.1) garante que cada item i tem um preço p associado e a restrição (5.2) garante que obtemos lucro no máximo uma vez de cada comprador, a restrição (5.3) garante que, se obtemos lucro p de um comprador b , então existe um item i de preço p tal que $p \leq v_{ib}$, isto é, i é viável para b . As restrições (5.4) e (5.5) são as restrições de integralidade. Por fim, note que a função objetivo é exatamente o lucro do vendedor.

O algoritmo apresentado por Aggarwal et al. [AFMZ04] utiliza a relaxação linear da formulação acima para obter uma solução fracionária através de um arredondamento probabilístico. O preço p_i de um item i é escolhido aleatoriamente em P de acordo com a distribuição de probabilidade dada por w_i e os compradores recebem um dos itens viáveis mais caros, se existir um tal item.

ARREDONDAMENTO-CM(I, B, v)

- 1 seja x uma solução ótima da relaxação linear de (CM)
- 2 **para cada** item i em I
- 3 escolha p_i com probabilidade $\mathbb{P}(p_i = p) = w_{ip}$
- 4 **para cada** comprador b em B
- 5 **se** existe item i tal que $p_i \leq v_{ib}$
- 6 seja i um item com p_i máximo tal que $p_i \leq v_{ib}$
- 7 venda o item i para o comprador b
- 8 **senão**
- 9 o comprador b não receba nenhum item

Teorema 5.1.2. ARREDONDAMENTO-CM é uma $e/(e-1)$ -aproximação para o problema da compra máxima.

Demonstração. Considere um comprador b . Pelo algoritmo, temos que a probabilidade que um item i tenha preço p é w_{ip} e, portanto, a probabilidade de que um item i tenha preço pelo menos p e no máximo v_{ib} é $\sum_{p \leq p' \leq v_{ib}} w_{ip'}$. Seja $\omega_p = \sum_{i \in I} \sum_{p \leq p' \leq v_{ib}} w_{ip'}$. Note que ω_p é a soma dos lados esquerdos das desigualdades (5.3) com b fixo e $p \leq p'$. Como (w, ℓ) é uma solução é ótima, $\sum_{p \leq p'} \ell_{bp'} = \min(\omega_p, 1)$, pois, caso contrário, poderíamos aumentar o valor de ℓ_{bp} possivelmente diminuindo o valor de $\ell_{bp'}$ para $p' < p$ e obter uma solução melhor. Assim, usando o fato de que $1 - z \leq e^{-z}$ para todo z , a probabilidade que nenhum item seja viável para b e tenha preço pelo menos p é

$$\prod_{i \in I} \left(1 - \sum_{p \leq p' \leq v_{ib}} w_{ip'} \right) \leq e^{-\omega_p}.$$

Como, para todo $z \geq 0$, temos que $1 - e^{-z} \geq 1 - e^{-1}$ se $z \geq 1$ e $1 - e^{-z} \geq z(1 - e^{-1})$ se $0 \leq z \leq 1$, segue que a probabilidade q_p de b comprar um item de preço pelo menos p é no mínimo

$$1 - e^{-\omega_p} \geq (1 - e^{-1}) \min(\omega_p, 1) = (1 - e^{-1}) \sum_{p \leq p'} \ell_{bp'}.$$

Considere uma ordenação decrescente (p_1, p_2, \dots, p_k) dos elementos de P e, por simplicidade, considere que $p_{k+1} = 0$. Temos que

$$\sum_{i=1}^k (p_i - p_{i+1}) q_{p_i} \geq (1 - e^{-1}) \sum_{i=1}^k \left((p_i - p_{i+1}) \sum_{p_i \leq p'} \ell_{bp'} \right) \quad (5.6)$$

para todo $1 \leq i \leq k$.

Seja então r_p a probabilidade do comprador b comprar um item de preço exatamente p . Temos que $q_p = \sum_{p' \geq p} r_{p'}$. Note que o coeficiente de r_{p_j} no lado esquerdo da equação (5.6) e o coeficiente de ℓ_{bp_j} no lado direito da equação (5.6) são ambos iguais a r_{p_j} . Assim, o lucro esperado obtido do comprador b é pelo menos $(1 - e^{-1}) \sum_{p \in P} \ell_{bp} p$ e o resultado segue pela linearidade da esperança. \square

Apesar de se tratar de um algoritmo probabilístico, Aggarwal et al. [AFMZ04] mostraram que é possível desaleatorizar tal algoritmo, isto é, obter uma $e/(e - 1)$ -aproximação determinística para o problema da compra máxima. Eles mostraram também que não é possível aproximar o problema da compra máxima por um fator melhor do que $16/15$, a não ser que $P = NP$. De fato, tal resultado pode ser melhorado para $8/7$ considerando um resultado de Guruswami e Khot [GK05] que não havia sido publicado na época de publicação do artigo de Aggarwal et al.

5.1.2 Oferta limitada

Nessa seção abordamos uma generalização interessante do problema da compra máxima onde consideramos *oferta limitada*, isto é, que cada item tem uma quantidade finita de cópias e, portanto, não pode ser vendido para mais compradores do que o seu número de cópias.

Definição. No *problema da compra máxima limitada*, dados uma valoração v e um vetor C indexado por I , desejamos encontrar uma precificação p e uma alocação x que maximiza o lucro de forma que cada item i é alocado a no máximo C_i compradores e cada comprador ou recebe um item viável ou não recebe nenhum item.

Como um item não pode ser vendido para mais do que $|B|$ compradores, assumimos, sem perda de generalidade, que $C_i \leq |B|$ para todo item i . Note que, se $C_i = |B|$ para todo item i , o problema torna-se equivalente ao problema da compra máxima (com oferta ilimitada).

A seguir, apresentamos uma $e/(e - 1)$ -aproximação para o problema da compra máxima limitada apresentada por Fernandes e Schouery [FS14]. Começamos com uma definição simples, mas que é essencial no algoritmo.

Definição. Dado um item i e um conjunto S de compradores com $|S| \leq C_i$, chamamos o par (i, S) de uma *estrela* de i . Ademais, denotamos $\min\{v_{ib} : b \in S\}$ por $P_{(i,S)}$. Ou seja, $P_{(i,S)}$ é o menor valor de um comprador em S para o item i . Denotamos por $\mathcal{S}(i)$ o conjunto de estrelas do item i e denotamos por \mathcal{S} o conjunto de todas as estrelas, isto é, $\mathcal{S} = \bigcup_{i \in I} \mathcal{S}(i)$.

Note que, para uma estrela (i, S) tal que $S = \emptyset$, temos que $P_{(i,S)} = \min\{v_{ib} : b \in S\} = \infty$.

Considere uma solução (x, p) do problema da compra máxima limitada. Tal solução pode ser vista como uma coleção de estrelas onde existe uma estrela para cada item e cada comprador está em no máximo uma estrela. Ademais, se (x, p) é uma solução ótima e o item i é vendido para um conjunto S não-vazio de compradores, então $p_i \leq \min\{v_{ib} : b \in S\} = P_{(i,S)}$. De fato, temos que $p_i = P_{(i,S)}$ pois, caso contrário, poderíamos aumentar o preço de i e obter uma solução estritamente melhor. A partir desses conceitos é possível definir uma formulação por programação linear inteira para o problema da compra máxima limitada.

A seguir apresentamos a formulação (SF) (de *star formulation*), na qual temos um vetor binário x de variáveis indexado por \mathcal{S} . Para todo $(i, S) \in \mathcal{S}$, $x_{(i,S)}$ é igual a 1 se e somente se o conjunto de compradores que recebe o item i é precisamente S . O objetivo é determinar x que seja solução ótima do seguinte programa linear inteiro.

$$(SF) \quad \max \sum_{(i,S) \in \mathcal{S}} |S| P_{(i,S)} x_{(i,S)}$$

$$\text{sujeito a} \quad \sum_{(i,S) \in \mathcal{S}(i)} x_{(i,S)} = 1 \quad \forall i \in I, \quad (5.7)$$

$$\sum_{(i,S) \in \mathcal{S} : b \in S} x_{(i,S)} \leq 1 \quad \forall b \in B, \quad (5.8)$$

$$x_{(i,S)} \in \{0, 1\} \quad \forall (i, S) \in \mathcal{S}. \quad (5.9)$$

A restrição (5.7) garante que cada item i tenha exatamente uma estrela selecionada e a restrição (5.8) garante que cada comprador b esteja em no máximo uma estrela, isto é, b compra apenas um item. A restrição (5.9) garante que o vetor x de variáveis é binário.

Apesar dessa formulação poder ter $\Theta(|I|2^{|B|})$ variáveis, é possível resolver a relaxação linear em tempo polinomial como mostrado por Fernandes e Schouery [FS14].

Lema 5.1.3. A relaxação linear da formulação (SF) pode ser resolvida em tempo polinomial.

A seguir mostramos como utilizar a formulação por programação inteira apresentada para obter um algoritmo de aproximação para o problema da compra máxima limitada. A ideia é, a partir de uma solução ótima x da relaxação linear de (SF), escolher uma estrela S_i para cada item i de forma

que a probabilidade de uma estrela (i, S) ser escolhida seja $x_{(i,S)}$. Assim, fixamos que o preço do item i como $P_{(i,S_i)}$. Como as estrelas são escolhidas de forma aleatória, é possível que existam compradores que estejam em mais de uma estrela selecionada. Como nosso objetivo é maximizar o lucro, atribuímos um dos itens de maior preço para cada um desses compradores.

STARROUNDING(I, B, v, C)

- 1 seja x uma solução ótima da relaxação linear de (SF)
- 2 **para cada** item i em I
- 3 escolha $S_i \in \mathcal{S}(i)$ com probabilidade $\mathbb{P}(S_i = (i, S)) = x_{(i,S)}$
- 4 defina o preço de i como $P_{(i,S_i)}$
- 5 **para cada** comprador b em B
- 6 **se** b pertence a alguma das estrelas $\{S_1, \dots, S_{|I|}\}$
- 7 seja i um item com P_{S_i} máximo tal que $b \in S_i$
- 8 venda o item i para o comprador b

É possível mostrar que o algoritmo STARROUNDING é uma $e/(e-1)$ -aproximação para o problema da compra máxima. Como a prova é um pouco complicada optamos por mostrar que o STARROUNDING é uma 2-aproximação utilizando as mesmas ideias, o que fazemos a seguir. O leitor interessado pode ver a prova de que STARROUNDING é uma $e/(e-1)$ -aproximação em [FS14].

Teorema 5.1.4. STARROUNDING é uma 2-aproximação para o problema da compra máxima limitada.

Demonstração. Começamos provando que o valor esperado pago por um comprador b é pelo menos $\frac{1}{2} \sum_{(i,S) \in \mathcal{S}: b \in S} P_{(i,S)} x_{(i,S)}$. Considere uma ordenação não-crescente (em $P_{(i,S)}$) das estrelas (i, S) que contém b e seja k o número de tais estrelas. Se $k = 0$, então o resultado é trivialmente válido. De agora em diante, assumimos que $k > 0$.

Denotamos a ℓ -ésima estrela nessa ordenação simplesmente por ℓ , seu preço por P_ℓ e sua variável primal por x_ℓ . Além disso, denotamos por E_ℓ o evento no qual a estrela ℓ foi escolhida por STARROUNDING.

Seja $L(b)$ o lucro que obtemos a partir do comprador b . Provamos, por indução em $k-\ell$, que $\mathbb{E}[L(b) | \overline{E}_1, \overline{E}_2, \dots, \overline{E}_{\ell-1}] \geq \frac{1}{2} \sum_{i=\ell}^k P_i x_i$ para todo $1 \leq \ell \leq k$. A partir disso, concluímos que $\mathbb{E}[L(b)] \geq \frac{1}{2} \sum_{i=1}^k P_i x_i$.

