Problemas de alocação e precificação de itens

Rafael Crivellari Saliba Schouery

Tese apresentada
ao
Instituto de Matemática e Estatística
da
Universidade de São Paulo
para
obtenção do título
de
Doutor em Ciências

Programa: Ciência da Computação
Orientadora: Profa. Dra. Cristina Gomes Fernandes

Durante o desenvolvimento deste trabalho o autor
recebeu auxílio financeiro da FAPESP e do CNPq

São Paulo, janeiro de 2014

Problemas de alocação e precificação de itens

Esta tese contém as correções e alterações sugeridas
pela Comissão Julgadora durante a defesa realizada
por Rafael Crivellari Saliba Schouery em 14/02/2014.
O original encontra-se disponível no Instituto de
Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Profa. Dra. Cristina Gomes Fernandes (orientadora) - IME-USP
- Prof. Dr. Walter Figueiredo Mascarenhas - IME-USP
- Prof. Dr. Flávio Keidi Miyazawa - UNICAMP
- Prof. Dr. Eduardo Sany Laber - PUC-RIO
- Prof. Dr. Claudson Ferreira Bornstein - UFRJ

"DON'T PANIC"

(Douglas Adams)

Agradecimentos

Diversas pessoas merecem meus agradecimentos pois fizeram parte do processo desse doutorado, seja de forma acadêmica ou na esfera pessoal.

Minha esposa, Raquel, foi, em todos os momentos desse doutorado, meu porto seguro. Juntos transpusemos as dificuldades que passamos nesses cincos anos e juntos iremos sobrepor as dificuldades das próximas fases de nossas vidas. A Raquel me deu forças para seguir em frente, aguentou as minhas preocupações com o futuro e suportou diversos dias em que precisei trabalhar até tarde, até mesmo em finais de semana. Junto com ela, transformamos um simples apartamento em um lar e nos tornamos uma família, algo essencial para mim e para a conclusão desse doutorado.

Meu pai, Farid (ou Saliba, como era chamado por todos), que infelizmente faleceu antes mesmo de eu entrar na graduação, me deu uma base sólida. Ele se esforçou para que eu tivesse uma boa educação, não apenas acadêmica, mas também em termos de ética e princípios pessoais. Devo tudo o que sou hoje a esse homem que me acolheu e me recebeu como filho após a morte de minha mãe. Apesar da grande diferença de idade, criamos uma forte conexão que carregarei comigo por toda a minha vida. Curiosamente, meu pai era formado em economia e fico feliz de, mesmo que num contexto específico, ter abordado a sua área de formação nesse trabalho.

A Professora Cristina foi capaz de dar o direcionamento necessário para o nosso projeto. Vindo diretamente da graduação, eu era um rapaz imaturo em diversos aspectos. Por isso, imagino que por muitas vezes não tenha sido fácil trabalhar comigo, principalmente nos primeiros anos, onde eu ainda não compreendia diversos aspectos do trabalho de um pesquisador. Além disso, acredito que a qualidade da minha escrita tenha melhorado muito graças ao fato da Cris ser detalhista em todos os textos que produz.

Além disso, vários professores do Instituto de Matemática e Estatística ajudaram na minha formação acadêmica. O IME é feliz em ter diversos professores excelentes como o Coelho, o Arnaldo, o José Augusto e a Yoshiko, entre outros. Em particular, gostaria de mencionar o Professor Roberto Hirata Jr. (meu primeiro orientador de iniciação científica) que me mostrou que eu deveria me esforçar ao máximo e que também me incentivou a fazer o doutorado direto. Gostaria de agradecer também ao Professor Yoshiharu Kohayakawa. Além de ser um ótimo professor e pesquisador, ele me ajudou a tomar a decisão de fazer esse doutorado direto ao invés de entrar no programa de mestrado. Por fim, agradeço também ao Carlinhos, meu co-autor no trabalho do Capítulo 3 e que foi um dos melhores professores que tive durante a minha estadia no IME.

O mais interessante de se participar de um laboratório não é pelo equipamento ou pelo espaço em si, mas pelas pessoas que o compõem e com quem convivemos diariamente. Mesmo com cada um focado em seu próprio trabalho, a comunicação com outros pós-graduandos foi essencial para o meu desenvolvimento. Por isso, gostaria de agradecer ao Fábio, ao Guilherme, ao Roberto, ao Marcio, ao Tássio e a tantos outros que estiveram presentes no meu cotidiano no trabalho nesses anos. Em

particular, gostaria de agradecer ao Álvaro que foi meu co-autor no trabalho do Capítulo 3 e ao Alexandre, cuja tese de doutorado e algumas conversas serviram de inspiração para a criação de um dos algoritmos do Capítulo 2. Além disso, o Álvaro e o Alexandre sempre foram grandes amigos com quem pude conversar sobre diversos assuntos, desde questões acadêmicas até sobre questões mais filosóficas sobre o ensino e o mundo acadêmico.