Note que, para $1 \leq \ell \leq k$, $\mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}, E_\ell] = P_\ell$ pois o algoritmo STARROUNDING aloca um item com preço P_ℓ para o comprador b . Ademais, seja $y_\ell = \mathbb{P}(E_\ell|\bar{E}_1, \dots, \bar{E}_{\ell-1})$ e note que $y_\ell \geq x_\ell$. Se $\ell = k$, então

$$\mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}] = P_k \mathbb{P}(E_k|\bar{E}_1, \dots, \bar{E}_{k-1}) \geq P_k x_k \geq \frac{1}{2} P_k x_k.$$

Para $\ell < k$, assumamos que o resultado é válido para $\ell + 1$. Temos que $\mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}, \bar{E}_\ell] \geq \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i$. Assim,

$$\begin{aligned} \mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}] &= \mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}, E_\ell] y_\ell \\ &\quad + \mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}, \bar{E}_\ell] (1 - y_\ell) \\ &\geq P_\ell y_\ell + \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i (1 - y_\ell) \\ &= \left(P_\ell - \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i \right) y_\ell + \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i \end{aligned}$$

Agora, note que $P_\ell \geq \sum_{i=\ell+1}^k P_i x_i$ já que $\sum_{i=\ell+1}^k x_i \leq 1$ e $P_\ell \geq P_i$ para todo $i > \ell$. Assim,

$$\begin{aligned} \left(P_\ell - \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i \right) y_\ell + \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i &\geq \frac{1}{2} P_\ell y_\ell + \frac{1}{2} \sum_{i=\ell+1}^k P_i x_i \\ &\geq \frac{1}{2} \sum_{i=\ell}^k P_i x_i. \end{aligned}$$

Portanto, $\mathbb{E}[L(b)|\bar{E}_1, \dots, \bar{E}_{\ell-1}] \geq \frac{1}{2} \sum_{i=\ell}^k P_i x_i$. Segue, pelo princípio da indução, que $\mathbb{E}[L(b)] \geq \frac{1}{2} \sum_{i=1}^k P_i x_i$.

Considere então o lucro esperado L de uma solução obtida pelo STARROUNDING. Pela linearidade da esperança, temos que $\mathbb{E}[L] = \sum_{b \in B} \mathbb{E}[L(b)]$ e concluímos que

$$\mathbb{E}[L] \geq \frac{1}{2} \sum_{b \in B} \sum_{(i,S) \in S: b \in S} P_{(i,S)} x_{(i,S)} = \frac{1}{2} \sum_{(i,S) \in S} |S| P_{(i,S)} x_{(i,S)}$$

e o resultado segue. \square

Assim como no caso do algoritmo apresentado por Aggarwal et al. [AFMZ04] para o problema da compra máxima, Fernandes e Schouery [FS14] mostraram que é possível desaleatorizar o algoritmo STARROUNDING. Ademais, note que a razão de aproximação de ambos os algoritmos é a mesma, porém o algoritmo de Fernandes e Schouery funciona para o caso mais geral de oferta limitada diferentemente do algoritmo de Aggarwal et al.

Quando cada item tem exatamente uma cópia, podemos resolver o problema em tempo polinomial. Porém, Briest e Krysta [BK07] mostraram que o problema da compra máxima é APX-difícil mesmo quando nos restringimos a instâncias onde cada item tem no máximo 2 cópias.

5.2 O problema da compra por preferência

A seguir, apresentamos um novo problema de precificação onde os compradores têm preferências (ordenações totais) em relação aos itens. Para um comprador b , denotamos por $\pi_b: I \rightarrow I$ sua preferência em relação aos itens de forma que, se b prefere um item i a um item i' então $\pi_b(i) < \pi_b(i')$. Ou seja, π_b é uma permutação dos itens ordenados por preferência.

Definição. No *problema da compra por preferência*, dados uma valoração v e um vetor π de preferências dos compradores em relação aos itens, desejamos encontrar uma precificação p e uma alocação x que maximiza o lucro de forma que cada comprador b receba, se existir, o item preferido (de acordo com π_b) dentre os que são viáveis.

Ademais, adotamos a hipótese de que, para todo comprador b e quaisquer dois itens i e i' , se b prefere i a i' (isto é, $\pi_b(i) < \pi_b(i')$) então $v_{ib} \geq v_{i'b}$. Isto é, se um comprador prefere um item i a um item i' então ele está disposto a pagar pelo item i pelo menos o valor que ele está disposto a pagar pelo item i' .

Uma variante interessante para problemas de precificação é impor uma *escada de preços*, isto é considerar apenas precificações onde $p_1 \geq p_2 \geq \dots \geq p_n$ onde n é o número de itens a serem vendidos. A motivação de tal restrição é que por vezes o vendedor deseja ter uma precificação onde itens melhores tenham preços maiores.

Definição. O problema da compra por preferência com escada de preços é uma variante do problema da compra máxima limitada em que os preços estão restritos a serem não-crescentes, isto é, $p_1 \geq \dots \geq p_n$.

Podemos considerar uma escada de preços também para o problema da compra máxima.

Definição. O problema da compra máxima com escada de preços é uma variante do problema da compra máxima em que os preços estão restritos a serem não-crescentes, isto é, $p_1 \geq \dots \geq p_n$.

Aggarwal et al. [AFMZ04] mostraram que existe uma conexão importante entre o problema da compra por preferência com escada de preços com o problema da compra máxima com escada de preços.

Lema 5.2.1. Dada uma α -aproximação para o problema da compra máxima com escada de preços, é possível projetar uma α -aproximação para o problema da compra por preferência com escada de preços.

Demonstração. Seja (v, π) uma instância do problema da compra por preferência com escada de preços. Considere um comprador b e dois itens i e i' tais que $i < i'$ (ou seja, $p_i \leq p_{i'}$ pela escada de preços) e $\pi_b(i) < \pi_b(i')$, isto é, b prefere i a i' . Note que, se i' é viável para b , então i é viável para b , já que $p_i \leq p_{i'} \leq v_{i'b} \leq v_{ib}$ (a última desigualdade segue da hipótese de que se $\pi_b(i) < \pi_b(i')$ então $v_{ib} \geq v_{i'b}$). Considere então a matriz v' obtida através da modificação da matriz v de forma que para todo comprador b e pares de itens i e i' tais que $i < i'$ e $\pi_b(i) < \pi_b(i')$, definimos $v'_{i'b} = 0$ e as outras entradas de v' não são modificadas em relação a v .

Seja (x, p) uma solução para o problema da compra máxima com escada de preços para a instância v' . Como consideramos oferta ilimitada, podemos, sem perda de generalidade, supor que (x, p) aloca a cada comprador b o item viável de menor índice (note que esse item, pela escada de preço, tem preço máximo entre os viáveis). Com isso, se o comprador b recebe o item i' então não existe item $i < i'$ e $\pi_b(i) < \pi_b(i')$ e, portanto, i' é o item viável preferido de b . Assim, (x, p) é uma solução viável para o problema da compra por preferência com escada de preços. Concluímos que o lucro de uma solução ótima para o problema da compra por preferência com escada de preços com instância (v, π) coincide com o lucro de uma solução ótima para o problema da compra máxima com escada de preços com instância v . Ademais, uma solução (x, p) obtida de

uma α -aproximação para o problema da compra máxima com escada de preços para a instância v é uma solução com razão de aproximação α para o problema da compra por preferência com escada de preços. \square

Aggarwal et al. [AFMZ04] apresentaram um esquema de aproximação polinomial para o problema da compra máxima com escada de preços. Com isso, concluímos que existe um esquema de aproximação polinomial para o problema da compra por preferência com escada de preços.

Teorema 5.2.2. Existe um esquema de aproximação polinomial para o problema da compra máxima com escada de preços.

Corolário 5.2.3. Existe um esquema de aproximação polinomial para o problema da compra por preferência com escada de preços.

Briest e Krysta [BK07] mostraram que o problema da compra máxima com escada de preços é fortemente NP-difícil mesmo se, para cada comprador b , existirem no máximo dois itens i tal que $v_{ib} \neq 0$. O mesmo resultado vale para o problema da compra por preferências com escada de preços. Assim, não é possível obter um FPTAS para tais problemas exceto se $P = NP$.

5.2.1 Oferta limitada

A partir do esquema de aproximação polinomial para o problema da compra máxima com escada de preços apresentado por Aggarwal et al. [AFMZ04] é possível obter uma 4-aproximação para o caso de oferta limitada.

Como o problema da compra máxima com escada de preços é fortemente NP-difícil (como mostrado por Briest e Krysta [BK07]), sabemos que o problema da compra máxima limitada com escada de preços também é fortemente NP-difícil por se tratar de um problema mais geral. Porém, o melhor resultado de aproximação conhecido para tal problema é uma família de algoritmos indexada por ε tal que cada membro dessa família é uma $2 + \varepsilon$ -aproximação [FS14]. Assim, resta a dúvida se o problema da compra máxima limitada com escada de preços é APX-difícil ou se existe um PTAS para o mesmo.

Em relação ao problema da compra por preferência com escada de preços com oferta limitada, é importante notar que a prova do Lema 5.2.1

não é válida para o caso onde temos oferta limitada. Isso acontece porque, no caso de oferta limitada, pode ser que um comprador b receba um item i' e exista um item i viável tal que $\pi_b(i) < \pi_b(i')$. Esse caso pode acontecer quando i já foi vendido para o número máximo de compradores. Assim, os resultados de aproximação para o problema da compra máxima limitada com escada de preços não se estendem para esse problema.

5.3 O problema da compra mínima

Consideramos agora o problema da compra mínima, onde cada comprador recebe, se existir, o item mais barato dentro dos que são viáveis.

Definição. No *problema da compra mínima*, dada uma valoração v , desejamos encontrar uma precificação p e uma alocação x que maximiza o lucro de forma que cada comprador b recebe um item de preço mínimo dentre dos que são viáveis.

A seguir, apresentamos o algoritmo PREÇOS-IGUAIS introduzido por Aggarwal et al. [AFMZ04] que devolve uma solução H_m -aproximada para o problema da compra mínima onde m é o número de compradores. Tal algoritmo pressupõe a existência de dois outros algoritmos LUCRO que dados I, B, v e uma precificação p calcula o lucro máximo para precificação e ALOC que dados I, B, v e uma precificação p calcula uma alocação que maximiza o lucro. Note que ambos os algoritmos são triviais no caso do problema da compra mínima, principalmente no caso de oferta ilimitada. A seguir, apresentamos o algoritmo PREÇOS-IGUAIS.

PREÇOS-IGUAIS(I, B, v)

```

1   $\ell = 0$ 
2   $p_1 = p_2 = \dots = p_n = 0$ 
3  para cada comprador  $b$  em  $B$ 
4       $V_b = \max\{v_{ib} : i \in I\}$ 
5       $p'_1 = p'_2 = \dots = p'_n = V_b$ 
6      se LUCRO( $I, B, v, p'$ )  $> \ell$ 
7           $\ell = \text{LUCRO}(I, B, v, p')$ 
8           $p = p'$ 
9  devolva (ALOC( $I, B, v, p$ ),  $p$ )
```

Teorema 5.3.1. O algoritmo PREÇOS-IGUAIS é uma H_m -aproximação para o problema da compra mínima, onde $H_m = \sum_{i=1}^m 1/i$, isto é, o m -ésimo número harmônico.

Demonstração. Por simplicidade, considere que os compradores estão ordenados de forma que $V_1 \geq V_2 \geq \dots \geq V_m$. Note que, para $p'_1 = p'_2 = \dots = p'_n = V_k$, $\text{LUCRO}(I, B, v, p') \geq k \cdot V_k$ já que existem itens viáveis para pelo menos k compradores e cada item tem preço V_k . Assim, o algoritmo escolhe $k^* \in \arg \max\{k \cdot V_k\}$ e obtém lucro $L = k^* \cdot V_{b_{k^*}}$. Assim, $L \geq k \cdot V_k$ para todo k e, portanto, temos que $\text{OPT} \leq \sum_{k=1}^m V_k \leq \sum_{k=1}^m L/j = H_m L$ e o resultado segue. \square

Note que essa análise não funciona para oferta limitada, já que não podemos garantir que se o preço for V_k então k itens serão vendidos. Ademais, é fácil ver que esse algoritmo é também uma H_m -aproximação para vários problemas de precificação abordados nesse capítulo. Isso porque sua análise baseia-se somente no fato de que se os preços dos itens forem V_k então k compradores receberão itens e que o lucro de uma solução ótima é limitada superiormente $\sum_{k=1}^m V_k$. Assim, temos o seguinte resultado.

Corolário 5.3.2. O algoritmo PREÇOS-IGUAIS é uma H_m -aproximação para o problema da compra máxima, para o problema da compra por preferências e para o problema da compra mínima. Ademais, o algoritmo PREÇOS-IGUAIS é uma H_m -aproximação para as variantes de tais problemas onde exigimos uma escada de preços.