Também existem vários outros amigos que tiveram um papel importante durante o doutorado. Durante esses anos tive dois grupos de RPG (*Role-Playing Game*) onde nos encontrávamos para jogar, conversar, comer pizza e aproveitar um pouco mais a vida. Essas pessoas tiveram um papel essencial para que eu me mantivesse feliz e motivado mesmo nos momentos mais difíceis dos últimos anos.

Acho válido mencionar também diversos softwares e ferramentas que me foram úteis durante o doutorado. Talvez isso não seja o usual em agradecimentos em teses de doutorado, mas a verdade é que vários dos softwares que utilizei ao longo desses cinco anos me economizaram tempo e esforço em diversas tarefas. Em primeiro lugar, gostaria de agradecer aos criadores e colaboradores do LaTeX, que me permitiu escrever todos os artigos e, inclusive, essa tese. Vale mencionar também o pacote TikZ, que foi utilizado para fazer todas as imagens dessa tese e dos artigos publicados (juntamente com o PGF para os gráficos). Dois softwares que utilizei alternadamente durante o doutorado foram os editores de texto Sublime Text e o Textmate, que são excelentes editores de texto para quem trabalha com LaTeX e com programação. Outros três softwares essenciais para esse doutorado foram o Dropbox que permitiu a minha colaboração com os meus co-autores, o Evernote que me permitiu armazenar os artigos estudados e o JabRef, que me permitiu manter as referências organizadas para serem usadas nos vários artigos.

Por fim, agradeço a Fundação de Amparo à Pesquisa do Estado de São Paulo e o Conselho Nacional de Desenvolvimento Científico e Tecnológico pelo financiamento da pesquisa realizada.

Resumo

Nessa tese consideramos problemas de alocação e precificação de itens, onde temos um conjunto de itens e um conjunto de compradores interessados em tais itens. Nosso objetivo é escolher uma alocação de itens a compradores juntamente com uma precificação para tais itens para maximizar o lucro obtido, considerando o valor máximo que um comprador está disposto a pagar por um determinado item. Em particular, focamos em três problemas: o Problema da Compra Máxima, o Problema da Precificação Livre de Inveja e o Leilão de Anúncios de Segundo Preço.

O Problema da Compra Máxima e o Problema da Precificação Livre de Inveja modelam o problema que empresas que vendem produtos ou serviços enfrentam na realidade, onde é necessário escolher corretamente os preços dos produtos ou serviços disponíveis para os clientes para obter um lucro interessante. Já o Leilão de Anúncios de Segundo Preço modela o problema enfrentado por empresas donas de ferramentas de busca que desejam vender espaço para anunciantes nos resultados das buscas dos usuários. Ambas as questões, tanto a precificação de produtos e serviços quanto a alocação de anunciantes em resultados de buscas, são de grande relevância econômica e, portanto, são interessantes de serem atacadas dos pontos de vista teórico e prático.

Nosso foco nesse trabalho é considerar algoritmos de aproximação e algoritmos de programação inteira mista para os problemas mencionados, apresentando novos resultados superiores àqueles conhecidos previamente na literatura, bem como determinar a complexidade computacional destes problemas ou de alguns de seus casos particulares de interesse.

Palavras-chave: Precificação, Leilão, Teoria dos Jogos Algorítmica, Otimização Combinatória, Algoritmo de Aproximação, Programação Inteira.

Abstract

In this thesis we consider allocation and pricing problems, where we have a set of items and a set of consumers interested in such items. Our objective is to choose an allocation of items to consumers, considering the maximum value a consumer is willing to pay in a specific item. In particular, we focus in three problems: the Max-Buying Problem, the Envy-Free Pricing Problem and the Second-Price Ad Auction.

The Max-Buying Problem and the Envy-Free Pricing Problem model a problem faced in reality by companies that sell products or services, where it is necessary to correctly choose the price of the products or services available to clients in order to obtain an interesting profit. The Second-Price Ad Auction models the problem faced by companies that own search engines and desire to sell space for advertisers in the search results of the users. Both questions, the pricing of items and services and the allocation of advertisers in search results are of great economical relevance and, for this, are interesting to be attacked from a theoretical and a practical perspective.

Our focus in this work is to consider approximation algorithms and mixed integer programming algorithms for the aforementioned problems, presenting new results superior than the previously known in the literature, as well as to determine the computational complexity of such problems or some of their interesting particular cases.

Keywords: Pricing, Auction, Algorithmic Game Theory, Combinatorial Optimization, Approximation Algorithm, Integer Programming.