Infelizmente, o problema da compra mínima é muito difícil de ser aproximado. Uma restrição que podemos impor na instância é exigir que, para cada comprador b exista um valor V_b tal que, para todo item i , ou $v_{ib} = V_b$ ou $v_{ib} = 0$. Isto é, cada comprador tem interesse em um determinado conjunto de itens e dá o mesmo valor para cada um desses itens. Tal restrição é chamada de *orçamentos uniformes*. Seja ℓ um limitante superior no número de valorações não nulas de cada comprador. Mesmo no caso de orçamentos uniformes, Chalermsook et al. [CCKK12] mostraram que o problema não pode ser aproximado por $\Omega(\ell^{1/2-\varepsilon})$ a não ser que $P = NP$. Ademais, existem resultados outros resultados [CCKK12, CLN13] mostrando que o resultado acima é assintoticamente ótimo se considerarmos algumas hipóteses mais fortes do que $P \neq NP$.

5.4 O problema da precificação livre de inveja

Nessa seção abordamos outro modelo de precificação, o *problema da precificação livre de inveja*. Nesse problema, se um comprador b recebe um item i então esse item é viável e $v_{ib} - p_i$ é máximo. Assim, não existe um item i' que o comprador b “tenha inveja”. Novamente, o objetivo do vendedor é maximizar o seu lucro. Começamos definindo o conceito de alocação livre de inveja.

Definição. Dada uma precificação p e uma alocação x , dizemos que x é uma *alocação livre de inveja* para p se, para cada comprador b , ou b recebe um item viável i e $v_{ib} - p_i$ é máximo ou b não recebe item e $p_i \geq v_{ib}$ para todo item i .

Estamos prontos para definir o nosso problema de interesse.

Definição. No *problema da precificação livre de inveja*, dada uma valoração v , desejamos encontrar uma precificação p e uma alocação x que maximiza o lucro de forma que x seja uma alocação livre de inveja para p .

Sem perda de generalidade, consideramos que cada item tem apenas uma cópia. De fato, considere uma instância (v, C) onde cada item tem C_i cópias. Podemos criar uma nova instância w onde, para cada item i original, existem C_i novos itens j com $w_{jb} = v_{ib}$. Nessa nova instância, se um comprador b recebe um item j que foi criado à partir do item original i e existe um outro item k que também foi criado à partir do item original i , então $w_{jb} - p_j \geq w_{kb} - p_k$, ou seja, $p_j \leq p_k$. Ademais, se k é vendido para outro comprador, então $p_j = p_k$. Assim, todos os itens vendidos que foram criados à partir do mesmo item original têm o mesmo preço, como desejado.

O problema da precificação livre de inveja está relacionado com o conceito de equilíbrio walrasiano apresentado no Capítulo 4. Porém, como estamos considerando demanda unitária, precisamos redefinir alguns conceitos para tal cenário.

Definição. Dada uma valoração v e uma precificação p , a *demanda* de um comprador b é o conjunto de itens viáveis i tal que $v_{ib} - p_i$ é máximo.

Note que, numa solução viável do problema da precificação livre de inveja, cada comprador que recebe um item, recebe um item de sua demanda. Assim, uma solução viável de tal problema onde os preços dos itens não vendidos é igual a 0 é um equilíbrio walrasiano.

Definição. Dada uma valoração v , um *equilíbrio walrasiano* é um par (x, p) onde x é uma alocação livre de inveja para p tal que todo item não vendido tem preço zero.

Lembre-se que um emparelhamento é um conjunto de arestas de um grafo que não tem pontas em comum. O conceito de equilíbrio walrasiano para demanda unitária está relacionado com o problema do emparelhamento bipartido de peso máximo, onde desejamos encontrar um emparelhamento de peso máximo em um grafo bipartido. Nesse caso, podemos interpretar um emparelhamento como uma alocação de itens a compradores, onde desejamos vender cada item a no máximo um comprador de forma que cada comprador receba no máximo um item. Ademais queremos maximizar $\sum_{(i,b) \in M} v_{ib}$ onde M é um emparelhamento, ou seja, o bem-estar social.

A seguir apresentamos uma formulação por programação linear inteira para o problema do emparelhamento bipartido de peso máximo, na qual temos um vetor binário x de variáveis indexado por $I \times B$. Para todo $(i, b) \in I \times B$, $x_{i,b}$ é 1 se e somente se o item i é vendido para o comprador b . O objetivo é determinar x que

$$\begin{aligned} \max \quad & \sum_{i \in I} \sum_{b \in B} v_{ib} x_{ib} \\ \text{sujeito a} \quad & \sum_{i \in I} x_{ib} \leq 1 \quad \forall b \in B, \\ & \sum_{b \in B} x_{ib} \leq 1 \quad \forall i \in I, \\ & x_{(i,b)} \in \{0, 1\} \quad \forall i \in I, \forall b \in B. \end{aligned}$$

Um fato interessante é que sempre existe uma solução inteira ótima para a relaxação linear do programa linear inteiro acima. Assim, é possível resolver o problema do emparelhamento bipartido de peso máximo em tempo polinomial. Ademais, sempre existe um equilíbrio walrasiano para v (que pode ser encontrado em tempo polinomial). A prova desse fato segue na mesma linha da prova do Primeiro e do Segundo Teorema do Bem-Estar Social do Capítulo 4.

A seguir estendemos o conceito de equilíbrio walrasiano para considerar um preço mínimo de venda para cada item.

Definição. Dada uma valoração v e um racional positivo r , um equilíbrio walrasiano com preço de reserva r é um par (x, p) onde x é uma alocação livre de inveja para p tal que $p_i \geq r$ para todo item i e todo item não alocado por x tem preço r .

Ademais, note que em um equilíbrio walrasiano com preço de reserva r pode acontecer de um comprador b ter demanda não-vazia e não receber item. Isso acontece somente se, para todo item i na demanda de b , $p_i = v_{ib}$. Assim, definimos o conceito de um equilíbrio walrasiano maximal da seguinte forma.

Definição. Dada uma valoração v , um equilíbrio walrasiano é *maximal* se para todo comprador b , se existe um item i na demanda de b não-alocado para um comprador b então b recebe um item de sua demanda.

Guruswami et al. [GHK⁺05] mostraram como transformar um algoritmo ALGO que computa um equilíbrio walrasiano em um algoritmo ALGO-RESERVE que computa um equilíbrio walrasiano com preço de reserva r maximal.

ALGO-RESERVE(v, r)

- 1 $w = v$
- 2 **para cada** item i **em** I
- 3 adicione dois novos compradores c_i e d_i à w de forma que
 $w_{ic_i} = w_{id_i} = r_i$ e $w_{i'c_i} = w_{i'd_i} = 0$ para todo item $i' \neq i$.
- 4 $(x, p) = \text{ALGO}(w)$
- 5 $y = x$
- 6 **para cada** item i **em** I
- 7 remova de y as atribuições de itens a c_i e d_i
- 8 **para cada** comprador b **em** B que não recebe item
- 9 **se** existe item i na demanda de b não-vendido
- 10 aloque i para b em y
- 11 **devolva** (y, p)

Teorema 5.4.1. Se ALGO é um algoritmo que com entrada v computa um equilíbrio walrasiano para v então ALGO-RESERVE com entrada (v, r) computa um equilíbrio walrasiano com preço de reserva r maximal para v .

Demonstração. Considere (x, p) . Para cada item i , pelo menos um dos dois compradores c_i e d_i não recebe i . Assim, como (x, p) é livre de inveja,

temos que $p_i \geq r$. Como $p_i > 0$ e (x, p) é um equilíbrio walrasiano, temos que i é vendido em (x, p) . Assim, se i não é vendido em y então ele é vendido para um dos dois compradores c_i e d_i em x , de onde concluímos que $p_i = r_i$ para todo item não vendido em (y, p) . Ademais, note que como (x, p) é livre de inveja para v , temos que (y, p) é livre de inveja para w . Por fim, note que, por construção, (y, p) é maximal. \square

Ademais, Guruswami et al. [GHK⁺05] mostraram que existe uma conexão entre um emparelhamento de peso máximo para v e um equilíbrio walrasiano com preço de reserva r .

Lema 5.4.2. Seja M um emparelhamento de peso máximo para v . Dado um valor r , seja k_r o número de arestas de M com valor pelo menos r . Se (x, p) é um equilíbrio walrasiano com preço de reserva r maximal, então o lucro de (x, p) é pelo menos $rk_r/2$.

Demonstração. Vamos mostrar que, para toda aresta (i, b) de M com valor pelo menos r , temos que se i não está vendido, então b recebe um item. Se i não é vendido, então $p_i = r$. Ademais, se i está na demanda de b , então b recebe um item e se i não está na demanda, então $v_{ib} \geq r = p_i$ e, portanto, a demanda de b é não-vazia já que i é viável para b , ou seja, b recebe um item. Assim, temos que, o número de vendas realizadas é pelo menos $k_r/2$, já que uma venda pode ser contada no máximo duas vezes (uma do ponto de vista do comprador e outra do ponto de vista do item). Ademais, (x, p) vende itens com preço pelo menos r para pelo menos $k_r/2$ compradores e, portanto, o lucro é pelo menos $rk_r/2$. \square

Seja MAXWEQ-RESERVE um algoritmo que dado v e um preço de reserva r , devolve um equilíbrio walrasiano maximal com preço de reserva r para v . Utilizando o resultado acima, é possível projetar um algoritmo de aproximação para o problema da precificação livre de inveja. A ideia é encontrar um equilíbrio walrasiano com preço de reserva r maximal onde r é escolhido apropriadamente.

EF(v)

- 1 seja M um emparelhamento de peso máximo para v
- 2 seja $r_1 \geq r_2 \geq \dots \geq r_\ell$ os valores das arestas de M
- 3 **para cada** $j = 1$ **ate** ℓ
- 4 $(x^j, p^j) = \text{MAXWEQ-RESERVE}(v, r_j)$
- 5 **devolva** (x^k, p^k) que maximize o lucro com $1 \leq k \leq \ell$

Teorema 5.4.3. EF é uma $(2H_\rho)$ -aproximação para o problema da precificação livre de inveja onde $\rho = \min\{n, m\}$.

Demonstração. É fácil ver que o algoritmo devolve uma precificação livre de inveja e que executa em tempo polinomial. Basta então comparar o lucro da solução obtida com o lucro de uma solução ótima. O lucro de uma solução ótima é no máximo o valor de um emparelhamento de peso máximo. Isto é,

$$\text{OPT} \leq \sum_{j=1}^{\ell} r_j \leq 2 \sum_{j=1}^{\ell} \frac{L(x^j, p^j)}{j} \leq 2H_{|M|}L \leq 2H_\rho L,$$

onde a desigualdade $2L(x^j, p^j)/j \geq r_j$ segue do lema anterior. \square

Quando consideramos orçamentos uniformes, o problema da precificação livre de inveja e o problema da compra mínima coincidem. Por isso, os limitantes inferiores mencionados anteriormente [CCKK12, CLN13] são válidos também para o problema da precificação livre de inveja.

Compartilhamento de Custos

A maioria dos jogos analisados nos outros capítulos considera situações onde dado um conjunto de jogadores, cada um age de forma individual tentando minimizar seu custo (ou maximizar sua utilidade) em resposta as escolhas de outros jogadores. Neste capítulo consideramos jogos cooperativos, onde os jogadores também agem de forma a maximizar sua utilidade, mas isto pode se dar também de forma cooperativa. Nos jogos vistos anteriormente um jogador não muda de estratégia a menos que seja vantajoso para si, considerando que todos os demais jogadores continuem nas suas estratégias atuais. Pode ser que em um jogo cooperativo em um determinado perfil de estratégias (por exemplo em um equilíbrio puro de Nash) um determinado jogador sozinho não tenha benefício se mudar de estratégia, mas se uma *coalização* de jogadores mudar de estratégia, então este jogador terá algum benefício. Neste caso a utilidade para cada jogador da coalizãõ deve ser tão boa quanto antes e ao menos um jogador deve ter sua utilidade aumentada. Em geral, em jogos cooperativos, o interesse está em como partilhar o resultado do jogo entre jogadores de tal forma que seja vantajoso para os jogadores formarem a *grande coalizãõ* constituída de todos os jogadores.