Sumário

Lista de Figuras **xi**

Lista de Tabelas **xiii**

1 Introdução **1**

1.1 Teoria dos Jogos . . . 3
1.2 Teoria dos Leilões . . . 7
1.2.1 Leilões de um Único Item . . . 7
1.2.2 Leilões Combinatórios . . . 8
1.2.3 Teoria dos Jogos para Leilões . . . 9
1.3 Algoritmos de Aproximação . . . 10
1.3.1 Algoritmos Online . . . 11
1.4 Programação Linear . . . 12
1.4.1 Programação Inteira Mista . . . 14
1.5 Notações e Definições . . . 15
1.6 Resultados Obtidos e Organização do Texto . . . 16

2 O Problema da Compra Máxima **17**

2.1 Modelo e Notação . . . 19
2.2 Um Limitante Inferior para o Problema da Compra Máxima . . . 20
2.3 Um Algoritmo para Oferta Limitada . . . 22
2.4 Desaleatorização do StarRounding . . . 27
2.5 Um Algoritmo para Oferta Limitada e Escada de Preços . . . 31
2.6 O Problema da Compra Máxima Online . . . 34

3 Precificações Livres de Inveja **37**

3.1 Modelo e Notação . . . 38
3.2 Formulações MIP . . . 39
3.3 Geradores de Testes para Leilões de Demanda Unitária . . . 44
3.3.1 Modelo das Características . . . 44
3.3.2 Modelo da Vizinhança . . . 45
3.3.3 Modelo da Popularidade . . . 45
3.4 Resultados Empíricos . . . 46
3.5 A Dificuldade de Aproximar Preços Geométricos . . . 50

4 Leilões de Anúncios **53**
4.1 Modelo e Notação 54
4.2 Leilões de Anúncios de Segundo Preço com Lances Binários 57
4.3 Resultados de Complexidade para Mercados Competitivos 58
4.4 O Algoritmo ReverseMatching em Mercados Competitivos 61
4.5 Novos Resultados para Mercados Competitivos 62
4.6 A Análise do Marcar-e-Matar é Justa 66
4.7 Análise do Uso de Emparelhamentos Máximos 68
4.8 Emparelhamento de Segundo Preço Online 72

5 Conclusão **75**

A Formulações para o Problema proposto por Shioda et al. **77**

B Resultados para o Problema da Precificação Livre de Inveja **79**

Referências Bibliográficas **85**

Índice Remissivo **90**

Lista de Figuras

1.1 Figura ilustrando o Jogo de Congestionamento 5
1.2 Exemplo das convenções utilizadas nas figuras 15

2.1 Dois exemplos de alocação e precificação para o Problema da Compra Máxima . . 20
2.2 Uma solução para o Problema da Compra Máxima com preço baixo 23
2.3 A execução do algoritmo StarRounding 25
2.4 Representação de uma alocação para um comprador no Problema da Compra Máxima Limitada EP-(α, t) 31

3.1 Uma alocação livre de inveja 39
3.2 A construção do grafo do Modelo das Características 45
3.3 A construção do grafo do Modelo da Vizinhança 46
3.4 A construção do grafo do Modelo da Popularidade 47
3.5 Número de soluções encontradas para o Modelo das Características. 48
3.6 Número de soluções encontradas para o Modelo da Vizinhança. 48
3.7 Número de soluções encontradas para o Modelo da Popularidade. 48
3.8 Gap final médio para instâncias grandes para o Modelo das Características. 49
3.9 Gap final médio para instâncias grandes para o Modelo da Vizinhança. 49
3.10 Gap final médio para instâncias grandes para o Modelo da Popularidade. 49

4.1 Um exemplo do Leilão de Anúncios de Segundo Preço 55
4.2 Um exemplo de O-emparelhamento de segundo preço 56
4.3 Um exemplo da construção de G_O^M 57
4.4 A construção de uma instância $f(H)$ 58
4.5 A construção de uma instância $f_\delta(H)$ 59
4.6 Representação de uma aresta para cima e uma aresta para baixo 61
4.7 O grafo G da prova do Lema 4.4.1 62
4.8 Simulação do Algoritmo Marcar-e-Matar 64
4.9 A construção de uma instância que mostra um limitante inferior da razão de aproximação do algoritmo Marcar-e-Matar 67
4.10 Heurística de atribuição de pesos para instâncias do 2PM. 68
4.11 Comparação entre os diversos algoritmos para o 2PM 70
4.12 O grafo da prova do Teorema 4.7.2 71
4.13 Uma instância onde um emparelhamento de custo máximo não é um bom emparelhamento de segundo preço 72

B.1 Gap final médio dos três modelos para instâncias pequenas que não foram resolvidas dentro do limite de tempo. 80
B.2 Tempo médio (em segundos) para resolver a relaxação linear do primeiro nó para instâncias grandes para os três modelos. 83