Primeiramente vamos definir um jogo cooperativo com utilidades não transferíveis. Neste jogo temos um conjunto de jogadores A e chamamos de coalizãõ qualquer $S \subseteq A$. No caso particular em que $S = A$ chamamos S

de grande coalizão. No jogo cooperativo com utilidades não transferíveis, para cada coalizão é especificado quanto cada jogador ganha na coalizão.

Definição. Um *jogo cooperativo com utilidades não transferíveis* (jogo UNT) consiste de um par (A, v) onde

- A é um conjunto de n jogadores, indexados por j e
- $v: 2^A \rightarrow \mathbb{R}^{|A|}$ é uma função que especifica para cada possível coalizão $S \subseteq A$ de jogadores, o valor de utilidade para cada jogador da coalizão. Assume-se que $v(\emptyset) = 0$.

Note que v mapeia para cada possível coalizão S um vetor $v(S) \in \mathbb{R}^{|A|}$, onde denotamos por $v(j)$ o valor de utilidade de um jogador $j \in S$ e para cada $j \notin S$ temos $v(j) = 0$.

De forma similar definimos jogos onde há um custo e jogadores formam coalizões com objetivo de reduzir seus custos. Quanto se tratar de uma função de custo denotamos esta por $c: 2^A \rightarrow \mathbb{R}^{|A|}$.

Um caso particular de jogo cooperativo é quando a função de utilidade v (ou custo dependendo do jogo), mapeia cada coalizão S para um valor real, ou seja $v: 2^A \rightarrow \mathbb{R}$. Neste caso podemos definir arbitrariamente como será a divisão do valor total $v(S)$ entre os jogadores da coalizão S . Isto é o que se chama de *jogo cooperativo com utilidades transferíveis*.

Definição. Um *jogo cooperativo com utilidades transferíveis* (jogo UT) consiste de um par (A, v) onde

- A é um conjunto de n jogadores, indexados por j e
- $v: 2^A \rightarrow \mathbb{R}$ é uma função que especifica para cada possível coalizão $S \subseteq A$ de jogadores, o valor de utilidade a ser distribuído na coalizão S . Assume-se que $v(\emptyset) = 0$.

O foco deste capítulo se dará em jogos UT que são os mais estudados na literatura e podem modelar diversas situações práticas. O foco em geral está em como dividir o valor de utilidade entre os jogadores de uma coalizão de tal forma a se garantir propriedades interessantes no resultado final do jogo.

Da mesma forma como anteriormente, se em um determinado jogo temos custos ao invés de utilidade, consideramos então um jogo UT com uma função de custo c ao invés de utilidade.

6.1 Soluções para jogos cooperativos

Pessoas diferentes podem propor diferentes definições para o que é uma solução em um jogo cooperativo. No caso de jogos não-cooperativos, como os estudados em capítulos anteriores, um conceito de solução é o equilíbrio de Nash, que é bem aceito pela comunidade por ter um significado prático e lógico. A questão central em jogos cooperativos está em como dividir o resultado $v(A)$ do jogo entre os jogadores de forma “justa”. No caso de jogos cooperativos a comunidade da área vem debatendo o conceito de solução há décadas. Nos concentramos nas principais propriedades aceitas atualmente na comunidade sobre o que é uma solução “justa”.

Dado um jogo cooperativo (A, v) definimos $\psi: \mathbb{R}^{2^{|A|}} \rightarrow \mathbb{R}^{|A|}$ como uma função de compartilhamento que mapeia a função de utilidade do jogo (A, v) para um vetor real $\alpha \in \mathbb{R}^{|A|}$. Cada posição α_j deste vetor indica quanto ganha o jogador j , ou melhor, qual a parte que cabe ao jogador j do valor total $v(A)$ do jogo. Usamos a notação α_j como simplificação para $\psi_j(v)$.

A seguir denotamos algumas propriedades que a função de compartilhamento ψ deveria ter.

Definição. Dado um jogo cooperativo (A, v) o conjunto de *pagamentos viáveis* são aqueles em

$$\left\{ \alpha \in \mathbb{R}^{|A|} \mid \sum_{j \in A} \alpha_j \leq v(A) \right\}.$$

Ou seja, o conjunto de pagamentos viáveis é aquele formado por pagamentos que não distribuem mais do que é possível, que é o valor da grande coalizão.

Definição. Dado um jogo cooperativo (A, v) o conjunto de *pagamentos pré-imputáveis* são aqueles em

$$\left\{ \alpha \in \mathbb{R}^{|A|} \mid \sum_{j \in A} \alpha_j = v(A) \right\}.$$

Tais vetores são *eficientes* no sentido econômico pois distribuem entre os jogadores todo o valor possível. Vetores de pagamentos satisfazendo

tal propriedade também são conhecidos como vetores de *orçamento equilibrado*. Note que um pagamento pré-imputável é também um pagamento viável.

Definição. Dado um jogo cooperativo (A, v) o conjunto de *pagamentos imputáveis* são aqueles em $\{\alpha \in \mathbb{R}^{|A|} \mid \forall j, \alpha_j \geq v(j)\}$.

Um pagamento é imputável se garantidamente cada jogador irá receber pelo menos aquilo que ganharia caso formasse uma coalizão sozinho.

Dados dois jogadores i e j , dizemos que estes são *intercambiáveis* se a contribuição deles são sempre iguais para as coalizões que não os contém, ou seja, para todo S tal que $i \notin S$ e $j \notin S$, temos $v(S \cup i) = v(S \cup j)$.

Definição. Uma função de pagamento ψ satisfaz a *simetria* se para i e j intercambiáveis então $\psi_i(v) = \psi_j(v)$.

Esta definição garante que se dois jogadores são intercambiáveis então devem receber o mesmo pagamento.

Um jogador é *dummy* se o que ele acrescenta para cada possível coalizão é exatamente o seu valor sozinho, ou seja, j é *dummy* se para qualquer S tal que $j \notin S$, temos $v(S \cup j) = v(S) + v(j)$. O princípio do jogador *dummy* especifica que este deve receber como pagamento exatamente $v(j)$.

Definição. Uma função de pagamento ψ satisfaz esta propriedade se dado jogador j que é *dummy* então $\psi_j(v) = v(j)$.

Suponha um conjunto de jogadores em um jogo cooperativo sujeito a duas funções de utilidades v_1 e v_2 . O princípio da *aditividade* estabelece que a função de pagamento ψ para o jogo com função de utilidade $(v_1 + v_2)$ deve gerar como pagamento para cada jogador j exatamente a soma dos pagamentos nos jogos com v_1 e v_2 .

Definição. Dados quaisquer dois jogos (A, v_1) e (A, v_2) onde altera-se apenas a função de utilidade, dizemos que a função de pagamento ψ é *aditiva* se para todo jogador j vale $\psi_j(v_1 + v_2) = \psi_j(v_1) + \psi_j(v_2)$.

6.2 A solução de Shapley

Um resultado devido a Shapley [Sha52] estabelece como computar uma função de pagamento ψ que satisfaz a pré-imputação, simetria, jogador

dummy e aditividade para um jogo cooperativo UT. Além do mais, tal resultado estabelece que esta é a única função de pagamento satisfazendo tais propriedades.

Dado um jogo cooperativo com utilidades transferíveis (A, v) definimos como o valor de Shapley para o jogador j como

$$\psi_j(v) = \frac{1}{|A|!} \sum_{S \subseteq A \setminus \{j\}} |S|!(|A| - |S| - 1)! [v(S \cup j) - v(S)]$$

Teorema 6.2.1. Dado um jogo cooperativo com utilidades transferíveis (A, v) a única função de pagamento que satisfaz as propriedades de pré-imputação, simetria, jogador *dummy* e aditividade é aquela definida pelo valor de Shapley.

Exemplo 6.2.2. Como exemplo de aplicação do valor de Shapley considere o seguinte jogo: temos os jogadores Arnaldo (A), Beatriz (B), e Célio (C) interessados em comprar um pote de sorvete. A possui R\$7,00, B possui R\$8,00, e C possui R\$8,00. Existem dois potes de sorvete, um de R\$15,00 com 700 gramas e outro de R\$20,00 com 1000 gramas. Temos a seguinte função de mapeamento de utilidade:

$$\begin{array}{lll} v(\emptyset) = 0 & v(\{A\}) = 0 & v(\{B\}) = 0 \\ v(\{C\}) = 0 & v(\{A, B\}) = 700 & v(\{A, C\}) = 700 \\ v(\{B, C\}) = 700 & v(\{A, B, C\}) = 1000 & \end{array}$$

O valor de Shapley para o jogador A pode ser computado como

$$\frac{0! \cdot (2)! \cdot [0] + 1! \cdot (1)! \cdot [700] + 1! \cdot (1)! \cdot [700] + 2! \cdot (0)! \cdot [300]}{6!} = \frac{1000}{3}.$$

O mesmo valor será obtido para os jogadores B e C . Neste caso a função de pagamento consiste em distribuir igualmente as 1000 gramas de sorvete da coalizão entre os jogadores.

6.3 Compartilhamento de custos

Note que a função de utilidade v pode mapear coalizões para valores negativos. A interpretação que damos neste caso é que v representa o custo de uma coalizão. Como mencionado anteriormente, utilizamos uma função de

custo c neste caso para uma interpretação mais clara. Os jogadores querem construir algum bem e uma autoridade central deseja definir como *compartilhar os custos* entre os jogadores de tal forma que todos jogadores participem da grande coalizção, ou seja não existirá benefício para que um subconjunto de jogadores $S \subset A$ formem uma coalizção separada.

Como exemplo considere uma comunidade que deseja construir um posto de saúde local. Há um custo que incorre da construção deste posto e os moradores da comunidade estão dispostos a arcar com os custos da construção desde que este traga o benefício esperado para os mesmos. Não necessariamente todos os moradores da comunidade participarão da construção do posto, e os que participarem poderão utiliza-lo mas terão que dividir entre si o custo de construção do posto. Para cada subconjunto de moradores $S \subseteq A$ que formarem uma coalizção, há um custo $c(S)$ de construção de um posto que atenda S . É natural pensar que o prefeito desta comunidade deseja que todos os moradores possam utilizar o posto. Portanto o seu objetivo é estabelecer um pagamento para cada cidadão de tal forma que todos formem uma grande coalizção única A . Ou seja, não haverá um subgrupo dos moradores que terá benefício se construírem um posto particular para eles. Chamamos jogos UT deste tipo como *jogos de compartilhamento de custos*.

Definição. Um *jogo de compartilhamento de custos* (A, c) é um jogo cooperativo UT onde o objetivo é estabelecer uma função de pagamento ψ de tal forma que os jogadores não tenham incentivo para saírem da grande coalizção.

Note que apesar de usarmos uma função de custo, as mesmas propriedades vistas na Seção 6.1 podem ser adaptadas para este caso. Podemos definir, por exemplo, a imputação como abaixo, já que não faz sentido cobrar mais de um jogador do que o custo que ele teria se formasse uma coalizção isolada.

Definição. Dado um jogo de compartilhamento de custos (A, c) o conjunto de *pagamentos imputáveis* são aqueles em $\{\alpha \in \mathbb{R}^{|A|} \mid \forall j, \alpha_j \leq c(j)\}$.

Da mesma forma poderíamos adaptar as demais propriedades. Mas no caso de jogos de compartilhamento de custos o conceito de solução mais utilizado na análise destes jogos é o de núcleo:

Definição. Considere um jogo cooperativo (A, c) de compartilhamento de custos onde c mapeia o custo de um bem para cada possível coa-

lização $S \subseteq A$. Seja $\alpha \in \mathbb{R}^{|A|}$ um vetor onde a posição j contém o valor α_j que deverá ser pago pelo jogador j na construção do bem de custo $c(A)$. Dizemos que um vetor $\alpha \in \mathbb{R}^{|A|}$ pertence ao *núcleo* se

- $\sum_{j \in A} \alpha_j = c(A)$. Esta propriedade é conhecida como *orçamento balanceado*.
- Para todo $S \subseteq A$ deve valer que $\sum_{j \in S} \alpha_j \leq c(S)$. Esta propriedade é conhecida como *estabilidade*.

A propriedade do orçamento balanceado garante que o custo do bem deve ser pago pelos jogadores (note que é a mesma propriedade da pré-imputação). A segunda propriedade garante que nenhuma coalizão $S \subset A$ terá benefício se sair da grande coalizão, pois o custo $c(S)$ do bem para atender apenas S é maior ou igual ao que os jogadores em S estão pagando na grande coalizão (o valor pago é $\sum_{j \in S} \alpha_j$).

Note que a função de pagamento definida conforme o valor de Shapley pode não ser estável. No exemplo do final da seção anterior a função não é estável, já que B e C poderiam se unir e estabelecer a divisão de 350 gramas de sorvete para cada, sendo vantajosa em relação a distribuição da grande coalizão.

O conceito de núcleo nos dá um conceito de solução equivalente ao que é conhecido como equilíbrio forte de Nash pois garante que nenhuma coalizão de jogadores terá um benefício se desviar da solução atual, que é a grande coalizão. Duas questões em geral são analisadas em jogos cooperativos com compartilhamento de custos:

- (a) Será que sempre há uma função de pagamento no núcleo?
- (b) Será que tal função é única?

Infelizmente no caso geral a resposta para ambas as questões é não. Por outro lado um resultado devido a Bondareva e Shapley [Sha67] estabelece uma condição necessária e suficiente para a existência de um núcleo não vazio. Primeiramente vamos definir o que é uma *coleção de pesos balanceado*:

Definição. Uma coleção de pesos balanceados λ para A é um vetor de $\mathbb{R}_+^{2^{|A|}}$ que atribui um peso não negativo λ_S para cada subconjunto $S \subseteq A$ de tal forma que para cada jogador $j \in A$ vale que $\sum_{S: j \in S} \lambda_S = 1$.

Exemplo 6.3.1. No jogo do final da seção anterior poderíamos estabelecer a seguinte coleção de pesos balanceados:

$$\begin{array}{lll} \lambda(\emptyset) = 0 & \lambda(\{A\}) = 1/2 & \lambda(\{B\}) = 1/2 \\ \lambda(\{C\}) = 1/2 & \lambda(\{A, B\}) = 0 & \lambda(\{A, C\}) = 0 \\ \lambda(\{B, C\}) = 0 & \lambda(\{A, B, C\}) = 1/2 & \end{array}$$

O seguinte resultado foi demonstrado por Bondareva e Shapley [Sha67] e sua prova requer apenas conhecimentos básicos sobre dualidade em programação linear (veja o Apêndice A.3).

Teorema 6.3.2. Um jogo cooperativo UT (A, c) possui um núcleo não vazio se e somente se para toda coleção de pesos balanceados λ for válido que $\sum_{S \subseteq A} \lambda_S \cdot c(S) \geq c(A)$.

Demonstração. Pela definição de núcleo, um jogo (A, c) possui núcleo não vazio se e somente se a solução do programa linear abaixo tem valor $c(A)$. Note que a solução ótima deste programa linear não pode ter valor maior do que $c(A)$ dado as suas restrições.

$$\begin{array}{ll} \max & \sum_{j \in A} \alpha_j \\ \text{sujeito a} & \sum_{j \in S} \alpha_j \leq c(S) \quad \forall S \subseteq A. \end{array}$$

As restrições do programa linear implicam que o pagamento α é estável, e além disso se a soma dos valores de α for $c(A)$, então temos um orçamento balanceado.

Agora pelo teorema forte da dualidade o valor ótimo deste programa linear é igual ao de seu dual que é apresentado abaixo:

$$\begin{array}{ll} \min & \sum_{S \subseteq A} \lambda_S \cdot c(S) \\ \text{sujeito a} & \sum_{S: j \in S} \lambda_S = 1 \quad \forall j \in A, \\ & \lambda_S \geq 0 \quad \forall S \subseteq A. \end{array}$$

Note que cada solução viável do dual corresponde a uma coleção de pesos balanceados (o inverso também é verdadeiro).

Se o núcleo é não vazio então o valor ótimo do primal é $c(A)$. Este também é o ótimo do dual e portanto todas as soluções viáveis do dual terão valor maior ou igual a $c(A)$. Portanto para cada coleção de pesos balanceados vale que $\sum_{S \subseteq A} \lambda_S \cdot c(S) \geq c(A)$.

Por outro lado se toda solução viável do dual tiver valor maior ou igual a $c(A)$, então a solução de valor mínimo tem valor exatamente $c(A)$ pois esta deve ter valor igual ao ótimo do primal que é no máximo $c(A)$. Portanto o núcleo é não vazio. \square

Portanto temos uma forma de checar se é possível ou não estabelecer um pagamento para jogadores em um jogo de compartilhamento de custos (A, c) que esteja no núcleo. Basta resolver um modelo de programação linear. Caso a função de custos c seja explicitamente dada na entrada então tal checagem se dá em tempo polinomial. Mas na maioria dos casos a função é implícita: por exemplo é a solução ótima $c(S)$ de um problema de otimização combinatória. Nestes casos temos um número exponencial de restrições, e para cada restrição devemos computar $c(S)$ que em muitos casos trata-se de um problema NP-difícil. Portanto esta checagem em muitos jogos é um problema intratável. Na próxima seção apresentamos um exemplo de jogo de compartilhamento de custos desta forma.

6.4 Jogo de localização de instalações

Um problema clássico na área de otimização combinatória é o problema da localização de instalações. Seja $G = (F \cup C, E)$ um grafo onde os vértices de F representam pontos onde pode-se abrir instalações e os vértices em C representam clientes, onde cada um precisa se conectar a uma instalação. Para cada instalação $i \in F$ existe um custo de abertura f_i caso seja aberto a instalação em i , e para cada par $i \in F, j \in C$ temos um valor d_{ij} que representa o custo para se conectar o cliente j com a instalação i . O objetivo do problema é determinar um subconjunto $F' \subseteq F$ de instalações a serem abertas para atender todos os clientes de C de tal

forma a se minimizar o custo total da solução definido como:

$$\sum_{i \in F'} f_i + \sum_{j \in C} \min_{i \in F'} d_{ij}.$$

Note que definido o subconjunto F' de instalações abertas, em uma solução de custo mínimo cada cliente se conecta com a instalação mais próxima.

Este problema pode ser utilizado para modelar diversos problemas práticos, como determinação de abertura de centros de distribuição para uma rede de lojas, abertura de hospitais para atender comunidades, localização de servidores de vídeo para atender clientes de sistema de vídeo sob demanda, etc. Este problema é bem investigado na literatura sob o ponto de vista de otimização combinatória e se trata de um clássico problema NP-difícil [Dre95].

Neste capítulo consideramos o Jogo da Localização de Instalações como um jogo UT de compartilhamento de custos, dado que este também pode modelar diversas situações práticas de estabelecimento da partilha de custos entre jogadores visando a construção de um bem ou obtenção de um serviço.

Definição. No *Jogo da Localização de Instalações* temos um grafo $G = (F \cup C, E)$ satisfazendo a desigualdade triangular, onde C é um conjunto de jogadores (representando clientes) e F representa um conjunto de instalações. Temos os custos f_i de abertura da instalação i , e custos d_{ij} da aresta entre $i \in F$ e $j \in C$. Para cada subconjunto $S \subseteq C$ temos o custo da coalizão S definido como

$$c(S) = \min_{F' \subseteq F} \left\{ \sum_{i \in F'} f_i + \sum_{j \in S} \min_{i \in F'} d_{ij} \right\}.$$

Os jogadores têm como objetivo construir soluções de custo mínimo para atendê-los.

Note que dado $S \subseteq C$, $c(S)$ representa o custo de uma solução ótima para o problema de localização de instalações sobre o grafo induzido por S e F .

Exemplo 6.4.1. Considere a Figura 6.1 (retirada de [NRTV07]). Temos duas instalações com custo de abertura 2, e três clientes onde os custos de

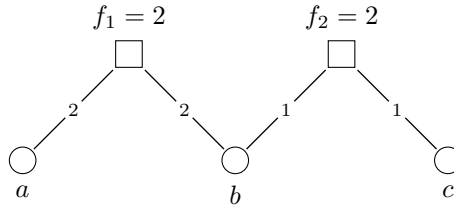


Figura 6.1: Um jogo de localização de instalações.

conexão estão especificados ao lado de cada aresta. A função de custo deve especificar o custo de cada possível coalizão. Para o exemplo temos os seguintes custos para cada possível coalizão:

$$\begin{array}{lll}
 c(\emptyset) = 0 & c(\{a\}) = 4 & c(\{b\}) = 3 \\
 c(\{c\}) = 3 & c(\{a, b\}) = 6 & c(\{a, c\}) = 7 \\
 c(\{b, c\}) = 4 & c(\{a, b, c\}) = 8 &
 \end{array}$$

Para este exemplo específico o núcleo é não vazio. Para ver isto considere o vetor de compartilhamento de custos $\alpha = (4, 2, 2)$. Esta distribuição satisfaz o orçamento balanceado e é fácil checar que para qualquer coalizão $S \subset C$, o custo desta é maior ou igual do que o pagamento α dos jogadores em S .

Por outro lado não é verdade que o núcleo é não vazio para todo jogo de localização de instalações. De fato, considere a Figura 6.2 (retirada de [NRTV07]). Nesta instância do jogo de localização de instalações, o núcleo é vazio.

Temos os seguintes custos para as coalizões:

$$\begin{array}{lll}
 c(\emptyset) = 0 & c(\{a\}) = 4 & c(\{b\}) = 3 \\
 c(\{c\}) = 3 & c(\{a, b\}) = 6 & c(\{a, c\}) = 5 \\
 c(\{b, c\}) = 4 & c(\{a, b, c\}) = 8 &
 \end{array}$$

Se existir um vetor α no núcleo do jogo do exemplo da Figura 6.2, então as seguintes desigualdades devem ser válidas:

$$\begin{array}{lll}
 \alpha_a + \alpha_b & \leq c(\{a, b\}) & = 6 \\
 \alpha_b + \alpha_c & \leq c(\{b, c\}) & = 4 \\
 \alpha_a + \alpha_c & \leq c(\{a, c\}) & = 5
 \end{array}$$

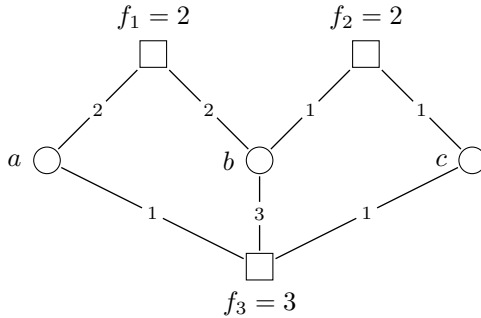


Figura 6.2: Um jogo de localização de instalações com núcleo vazio.

Somando estas desigualdades obtemos

$$2\alpha_a + 2\alpha_b + 2\alpha_c \leq 15$$

e portanto

$$\alpha_a + \alpha_b + \alpha_c \leq 7,5 < c(\{a, b, c\}).$$

Logo se α for estável ele não pode ser orçamento balanceado e portanto o núcleo deste jogo é vazio.

Dado um jogo de localização de instalações podemos determinar se o núcleo é ou não vazio utilizando a técnica do Teorema 6.3.2. Podemos resolver o modelo primal utilizando o método da separação (veja I.6 de [WN14,]), porém isto não poderá ser feito em tempo polinomial a menos que $P = NP$, visto que achar $c(S)$ para uma coalizão S significa resolver o problema da localização de instalações no grafo induzido por S , que é um problema NP-difícil.

6.5 Núcleo aproximado

Vimos na seção anterior que nem sempre existe um vetor no núcleo do jogo de localização de instalações. Também vimos que checar se o núcleo é ou não vazio é um problema intratável. Podemos relaxar a definição de núcleo buscando encontrar uma função de pagamento que satisfaça as condições de núcleo de forma relaxada.

Definição. Considere um jogo cooperativo (A, c) onde c mapeia o custo de um bem para cada possível coalizão $S \subseteq A$. Seja $\alpha \in \mathbb{R}^{|A|}$ um vetor onde a posição j contém o valor α_j que deverá ser pago pelo jogador j na construção do bem de custo $c(A)$. Dizemos que um vetor $\alpha \in \mathbb{R}^{|A|}$ pertence ao *núcleo γ -aproximado* se satisfaz as seguintes propriedades:

- *orçamento γ -balanceado:* $\gamma c(A) \leq \sum_{j \in A} \alpha_j \leq c(A)$
- *estabilidade:* para todo $S \subseteq A$ deve valer que $\sum_{j \in S} \alpha_j \leq c(S)$.