Lista de Tabelas

1.1 Representação do Jogo Dilema do Prisioneiro 4
1.2 Representação do Jogo de Congestionamento 5
1.3 Representação do Jogo Casando Centavos 6

4.1 Resultados para a execução dos algoritmo para o 2PM no modelo $D_{n,3}$ 69
4.2 Resultados para a execução dos algoritmo para o 2PM no modelo $D_{n,10}$ 70

B.1 Número de soluções ótimas encontradas para instâncias pequenas em cada um dos modelos. 79
B.2 Gap final médio dos três modelos para instâncias pequenas que não foram resolvidas dentro do tempo limite 81
B.3 Gap final médio para dos três modelos para instâncias grandes 82
B.4 Tempo médio (em segundos) para resolver a relaxação linear do primeiro nó para instâncias grandes (para os três modelos) 84

Capítulo 1

Introdução

Com a adoção da Internet em larga escala, 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, 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 Algorítmica difere da Teoria dos Jogos e de outros conceitos de microeconomia clássica em vários aspectos. 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 normalmente modela aplicações através de problemas de otimização concretos, buscando soluções ótimas, resultados de impossibilidade, limitantes inferiores e superiores em aproximações e assim por diante. Além disso, a Teoria dos Jogos Algorítmica normalmente considera questões de complexidade computacional como restrições na viabilidade do comportamento dos participantes [Rou08].

Segundo Christos Papadimitriou [NRTV07, *Foreword*], a Teoria dos Jogos foi fundada por John von Neumann e Oskar Morgenstern em 1944 em sua publicação intitulada *Theory of Games and Economic Behavior* [vNM44]. 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, considerar que a Teoria dos Jogos Algorítmica é a conjunção das teorias iniciadas por von Neumann na década de 1940.

Dentro da Teoria dos Jogos Algorítmica, um dos assuntos de interesse é o estudo de leilões e, em particular, o estudo de leilões combinatórios. Um leilão combinatório pode ser visto como uma forma de distribuir vários itens indivisíveis e correlacionados a vários consumidores [NRTV07, Cap. 11].

Um exemplo prático de leilão combinatório aparece em sites de busca. Quando uma busca é realizada em um site como o Google ou o Bing, o resultado apresentado consiste em dois tipos de itens: os links patrocinados e os não patrocinados. Os links não patrocinados são o resultado da busca propriamente dita, enquanto que os links patrocinados foram escolhidos do conjunto de clientes do site, dispostos a pagar pela exibição de seu link junto com o resultado de uma busca no site, dependendo das palavras apresentadas na busca.

A escolha dos clientes, aqui chamados de anunciantes, cujo link é apresentado junto com o resultado de uma busca é em geral feita através de um leilão. O resultado do leilão consiste não apenas da lista de anunciantes escolhidos para serem exibidos mas também, em geral, dos valores que cada anunciante pagará pela exibição de seu link e as condições de pagamento.

Cada anunciante, ao se tornar um cliente da empresa dona do site de busca, disponibiliza para a empresa uma série de informações que serão usadas na determinação do resultado do leilão. Por exemplo, um anunciante apresenta a lista de palavras em que tem interesse, ou seja, palavras que, se buscadas, interessaria a este anunciante ter seu anúncio exibido para o usuário do site.

Quando um usuário realiza uma busca, vários anunciantes desejam que sua propaganda (re-

lacionada ao tema de tal busca) seja exibida para o usuário. A ferramenta deve, de uma forma algorítmica, escolher alguns desses anunciantes para mostrar numa determinada busca e também determinar quanto cada anunciante irá pagar.

Os anúncios são mostrados em espaços reservados na página de resultados da busca (normalmente acima e à direita dos resultados) que são chamados de *slots*. É possível perceber, empiricamente, que o posicionamento do slot influencia na quantidade de cliques que o mesmo recebe. Ou seja, alguns slots são mais eficazes que outros.

Cada um desses anunciantes tem uma estimativa de quanto ele espera arrecadar uma vez que um usuário clique em seu anúncio. Esse valor indica quanto vale para o anunciante a visita de um usuário ao seu site. Tal valor é uma informação privada do anunciante, isto é, outros anunciantes ou mesmo o site não têm acesso à tal informação.

Por outro lado, o lance de um anunciante é o valor que ele declara para o site para participar da disputa pelos slots e, potencialmente, é diferente do seu real valor para o clique no anúncio. Isso ocorre pois o anunciante procura estratégias de fazer lances e pagar o mínimo possível para que o seu anúncio apareça. Isto é, o anunciante procura maximizar a diferença entre o seu ganho estimado pela exibição do anúncio e o preço que irá pagar por tal exibição. Vale notar que o preço que um anunciante paga depende do leilão realizado, e não é necessariamente o valor do seu lance (mas é, normalmente, não superior ao lance[1]).