O pagamento sendo γ -balanceado garante que pelo menos uma fração γ do custo total será recuperado.

Exemplo 6.5.1. No exemplo da Figura 6.2 temos que $\alpha = (3.5, 2.5, 1.5)$ pertence ao núcleo $\frac{7.5}{8}$ -aproximado. Além do mais para este mesmo exemplo o núcleo γ -aproximado é sempre vazio para $\gamma > \frac{7.5}{8}$ pois o pagamento máximo para os clientes deve satisfazer $\alpha_a + \alpha_b + \alpha_c \leq 7.5$, enquanto que $c(A) = 8$. Portanto qualquer γ que leve a um núcleo aproximado não vazio deve satisfazer

$$8\gamma \leq \max_j \sum_j \alpha_j = 7,5.$$

Vimos na seção anterior (Teorema 6.3.2) como decidir se o núcleo de um jogo de compartilhamento de custos é ou não vazio. Este resultado pode ser adaptado para o caso de núcleo γ -aproximado.

Teorema 6.5.2. Um jogo cooperativo UT (A, c) possui um núcleo γ -aproximado não vazio se e somente se para toda coleção de pesos balanceados λ for válido que $\sum_{S \subseteq A} \lambda_S \cdot c(S) \geq \gamma c(A)$.

Demonstração. Pela definição, um jogo (A, c) possui núcleo γ -aproximado não vazio se e somente se a solução do programa linear abaixo tem valor maior ou igual a $\gamma c(A)$. Denotamos este programa linear por (PL1).

$$\begin{aligned} \text{(PL1)} \quad & \max \sum_{j \in A} \alpha_j \\ & \text{sujeito a } \sum_{j \in S} \alpha_j \leq c(S) \quad \forall S \subseteq A. \end{aligned}$$

Pelo teorema forte da dualidade o valor ótimo deste programa linear é igual ao de seu dual, denotado por (PL2), que é apresentado abaixo:

$$\begin{aligned}
 \text{(PL2)} \quad & \min \sum_{S \subseteq A} \lambda_S \cdot c(S) \\
 & \text{sujeito a} \quad \sum_{S: j \in S} \lambda_S = 1 \quad \forall j \in A, \\
 & \lambda_S \geq 0 \quad \forall S \subseteq A.
 \end{aligned}$$

Como vimos anteriormente, cada solução viável do dual corresponde a uma coleção de pesos balanceados (o inverso também é verdadeiro).

Se o núcleo γ -aproximado é não vazio então o valor ótimo do primal é maior ou igual a $\gamma c(A)$. Portanto todas as suas soluções viáveis do dual terão valor maior ou igual a $\gamma c(A)$. Assim para cada coleção de pesos balanceados vale que $\sum_{S \subseteq A} \lambda_S \cdot c(S) \geq \gamma c(A)$.

Se toda solução viável do dual tiver valor maior ou igual a $\gamma c(A)$, então a solução de valor mínimo tem valor maior ou igual a $\gamma c(A)$. Portanto o valor ótimo do primal será pelo menos $\gamma c(A)$ e obtemos um pagamento γ -balanceado. Portanto o núcleo γ -aproximado é não vazio. \square

Isto dá origem a um problema interessante que é o de se determinar o maior valor de γ possível para o qual o núcleo γ -aproximado é não vazio. Em alguns casos é possível determinar este valor baseado no *gap* de integralidade do modelo (PL2).

Definição. Considere um problema de minimização, \mathcal{I} o conjunto de instâncias deste problema e uma instância $I \in \mathcal{I}$. Seja (PLI) um modelo de programação linear inteira para o problema. Denotamos por $\text{PLI}(I)$ o valor de uma solução ótima inteira de I e por $\text{PL}(I)$ o valor de uma solução ótima do modelo relaxado. Definimos o *gap* de integralidade do modelo como $\max_{I \in \mathcal{I}} \frac{\text{PLI}(I)}{\text{PL}(I)}$. No caso de um problema de maximização a fração é invertida.

Seja \mathcal{G} o conjunto de todas as instâncias de um determinado jogo de compartilhamento de custos. O maior valor de γ para o qual toda instância do jogo tem núcleo γ -aproximado não vazio, é precisamente dado por $\min_{(A,c) \in \mathcal{G}} \frac{\text{PLI}(A,c)}{c(A)}$, onde $\text{PLI}(A,c)$ é o valor da solução ótima do modelo (PL1) para a instância (A,c) . Note que $\text{PLI}(A,c)$ tem o mesmo valor

do que a solução do modelo dual (PL2), e portanto o maior valor de γ possível é dado por $\min_{(A,c) \in \mathcal{G}} \frac{\text{PL2}(A,c)}{c(A)}$.

Um caso interessante ocorre quando o valor do modelo (PL2) inteiro (quando as variáveis λ são inteiras) é sempre $c(A)$. Neste caso o maior valor possível para γ é exatamente o inverso do *gap* de integralidade do modelo (PL2).

Um exemplo deste caso é quando temos um problema de compartilhamento de custos (A, c) onde a função de custo é subaditiva (c é subaditiva se para quaisquer subconjuntos disjuntos $S_1, S_2 \subseteq A$, vale que $c(S_1 \cup S_2) \leq c(S_1) + c(S_2)$). Neste caso se considerarmos as variáveis λ inteiras no modelo (PL2), temos que o valor da solução ótima é igual a $c(A)$ ($\lambda_A = 1$ e $\lambda_S = 0$ para todo $S \subset A$). Pois qualquer outra solução inteira que contenha $\lambda_{S_1} = 1$ e $\lambda_{S_2} = 1$ pode ser modificada zerando-se estas variáveis e fazendo-se $\lambda_{S_1 \cup S_2} = 1$ sem piorar a solução. Isto pode ser repetido até se obter a solução acima.

Teorema 6.5.3. Considere um jogo qualquer de compartilhamento de custos com função de custo subaditiva. O maior valor possível para γ , tal que o núcleo γ -aproximado seja não vazio, é igual ao inverso do *gap* de integralidade do modelo (PL2).

Demonstração. O *gap* de integralidade do modelo (PL2) é definido como

$$\max_{(A,c) \in \mathcal{G}} \frac{\text{PLI2}(A,c)}{\text{PL2}(A,c)} = \max_{(A,c) \in \mathcal{G}} \frac{c(A)}{\text{PL2}(A,c)},$$

pois c é subaditiva.

O maior valor de γ para o núcleo aproximado não vazio é dado por

$$\min_{(A,c) \in \mathcal{G}} \frac{\text{PL2}(A,c)}{c(A)}$$

que é exatamente o inverso do *gap* de integralidade do modelo (PL2). \square

6.5.1 Núcleo aproximado e localização de instalações

Nesta seção determinamos um valor γ para que o núcleo γ -aproximado do jogo de localização de instalações seja não vazio. Vimos pelo Teorema 6.5.3 que o maior valor possível é o inverso do *gap* de integralidade do

modelo (PL2) se a função de custo for subaditiva (este é o caso para o problema da localização de instalações). Um resultado interessante é que este *gap* é exatamente o mesmo de um modelo de programação linear natural para o problema.

Considere o seguinte modelo de programação linear inteira para o problema da localização de instalações que denotamos por (PLN1):

$$\begin{aligned}
 (\text{PLN1}) \quad & \min \sum_{i \in F} f_i x_i + \sum_{i \in F} \sum_{j \in A} d_{ij} y_{ij} \\
 & \text{sujeito a} \quad \sum_{i \in F} y_{ij} \geq 1 \quad \forall j \in A, \\
 & \quad \quad \quad x_i - y_{ij} \geq 0 \quad \forall i \in F, \forall j \in A, \\
 & \quad \quad \quad x_i, y_{ij} \in \{0, 1\} \quad \forall i \in F, \forall j \in A,
 \end{aligned}$$

cujo dual do modelo, denotado por (PLN2), relaxado é

$$\begin{aligned}
 (\text{PLN2}) \quad & \max \sum_{j \in A} \alpha_j \\
 & \text{sujeito a} \quad \sum_{j \in A} \beta_{ij} \leq f_i \quad \forall i \in F, \\
 & \quad \quad \quad \alpha_j - d_{ij} \leq \beta_{ij} \quad \forall i \in F, \forall j \in A, \\
 & \quad \quad \quad \alpha_j, \beta_{ij} \geq 0.
 \end{aligned}$$

Proposição 6.5.4. A função de custo do problema de localização de instalações é subaditiva.

Com isto temos que o maior valor de γ para um núcleo não vazio no jogo de localização de instalações é igual ao *gap* de integralidade do modelo (PL2). Veremos que este *gap* por sua vez é igual ao *gap* de integralidade do modelo (PLN1).

Teorema 6.5.5. Os modelos (PL1) e (PLN2) são equivalentes no sentido de que qualquer solução de um modelo é uma solução para o outro modelo com mesmo valor de função objetivo.

Demonstração. Seja α uma solução para o modelo (PL1). Vamos mostrar que este mesmo vetor é uma solução para o modelo (PLN2), atribuindo

ainda o valor de cada variável $\beta_{ij} = \max(0, \alpha_j - d_{ij})$. Suponha por absurdo que tenhamos uma restrição

$$\sum_{j \in A} \beta_{ij} > f_i$$

sendo violada no modelo (PLN2) para alguma instalação i . Seja S o subconjunto de jogadores j para os quais $\beta_{ij} > 0$ na restrição violada. Com isso temos que

$$\sum_{j \in S} (\alpha_j - d_{ij}) = \sum_{j \in S} \beta_{ij} > f_i$$

ou seja

$$\sum_{j \in S} \alpha_j > f_i + \sum_{j \in S} d_{ij}.$$

Note porém que no lado direito desta inequação temos o custo de uma solução para S , onde abre-se a instalação i e conecta-se cada $j \in S$ com a instalação i . Como $c(S)$ representa o custo de uma solução ótima para S temos

$$\sum_{j \in S} \alpha_j > c(S)$$

o que é um absurdo pois α é solução válida para o modelo (PL1).

Agora considere α como uma solução válida para o modelo (PLN2). Seja $S \subseteq A$ e considere $c(S)$ uma solução ótima para S utilizando instalações i_1, \dots, i_q . Nesta solução denotamos por S_{i_x} , $x = 1, \dots, q$, como sendo o subconjunto de jogadores que estão conectados à instalação i_x . Para cada instalação i_x considere a soma do segundo grupo de restrições do modelo (PLN2) para todo $j \in S_{i_x}$:

$$\sum_{j \in S_{i_x}} (\alpha_j - d_{i_x j}) \leq \sum_{j \in S_{i_x}} \beta_{i_x j} \leq f_{i_x}.$$

O que implica que

$$\sum_{j \in S_{i_x}} \alpha_j \leq f_{i_x} + \sum_{j \in S_{i_x}} d_{i_x j}.$$

Somando estas desigualdades para cada i_x , $x = 1, \dots, q$ temos

$$\sum_{j \in S} \alpha_j \leq c(S)$$

e portanto α é solução válida para o modelo (PL1). \square

Corolário 6.5.6. O maior valor de γ possível para que o jogo de localização de instalações tenha núcleo γ -aproximado não vazio é igual ao inverso do gap de integralidade do modelo (PLN1) para o problema da localização de instalações.

Demonstração. Pelo Teorema 6.5.3 sabemos que o maior valor de γ é igual ao inverso do *gap* de integralidade do modelo (PL2). Sabemos que o valor da solução ótima inteira dos modelos (PL2) e (PLN1) é igual a $c(A)$ e que o valor ótimo da solução fracionária de ambos os modelos tem o mesmo valor, já que os seus modelos duais são equivalentes pelo Teorema 6.5.5. Portanto o *gap* de integralidade dos modelos (PL2) e (PLN1) são iguais. \square

Uma maneira de se determinar limitantes para o *gap* de integralidade do modelo (PLN1) é via resultados de algoritmos de aproximação que são baseados em programação linear. Seja $PLN1(I)$ o valor da solução ótima fracionária para uma instância I do problema e suponha que em uma prova da correteza de um algoritmo aproximado alguém tenha demonstrado que o algoritmo sempre constrói uma solução inteira com custo menor ou igual a $\delta \cdot PLN1(I)$. Isto demonstra que o *gap* de integralidade do modelo (PLN1) é no máximo δ . Outra forma similar é via algoritmos baseados na técnica primal-dual. Neste caso, em geral, algoritmos constroem uma solução viável para o modelo dual (PLN2) com valor $PLN2(I)$ e ao mesmo tempo uma solução primal cujo valor é menor ou igual a $\delta \cdot PLN2(I)$. Da mesma forma este valor de δ será um limitante para o *gap* de integralidade do modelo primal (PLN1).

Um resultado clássico em aproximação é a 3-aproximação primal-dual de Jain e Vazirani [JV01] para o problema da localização de instalações. Com este resultado temos então que o jogo de localização de instalações tem núcleo $1/3$ -aproximado não vazio para qualquer instância do jogo.

Conceitos de Computação

Neste apêndice, apresentamos alguns conceitos de Teoria da Computação que utilizamos no decorrer do livro.

A.1 Complexidade computacional

Quando medimos a complexidade computacional de algoritmos, a maneira padrão é defini-la em função do tamanho (número de bits) da instância. Assim, podemos comparar algoritmos, por exemplo, pelo tempo (número de passos executados) para resolver uma instância de um dado tamanho.

Os algoritmos de tempo polinomial apresentam na sua grande maioria um bom desempenho computacional e com isso são chamados de *algoritmos eficientes*. Porém, existem diversos problemas na literatura para os quais não se conhece algoritmos de tempo polinomial para resolvê-los.

Duas importantes classes de complexidade computacional se destacam: as classes P e NP. A classe P consiste dos problemas de decisão (problemas com resposta SIM ou NÃO) para os quais há algoritmo de tempo polinomial para resolvê-lo.

A classe NP consiste dos problemas de decisão para os quais a resposta SIM pode ser verificada por um algoritmo de tempo polinomial usando um certificado de tamanho polinomial. Claramente, $P \subseteq NP$, porém não se sabe se $P = NP$. De fato, a questão “ $P = NP?$ ” é um dos principais problemas em aberto da Ciência da Computação. Uma classe de problemas

muito importante para o entendimento e possivelmente da solução desta questão é a dos problemas NP-completos. Um problema NP-completo é um problema em NP com a propriedade de que se existir um algoritmo de tempo polinomial para este, então todos os problemas em NP também podem ser resolvidos em tempo polinomial. Ou seja, a existência de um algoritmo de tempo polinomial para um problema NP-completo implica que $P = NP$. Atualmente, são conhecidos apenas algoritmos de tempo exponencial para resolver problemas da classe NP-completo [Coo71, GJ79].

Para exemplificar o crescimento de funções exponenciais comparadas com funções polinomiais, considere um computador com velocidade de 1 Terahertz. A Tabela A.1 mostra os tempos obtidos para algumas funções polinomiais e exponenciais, onde os tempos são dados em segundos (seg), dias e séculos (séc).

$f(n)$	$n = 20$	$n = 60$	$n = 100$
n	$2,0 \times 10^{-11}$ seg	$6,0 \times 10^{-11}$ seg	$1,0 \times 10^{-10}$ seg
n^2	$4,0 \times 10^{-10}$ seg	$3,6 \times 10^{-9}$ seg	$1,0 \times 10^{-8}$ seg
n^3	$8,0 \times 10^{-9}$ seg	$2,2 \times 10^{-7}$ seg	$1,0 \times 10^{-6}$ seg
2^n	$1,0 \times 10^{-6}$ seg	13,3 dias	$4,0 \times 10^8$ séc
3^n	$3,5 \times 10^{-3}$ seg	$1,3 \times 10^7$ séc	$1,6 \times 10^{26}$ séc

Tabela A.1: Comparação de funções de tempo computacional.

Note que um algoritmo de tempo 2^n é impraticável para instâncias grandes. Uma alternativa para contornar esta dificuldade seria tentar utilizar computadores mais velozes para lidar com instâncias maiores. Porém, isso é ineficaz para algoritmos exponenciais. Por exemplo, considere dois computadores, sendo o segundo mil vezes mais rápido do que o primeiro. Se um algoritmo com tempo de execução n^2 conseguir resolver instâncias de tamanho N no primeiro computador atual, ele conseguirá resolver instâncias de tamanho $31,6N$ no segundo, no mesmo tempo. Por outro lado, se o algoritmo tiver tempo de execução 2^n , então ele só conseguirá executar instâncias de tamanho $N + 9,97$ (apenas uma constante a mais) no segundo computador, no mesmo tempo. Para mais detalhes, veja [GJ79, Miy03].

Depois que Cook [Coo71] provou que o problema SAT é NP-completo, vários outros foram provados serem NP-completos (veja [CLRS01, GJ79, ACG⁺99]). Apesar da classe NP ser de problemas de decisão, ela contempla vários problemas que são simplificações de problemas de busca e de

otimização combinatória.

Um problema é dito NP-difícil quando a existência de um algoritmo de tempo polinomial para tal problema implica na existência de um algoritmo de tempo polinomial para algum problema NP-completo. Na prática existem diversos problemas NP-difíceis, como problemas de escalonamento de tarefas, balanceamento de carga, projeto de redes de telecomunicações e circuitos VLSI, roteamento de veículos, empacotamento de objetos em *containers*, localização de centros distribuidores, alinhamento de DNA e proteínas, classificação de dados, etc [Vaz01, CCD⁺01]. Para mais detalhes sobre modelos de computação, classes de complexidade e otimização combinatória, veja [CLRS01, GJ79, Pap94].

Indicamos o livro de Papadimitriou [Pap94] para um aprofundamento nos fundamentos principais de Complexidade Computacional.

A.2 Algoritmos de aproximação

Como vários dos problemas de otimização discreta são NP-difíceis, se $P \neq NP$, então não é possível obter um algoritmo para um problema NP-difícil que seja capaz de encontrar, para qualquer instância, soluções ótimas em tempo polinomial [WS11]. Uma forma de contornar esse problema é utilizar um *Algoritmo de Aproximação* que apesar de encontrar uma solução não necessariamente ótima, tem uma garantia de que tal solução não tem valor maior (no caso de minimização) ou menor (no caso de maximização) do que um determinado fator do valor de uma solução ótima.

Definição. Considere um problema de otimização em que \mathcal{I} é o seu conjunto de instâncias e $OPT(I)$ indica o valor de uma solução ótima para uma instância $I \in \mathcal{I}$.

Dizemos que um algoritmo é uma α -aproximação para tal problema se

- o algoritmo consome tempo polinomial no tamanho da representação de I ,
- se $SOL(I)$ é o valor da solução encontrada pelo algoritmo para a instância $I \in \mathcal{I}$, então

$$\max \left\{ \frac{SOL(I)}{OPT(I)}, \frac{OPT(I)}{SOL(I)} \right\} \leq \alpha.$$

Se um algoritmo é uma α -aproximação para um problema de otimização, dizemos que α é a *razão de aproximação* de tal algoritmo.

Apresentamos a seguir um conceito bem importante relacionado à algoritmos de aproximação.

Definição. Considere um problema de otimização. Um *esquema de aproximação polinomial* (PTAS, *polynomial-time approximation scheme*) é uma família de algoritmos $\{A_\varepsilon\}$ onde existe um algoritmo para cada constante $\varepsilon > 0$ e A_ε é uma $(1 + \varepsilon)$ -aproximação para tal problema.

Um PTAS pode ser visto como um único algoritmo que recebe uma instância I e um $\varepsilon > 0$ com duas características. Em primeiro lugar, tal algoritmo executa em tempo polinomial no tamanho da representação de I (mas não necessariamente em $1/\varepsilon$) e em segundo lugar ele devolve uma solução com valor no máximo $1 + \varepsilon$ vezes o valor de uma solução ótima (no caso de um problema de minimização). Se o algoritmo for polinomial também em $1/\varepsilon$, então temos um *esquema de aproximação totalmente polinomial* (FPTAS, *fully polynomial-time approximation algorithm*).

Além de encontrar algoritmos de aproximação para problemas de otimização que são NP-difíceis, estamos interessados também em encontrar bons limitantes inferiores para a razão de aproximação de qualquer algoritmo para um determinado problema. Em geral, isso é feito através da categorização dos problemas em classes, assim como fazemos com problemas de decisão.

Em particular, a classe APX é formada pelos problemas de otimização para os quais existe um algoritmo de aproximação de razão constante e, assim como no caso da classe NP, podemos considerar também a completude de problemas dentro da classe APX. Em particular, um problema é APX-difícil se a existência de um PTAS para o mesmo implicaria que $P = NP$.

Outro caso interessante é quando um problema é fortemente NP-difícil, isto é, mesmo a versão do problema onde a entrada é dada de forma unária é NP-difícil. Nesse caso, tal problema não tem um FPTAS a não ser que $P = NP$.

Existem também casos onde a existência de um algoritmo de aproximação com razão constante para um determinado problema de otimização implica que $P = NP$. Podemos considerar até mesmo restrições mais fortes. Por vezes, descobrimos que se formos capazes de aproximar um problema por uma determinada razão α (não necessariamente constante), então concluiríamos que $P = NP$.

Os livros de Williamson e Shmoys [WS11], Vazirani [Vaz01] e Carvalho et al. [CCD⁺01] são excelentes introduções à área de algoritmos de aproximação, abordando em mais detalhes as definições e conceitos apresentados nessa seção.

A.3 Programação linear e inteira

Começamos essa seção abordando a *Programação Linear*, cujos conceitos serão utilizados posteriormente para introduzir Programação Inteira Mista.

Em um *programa linear* desejamos, dados um vetor $c \in \mathbb{Q}^n$, um vetor $b \in \mathbb{Q}^m$ e uma matriz $A \in \mathbb{Q}^{m \times n}$, encontrar $x \in \mathbb{Q}^n$ que

$$(P) \quad \max \sum_{j=1}^n c_j x_j$$

$$\text{sujeito a} \quad \sum_{j=1}^n A_{ij} x_j \leq b_i \quad \forall i \in \{1, \dots, m\},$$

$$x_j \geq 0 \quad \forall j \in \{1, \dots, n\}.$$

Podemos considerar também algumas variantes (que são equivalentes à definição acima), onde ao invés de maximizar desejamos minimizar ou então consideramos desigualdades onde o limitante é inferior (ao invés de superior como acima), restrições de igualdade e outras opções para os sinais das variáveis (não-positivas ou livres de sinal). De fato, podemos fazer uma combinação desses vários tipos de restrições e posteriormente encontrar um programa linear equivalente no formato considerado acima.

Um fato importante em relação à programação linear é que existem algoritmos polinomiais (como o Método dos Elipsóides [Kha79] e Métodos de Pontos Interiores [Kar84]) capazes de encontrar soluções ótimas para programas lineares. Assim, a programação linear tem um papel importante em aspectos teóricos da Otimização Combinatória sendo, em particular, muito utilizada no projeto de algoritmos de aproximação.

Um conceito muito importante para a teoria da Programação Linear é o programa dual. Considere o programa linear (P), que chamamos de *Programa Primal*, apresentado acima. O *Programa Dual*, representado

por (D), consiste em encontrar $y \in \mathbb{Q}^m$ que

$$(D) \quad \min \sum_{j=1}^m b_j y_j$$

$$\text{sujeito a} \quad \sum_{i=1}^m A_{ij} y_i \geq c_j \quad \forall j \in \{1, \dots, n\},$$

$$y_i \geq 0 \quad \forall i \in \{1, \dots, m\}.$$

Existem algumas relações interessantes entre o programa primal e o programa dual. Em primeiro lugar, cada variável do programa primal corresponde a uma restrição do programa dual e cada restrição do programa primal corresponde a uma variável do programa dual. Em segundo lugar, existe um resultado bastante interessante que relaciona soluções ótimas do primal com soluções ótimas do dual. Antes de enunciarmos tal teorema, precisamos de duas definições.

Definição. Seja (P) um programa linear. Dizemos que (P) é *inviável* se não existe solução viável para (P).

Definição. Dizemos que um programa linear (P) de maximização é *ilimitado* se, para todo real M , existe uma solução viável de (P) com valor maior do que M . De forma análoga, dizemos que um programa linear (P') de minimização é *ilimitado* se, para todo real M , existe uma solução viável de (P') com valor menor do que M .