Esse comportamento estratégico que ocorre entre os anunciantes faz com que resultados e técnicas da Teoria dos Jogos e da Teoria de Leilões sejam interessantes para a determinação dos anunciantes exibidos nas buscas.

As empresas donas das ferramentas de buscas desejam obter o maior lucro possível, mas consideram importante oferecer algumas garantias de justiça para os seus anunciantes, de forma que o processo seja rentável também para eles (assim os anunciantes não buscam outras alternativas de propaganda). Um exemplo disso é cobrar apenas se um clique for realizado no anúncio e não pela simples exibição do anúncio, método chamado de CPC (custo por clique), diferente do método CPM (custo por mil) onde o anunciante paga pela exibição do anúncio, independente de receber cliques ou não.

Dessa forma, esse problema, juntamente com vários outros da Teoria de Leilões, tem características de problemas de otimização multiobjetivo, pois além de maximizar o lucro da empresa é necessário considerar também que cada anunciante quer minimizar o quanto paga para ser anunciado.

Existem ainda outros fatores que podem ser considerados na modelagem de tal problema. Uma grande quantidade de buscas são realizadas diariamente e, portanto, a empresa não quer maximizar o seu lucro em relação a uma única busca, mas sim nas buscas como um todo. Ademais, os anunciantes têm orçamentos limitados que impedem que eles sejam anunciados muitas vezes num mesmo dia. Por fim, a empresa pode querer adicionar outras restrições ao problema para procurar melhorar o seu lucro ou obter um leilão mais justo ou mais interessante para os seus anunciantes.

Formalizando o problema acima, temos que os anunciantes são os compradores em um leilão, os slots são os itens a serem vendidos e a empresa faz o papel do leiloeiro, isto é, a entidade que tem os itens e deseja vendê-los.

Atualmente outros tipos de leilão são feitos eletronicamente como por exemplo pregões eletrônicos para a contratação de serviços, para realizar a venda de espectros de banda eletromagnética e vendas de produtos em sites. Os objetivos do leiloeiro podem ser diferentes do descrito acima. Por exemplo, o leiloeiro pode estar interessado em maximizar o bem estar social em vez de maximizar o seu lucro. Por trás de tais leilões, sempre há um mecanismo que descreve as regras do leilão, isto é, como são determinados os vencedores e o preço que os vencedores irão pagar.

Na verdade, várias formas de vender itens a compradores são chamadas de leilão. Todas elas têm alguns aspectos em comum. Segundo Vijay Krishna [Kri09], um leilão utiliza informações em forma de lances, feitos por compradores em potencial, e que representam o valor que o comprador

[1] Um leilão onde pode ocorrer de um anunciante pagar mais do que o seu lance é considerado "injusto" pela Teoria de Leilões, já que poderia ser mais vantajoso para o comprador não participar do leilão.

está disposto a pagar. O resultado de um leilão, isto é, quais compradores recebem quais itens e quanto cada comprador paga, é determinado apenas a partir da informação recebida através dos lances. Tal condição implica que leilões podem ser usados para vender qualquer tipo de item[2], independente de seu valor monetário, e são anônimos, de forma que distinguimos entre os compradores apenas através de seus lances. No caso de anúncios em sites de buscas, algumas empresas consideram a identidade dos compradores [AGM06] para estimar qual será o valor esperado que um comprador terá ao receber um clique em seu anúncio. Em tais leilões, o leiloeiro faz distinção entre os compradores não só pelos seus lances mas também por um peso definido pelo próprio leiloeiro e, portanto, tais leilões não são anônimos.

Por exemplo, na venda de slots para anunciantes em uma busca, usualmente é irrelevante quem são os anunciantes, qual a busca realizada e qual é a ordem de grandeza dos lances realizados. Para determinar os anunciantes mostrados é necessário apenas saber quais são os seus lances para tal busca.

Na literatura é possível encontrar uma grande variedade de leilões, cada qual com objetivos e restrições diferentes. Nesse trabalho abordaremos alguns desses leilões com um enfoque computacional. Apresentaremos alguns modelos para leilões e os resultados relevantes nos capítulos seguintes. Primeiramente, introduziremos algumas noções básicas sobre a Teoria dos Jogos e a Teoria de Leilões.

1.1 Teoria dos Jogos

A Teoria dos Jogos é uma área de grande importância para a teoria econômica, como pode ser visto pelos Prêmios Nobel de Ciências Econômicas recebidos por pesquisadores da área pelas suas contribuições acadêmicas: Harsanyi, Nash e Selten (1994), Aumann e Schelling (2005), Hurwicz, Maskin e Myerson (2007) e, recentemente, Roth e Shapley (2012) [Nob13].

Nessa seção, abordamos a Teoria dos Jogos Não-Cooperativos que estuda o comportamento de agentes em situações onde a escolha ótima do agente pode depender das escolhas de seus oponentes [FT91]. No que se segue, apresentamos uma breve introdução baseada no livro *Algorithmic Game Theory* [NRTV07, Cap. 1].

Um exemplo clássico da Teoria dos Jogos (Não-Cooperativos) é o **Dilema do Prisioneiro**, que pode ser apresentado pela seguinte alegoria. Dois suspeitos foram apreendidos pela polícia numa cena de um crime. A polícia não tem provas suficientes para condenar os dois suspeitos pelo crime principal mas pode condená-los por crimes menores. Os suspeitos são levados cada um para uma sala diferente e, com o objetivo de convencer os dois suspeitos a confessarem o crime principal, 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 veremos a seguir, se ambos os suspeitos desejam 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

[2]Exceto em caso de diferenças mais estruturais como, por exemplo, um leilão que vende bens divisíveis pode não servir para a venda de bens indivisíveis.

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 $P = \{1, \ldots, n\}$ é o **conjunto de jogadores**, $\mathcal{S} = \{S_1, \ldots, S_n\}$ é a família de conjuntos de estratégias e $\mathcal{U} = \{u_1, \ldots, u_n\}$ é a família de funções de utilidade de S em $\mathbb{R}$ com $S = S_1 \times \cdots \times S_n$. Para cada jogador $i \in P$, S_i é o **conjunto de estratégias** de i e u_i é a **função de utilidade** de i.

A interpretação para tal modelo é que, para um jogador $i \in P$, S_i indica as possíveis estratégias que o jogador i pode jogar em tal jogo e se cada jogador $j \in P$ escolhe a estratégia $s_j \in S_j$, então $s = (s_1, \ldots, s_n) \in S$ é o **resultado do jogo** e u_i indica se i prefere o resultado s a um outro possível resultado $s' \in S$, 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. Em geral, assumimos que todos os jogadores do jogo são racionais.

Por vezes queremos considerar jogos onde os jogadores têm custos ao invés de utilidade e, portanto desejam minimizar o seu custo. Nesse caso, basta considerar que o custo é representado por uma utilidade negativa.

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 de 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 -4	-5 -1
N	-1 -5	-2 -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 e também jogos onde os jogadores não têm funções de utilidades mas sim relações de preferência sobre S. Como não iremos usar tais variações nesse texto, optamos por omiti-las da definição de jogo.

A seguir, iremos introduzir 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 três canais diferentes A, B e C. O canal A é um pouco mais rápido do que o canal B e C e, portanto, trás 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.

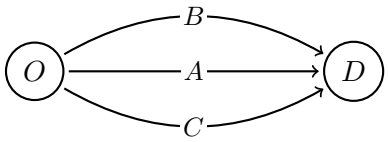


Figura 1.1: *Representação gráfica do Jogo de Congestionamento onde os jogadores* 1 *e* 2 *desejam enviar pacotes de* O *para* D *e têm como opção três caminhos:* A *(o mais curto) e* B *e* C *(um pouco mais longos do que* A *e de mesmo comprimento entre si).*

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

2 1	A	B	C
A	-5 -5	-2 -1	-2 -1
B	-1 -2	-6 -6	-2 -2
C	-1 -2	-2 -2	-6 -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 resultado* (A, B) *ou* (B, A) *em vez de* (A, A) *ou* (B, B).

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

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 $i \in P$, dizemos que uma estratégia $s'_i \in S_i$ é uma **melhor resposta** para $s_{-i} \in S_{-i}$ do que s_i se $u_i(s'_i, s_{-i}) > u_i(s_i, s_{-i})$.

Isto é, o jogador i prefere a estratégia s'_i à estratégia s_i quando os outros jogadores estão jogando s_{-i}. Note que, no caso do Dilema do Prisioneiro, temos que $u_1(C, C) > u_1(N, C)$ e portanto C é uma melhor resposta do que N para o jogador 1 quando o jogador 2 escolhe C. No Jogo do Congestionamento, $u_1(C, B) > u_1(B, B)$ e portanto C é uma melhor resposta do que B para o jogador 1 quando o jogador 2 escolhe B.

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

Como mencionamos anterior, no Jogo do Congestionamento, C é uma melhor resposta do que B para o jogador 1 quando o jogador 2 escolhe B, isto é, $u_1(C, B) > u_1(B, B)$. Mas note que A também é uma melhor resposta nesse caso. Porém, $u_1(A, B) > u_1(C, B)$ e portanto A é de fato uma resposta ótima para B.

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 Jogo do Congestionamento, A é resposta ótima para B (e para C), mas não para A. Já 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. De tal forma, podemos prever que, em tal situação, ambos os jogadores optariam por confessar, ou seja, escolheriam a estratégia C. Mas note que no Jogo de Congestionamento, nenhum jogador tem uma estratégia dominante.