Estamos prontos para apresentar o resultado de Gale, Kuhn e Tucker [GKT51] que relaciona o programa primal com o programa dual.

Teorema A.3.1 (Teorema Forte da Dualidade [GKT51]). Seja (P) um programa linear e seja (D) o seu programa dual. Temos que

- Se (P) tem uma solução ótima então (D) tem uma solução ótima e os valores de tais soluções são iguais.
- Se (P) é inviável, então (D) é inviável ou ilimitado.
- Se (P) é ilimitado, então (D) é inviável.

Outro resultado interessante envolvendo o programa linear primal e o programa linear dual é a condição de folgas complementares.

Teorema A.3.2. Seja (P) um programa linear e seja (D) o seu programa dual. Para toda solução ótima x^* de (P) existe uma solução ótima y^* de (D) tal que:

- Para $i \in \{1, \dots, m\}$, se $y_i = 0$, então $\sum_{j=1}^n A_{ij}x_j = b_i$.
- Para $j \in \{1, \dots, n\}$, se $x_j = 0$, então $\sum_{i=1}^m A_{ij}y_i = c_j$.

Na *Programação Linear Inteira*, temos um programa linear onde, algumas (ou até mesmo todas) as variáveis precisam assumir valores inteiros. Essa restrição nos dá um grande poder computacional em troca de uma maior dificuldade em resolver tais programas, já que a programação linear inteira é NP-difícil [GJ79].

Quando formulamos um determinado problema de otimização utilizando um programa linear inteiro, dizemos que tal programa é uma *Formulação PLI* para tal problema. Dado um PLI, podemos remover as restrições de integralidade para obter um programa linear. Chamamos esse programa linear de *Relaxação Linear* do PLI em questão. Tal conceito é muito útil para o projeto de algoritmos de aproximação já que a relaxação linear dá um limitante superior para o valor de uma solução ótima do PLI (no caso da maximização; para minimização trata-se de um limitante inferior).

O livro de Chvátal [Chv83] é uma excelente opção para um aprofundamento na área de programação linear, pois apresenta uma introdução a programação linear juntamente com os resultados essenciais da área.

Bibliografia

- [AAE05] B. Awerbuch, Y. Azar, e L. Epstein. The price of routing unsplitable flow. In *Proceedings of 37th Symposium on Theory of Computing*, p. 57–66, 2005.
- [ACG⁺99] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, e M. Protasi. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*. Springer, 1999.
- [ADK⁺08] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, e T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM Journal on Computing*, 38(4):1602–1623, 2008.
- [AFMZ04] G. Aggarwal, T. Feder, R. Motwani, e A. Zhu. Algorithms for multi-product pricing. In *Proceedings of the 31th International Colloquium on Automata, Languages and Programming*, p. 72–83, 2004.
- [BK07] P. Briest e P. Krysta. Buying cheap is expensive: hardness of non-parametric multi-product pricing. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, p. 716–725, 2007.

- [BM97] S. Bikhchandani e J. W. Mamer. Competitive equilibrium in an exchange economy with indivisibilities. *Journal of Economic Theory*, 74(2):385–413, 1997.
- [CCD⁺01] M. H. Carvalho, M. R. Cerioli, R. Dahab, P. Feofiloff, C. G. Fernandes, C. E. Ferreira, K. S. Guimarães, F. K. Miyazawa, J. C. Pina Jr., J. Soares, e Y. Wakabayashi. *Uma Introdução Sucinta a Algoritmos de Aproximação*. Editora do IMPA, Rio de Janeiro, 2001. M. R. Cerioli and P. Feofiloff and C. G. Fernandes and F. K. Miyazawa (editores).
- [CCKK12] P. Chalermsook, J. Chuzhoy, S. Kannan, e S. Khanna. Improved hardness results for profit maximization pricing problems with unlimited supply. In *Proceedings of the 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012*, p. 73–84, 2012.
- [Chv83] V. Chvátal. *Linear Programming*. W. H. Freeman, 1983.
- [CK05] G. Christodoulou e E. Koutsoupias. The price of anarchy of finite congestion games. In *Proceedings 37th ACM Symposium on Theory of Computing*, p. 67–73, 2005.
- [CKN04] G. Christodoulou, E. Koutsoupias, e A. Nanavati. Coordination mechanisms. In *Proceedings of the 31st International Colloquium On Automata, Languages, And Programming*, p. 345–357, 2004.
- [Cla71] E. H. Clarke. Multipart pricing of public goods. *Public Choice*, 11(1):17–33, 1971.
- [CLN13] P. Chalermsook, B. Laekhanukit, e D. Nanongkai. Independent set, induced matching, and pricing: connections and tight (subexponential time) approximation hardnesses. In *Proceedings of the 54th Annual IEEE Symposium Foundations of Computer Science*, p. 370–379, 2013.
- [CLRS01] T. H. Cormen, C. E. Leiserson, R. L. Rivest, e C. Stein. *Introduction to Algorithms*. The MIT Press, 2001.

- [Coo71] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd Annual ACM Symposium on the Theory of Computing*, p. 151–158, 1971.
- [CSM04] J. R. Correa, A. S. Schulz, e N. E. S. Moses. Selfish routing in capacitated networks. *Mathematics of Operations Research*, 29:961–976, 2004.
- [Cur13] R. L. Curi. Jogos de Roteamento. Master’s thesis, Instituto de Computação – UNICAMP, 2013.
- [CV02] A. Czumaj e B. Vöcking. Tight bounds for worst-case equilibria. In *Proceedings of the 13th annual ACM-SIAM symposium on Discrete algorithms*, p. 413–420, 2002.
- [Dij59] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271, 1959.
- [Dre95] Z. Drezner. *Facility location: a survey of applications and methods*. Springer Verlag, 1995.
- [EDKM07] E. Even-Dar, A. Kesselman, e Y. Mansour. Convergence time to nash equilibrium in load balancing. *ACM Transactions on Algorithms*, 3(3):Article 32, 2007.
- [FH79] G. Finn e E. Horowitz. A linear time approximation algorithm for multiprocessor scheduling. *BIT Numerical Mathematics*, 19(3):312–320, 1979.
- [FKK⁺09] D. Fotakis, S. C. Kontogiannis, E. Koutsoupias, M. Mavronicolas, e P. G. Spirakis. The structure and complexity of nash equilibria for a selfish routing game. *Theoretical Computer Science*, 410:3305–3326, 2009.
- [FS14] C. G. Fernandes e R. C. S. Schouery. Approximation algorithms for the max-buying problem with limited supply. In *Proceedings of the 11th Latin American Theoretical Informatics Symposium*, p. 707–718, 2014.
- [GHK⁺05] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, e F. McSherry. On profit-maximizing envy-free

- pricing. In *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, p. 1164–1173, 2005.
- [GJ79] M. R. Garey e D. S. Johnson. *Computers and Intractability: a Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
- [GK05] V. Guruswami e S. Khot. Hardness of Max 3SAT with no mixed clauses. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, p. 154–162, 2005.
- [GKT51] D. Gale, H. W. Kuhn, e A. W. Tucker. Linear programming and the theory of games. In *Activity Analysis of Production and Allocation*, p. 317–329, 1951.
- [GMV05] M.X. Goemans, V.S. Mirrokni, e A. Vetta. Sink equilibria and convergence. In *Proceedings 46th Symposium on Foundations of Computer Science*, p. 142–151, 2005.
- [Gra66] R. L. Graham. Bounds for certain multiprocessor anomalies. *Bell System Technical Journal*, 45:1563–1581, 1966.
- [Gro73] T. Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973.
- [HS01] J. Hershberger e S. Suri. Vickrey pricing in network routing: Fast payment computation. In *Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science*, p. 252–259, 2001.
- [JV01] K. Jain e V. V. Vazirani. Approximation algorithms for metric facility location and k-median problems using the primal-dual schema and lagrangian relaxation. *Journal of the ACM*, 48(2):274–296, 2001.
- [Kar84] N. Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.
- [Kha79] L. G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979.

- [Kou03] E. Koutsoupias. Selfish task allocation. *Bulletin of EATCS*, 81:79–88, 2003.
- [Kov10] A. Kovács. New approximation bounds for LPT scheduling. *Algorithmica*, 57(2):413–433, 2010.
- [Kow96] T. Kowaltowski. Von Neumann: suas contribuições à computação. *Estudos Avançados*, 26(26):237–260, 1996.
- [KP99] E. Koutsoupias e C. H. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Symposium on Theoretical Aspects of Computer Science*, p. 404–413, 1999.
- [Kri09] V. Krishna. *Auction Theory*. Academic Press, segunda ed., 2009.
- [KSW97] D. Karger, C. Stein, e J. Wein. Scheduling algorithms. In M. J. Atallah, editor, *Handbook of Algorithms and Theory of Computation*. CRC Press, 1997.
- [LP86] L. Lovász e M. D. Plummer. *Matching Theory*. North-Holland, 1986.
- [Miy03] F. K. Miyazawa. *XI Escola Regional de Informática*, capítulo Programação Inteira, p. 49–90. SBC–Paraná, 2003.
- [Miy10] F. K. Miyazawa. *XXIX Jornada de Atualizações em Informática da SBC*, capítulo Uma introdução à Teoria de Jogos Algorítmica, p. 365–417. Sociedade Brasileira de Computação, 2010.
- [MS96] D. Monderer e L. Shapley. Potential games. *Games and Economic Behavior*, 14:124–143, 1996.
- [Nas50] J. F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.
- [Nob15] Nobelprize.org. All prizes in economic sciences. http://www.nobelprize.org/nobel_prizes/economics/laureates/, Maio 2015.

- [NR99] N. Nisan e A. Ronen. Algorithmic mechanism design. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, p. 129–140, 1999.
- [NRTV07] N. Nisan, T. Roughgarden, E. Tardos, e V. V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.
- [OR94] M. J. Osborne e A. Rubinstein. *A course in game theory*. MIT press, 1994.
- [ORS93] A. Orda, R. Rom, e N. Shimkin. Competitive routing in multiuser communication networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.
- [Pap94] C. H. Papadimitriou. *Computational complexity*. Addison-Wesley, 1994.
- [Pig20] A. C. Pigou. *The Economics of Welfare*. Macmillan, 1920.
- [Ras81] S. J. Rassenti. *0-1 Decision Problems with Multiple Resource Constraints: Algorithms and Applications*. PhD thesis, University of Arizona, 1981.
- [RHP98] M. H. Rothkopf, R. M. Harstad, e A. Pekec. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [Ros73] R.W. Rosenthal. A class of games possessing pure-strategy nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- [Rou03] T. Roughgarden. The price of anarchy is independent of the network topology. *Journal Of Computer And System Sciences*, 67:341–364, 2003.
- [Rou08] T. Roughgarden. An algorithmic game theory primer. In *Proceedings of the 5th IFIP International Conference on Theoretical Computer Science*, 2008.
- [RRG06] P. Rusmevichientong, B. V. Roy, e P. W. Glynn. A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98, 2006.

- [RSB82] S. J. Rassenti, V. L. Smith, e R. L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982.
- [RT02] T. Roughgarden e E. Tardos. How bad is selfish routing? *Journal of the ACM*, 49:236–259, 2002.
- [San02] T. Sandholm. Algorithm for optimal winner determination in combinatorial auctions. *Artificial Intelligence*, 135(1–2):1–54, 2002.
- [Sch14] R. C. S. Schouery. *Problemas de alocação e precificação de itens*. PhD thesis, Universidade de São Paulo, 2014.
- [Sha52] L. S. Shapley. A value for n-person games. Technical report, DTIC Document, 1952.
- [Sha67] L. S. Shapley. On balanced sets and cores. *Naval research logistics quarterly*, 14(4):453–460, 1967.
- [SLB08] Y. Shoham e K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2008.
- [Vaz01] V. V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.
- [Vic61] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of Finance*, 16(1):8–37, 1961.
- [vN45] John von Neumann. First draft of a report on the EDVAC, 1945.
- [vNM44] J. von Neumann e O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University press, primeira ed., 1944.
- [War52] J. Wardrop. Some theoretical aspects of road traffic research. *ICE Proceedings: Engineering Divisions*, 1(3):325–362, 1952.
- [WN14] L. A. Wolsey e G. L. Nemhauser. *Integer and combinatorial optimization*. John Wiley & Sons, 2014.

- [WS11] D. P. Williamson e D. B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.