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 Equilibrio de Nash é uma situação onde nenhum jogador pode aumentar sua utilidade trocando de estratégia individualmente. O conceito de Equilibrio 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 Equilibrio na realidade, 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.

Já o Jogo de Congestionamento tem quatro Equilíbrios de Nash diferentes (a saber $(A, B), (A, C), (B, A)$ e (B, C)), portanto é difícil prever qual será o resultado do jogo 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 a obter que o (único) Equilíbrio de Nash seja ambos os prisioneiros confessarem. 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. Tal conceito é chamado de Projeto de Mecanismos.

Infelizmente, nem todo jogo tem um Equilíbrio de Nash. Considere, por exemplo, o jogo chamado **Casando Centavos** onde temos dois jogadores cada um com uma moeda e que precisam escolher cara ou coroa para sua moeda. O jogador 1 ganha se ambos os jogadores escolheram a mesma estratégia e o jogador 2 ganha se os jogadores escolheram estratégias diferente. A Tabela 1.3 representa formalmente o jogo.

2 1	Cara	Coroa
Cara	-1 1	1 -1
Coroa	1 -1	-1 1

Tabela 1.3: *Representação matricial do jogo Casando Centavos. O jogador* 1 *escolhe a linha e o jogador* 2 *escolhe a coluna.*

Podemos fazer uma extensão de 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. Nessa extensão as estratégias possíveis para um jogador são todas as distribuições de probabilidade sobre o conjunto de estratégias de tal jogador no jogo original. Em tal extensão, 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 $\mathcal{S}' = \{S'_1, \ldots, S'_n\}$ e $U' = \{u'_1, \ldots, u'_n\}$ tal que, para um jogador $i \in P$, S'_i é o espaço de distribuições de probabilidade sobre S_i e, para o vetor $\sigma \in S'$, onde $S' = S'_1 \times \cdots \times S'_n$ 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 Casando Centavos, os jogadores podem escolher entre cara ou coroa aleatoriamente.

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

Um jogo nem sempre tem um Equilíbrio de Nash, como é o caso do jogo Casando Centavos, 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 1.1.1 (Nash [Nas50])**.** Todo jogo finito com os conjuntos de estratégias finitos tem um Equilíbrio Misto de Nash.

1.2 Teoria dos Leilões

Os leilões são usados desde a antiguidade para a venda de diversos tipos de objetos. Uma grande variedade de itens são hoje vendidos através de leilões, incluindo objetos de arte e as mais variadas *commodities*. Eles também são utilizados para transferir bens públicos para empresas privadas e conceder direitos de utilização de recursos naturais de um governo. Além disso, vários leilões ocorrem na Internet nos chamados sites de leilões [Kri09].

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

1.2.1 Leilões 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].

Porém, existem outras 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.

No que segue, iremos considerar apenas leilões de carta fechada já que eles são mais fáceis de serem abordados teoricamente.

No **Leilão de Primeiro Preço**, um leiloeiro deseja vender um item a um conjunto de compradores. 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.

Cada comprador b tem um valor $v_b \in \mathbb{R}_+$ para tal item. Tal valor indica monetariamente o quanto o comprador está disposto a pagar por tal item. Além disso, tal valor é uma informação que apenas o comprador b conhece.

Portanto, se o comprador b receber o item e pagar um preço p, irá obter um lucro de $v_b - p$, já que v_b é o valor que b dá para o item. De ponto de vista da Teoria dos Jogos, é de se esperar que os compradores irão escolher estrategicamente o lance a ser feito.

Neste leilão, o leiloeiro gostaria de entregar o item para o comprador (ou um dos compradores) com maior v_i, independentemente do preço cobrado pelo item. Isto é, a motivação de tal leiloeiro não é maximizar seu lucro, mas sim maximizar o que chamamos de bem estar social[3].

Seja m o número de compradores e considere a função $p : \mathbb{R}_+^m \to \mathbb{R}_+$ tal que $p(s) = \max\{s_b : 1 \leq b \leq m\}$. Podemos então definir o jogo $(P, \mathcal{S}, \mathcal{U})$, onde $P = \{1, \ldots, m\}$, $S_b = \mathbb{R}_+$ e, para cada $s \in \mathbb{R}_+^m$, $u_b(s) = v_b - p(s)$ para algum comprador b tal que s_b é máximo em s (em caso de empate, escolhemos b de forma determinística) e $u_{b'}(s) = 0$ para todo $b' \neq b$. Isto

[3]Maximizar o bem-estar social é distribuir o bem para o comprador que dá maior valor (monetário) a tal item. Isso pode ser interessante em casos como a distribuição de itens do governo para a sociedade.

é, neste jogo, as estratégias de qualquer jogador são os números reais positivos (representando os lances) e a utilidade do jogador b é $v_b - p$ se b ganha o item e 0 caso contrário, onde p indica o maior lance feito por um jogador.

Note que, no leilão de primeiro preço, se o comprador b recebe o item e seu lance é $\ell_b = v_b$ então $u_b(\ell) = 0$, o mesmo lucro 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. Isto é, os compradores não podem obter um lucro menor ao submeter um lance estritamente menor do que v_i se comparado com o lance v_b. Dessa forma, há um estímulo para que o lance não reflita o real valor do comprador, e assim é difícil garantir que o comprador que ganha o item é o que dá maior valor a ele. Ou seja, o leiloeiro pode não estar maximizando o bem estar social como gostaria.

Já no **Leilão de Segundo Preço**[4], o preço cobrado do ganhador é o segundo maior lance. O grande interesse teórico desse leilão é que um comprador nunca obtém um lucro menor ao trocar de lance para indicar o seu real valor para o item, isto é, para um jogador b, dar v_b como lance é uma estratégia dominante. Por isso, dizemos que tal leilão é **compatível com incentivo** ou **à prova de estratégia**.

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 valor $\epsilon > 0$ e que os compradores estão ordenados de forma que $v_1 \geq v_2 \geq \ldots \geq v_m$. Note que, enquanto o lance atual é menor que $v_2 - \epsilon$ os compradores 1 e 2 continuaram dando lances. Quando o lance atual for maior ou igual a $v_2 - \epsilon$, o comprador 2 não terá mais interesse em dar um novo lance pois isso levaria a um lucro não positivo. Assim, ao final de tal leilão, o comprador 1 recebe o item pagando um preço entre v_2 e $v_2 + \epsilon$, um resultado muito próximo do resultado obtido com o Leilão de Segundo Preço.

1.2.2 Leilões Combinatórios

Quando desejamos vender vários itens para vários compradores, podemos realizar vários leilões de único item. Mas isso pode levar a um bem-estar social ou um lucro menor do que se considerássemos todos os itens em um único leilão.

Para resolver esse problema foram criados os leilões combinatórios. Segundo Smith [CSS06, *Foreword*], os leilões combinatórios foram introduzidos por Rassenti [Ras81] numa primeira tentativa de criar um mercado apoiado por computadores [RSB82]. Em tais leilões, *a priori*, um comprador pode ter valores diferentes para cada conjunto de itens.

Considere um conjunto finito de itens $I = \{1, \ldots, n\}$ e um conjunto finito de compradores $B = \{1, \ldots, m\}$. Em um leilão combinatório consideramos que, para cada comprador b, existe uma função $v_b : 2^I \to \mathbb{R}_+$ com $v_b(\emptyset) = 0$, 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 $\mathcal{V}$ o conjunto de valorações válidas para cada comprador[5]. Chamamos o vetor $(v_1, \ldots, v_m)$ simplesmente de **valoração**.

Usualmente, consideramos que, para todo comprador b, v_b é uma função crescente, isto é, $v_b(T) \geq v_b(S)$ para todo S e todo T tal que $S \subseteq T \subseteq I$. Essa restrição pode ser interpretada como a capacidade do comprador b de dispor dos itens do conjunto $T \setminus S$ sem custos e por tal motivo é chamada de **livre disposição**.

A valoração de um comprador é, normalmente, de conhecimento exclusivo do comprador. Por isso, quando desejamos fazer uma análise baseada na Teoria dos Jogos, consideramos também que cada comprador faz um **lance** para cada conjunto de itens. Tal lance representa a estratégia escolhida pelo comprador e é dada através de uma função ℓ_b de $\mathcal{V}$.

Quando o aspecto estratégico não é de interesse no assunto abordado, consideramos que, para todo comprador b, $\ell_b = v_b$, ou equivalentemente, as valorações são de conhecimento do leiloeiro (e

[4] Proposto pelo matemático e economista William Vickrey, ganhador do Prêmio Nobel de Ciências Econômicas em 1996 [Nob13]. Por tal motivo, esse leilão também é chamado de Leilão de Vickrey.

[5] Note que $\mathcal{V}$ pode ser uma restrição do conjunto de funções $v_b : 2^I \to \mathbb{R}_+$ com $v_b(\emptyset) = 0$ pois podemos desejar limitar as valorações válidas para os compradores.

possivelmente dos outros compradores).

Uma **alocação** de itens a compradores é uma família A de conjuntos de itens indexada por B tal que, se b_1 e b_2 pertencem a B, com $b_1 \neq b_2$, então $A_{b_1} \cap A_{b_2} = \emptyset$. Isto é, em uma alocação, cada comprador recebe um conjunto de itens (possivelmente vazio) e cada item é alocado a, no máximo, um comprador. Denotaremos por $\mathcal{A}$ o conjunto de alocações de itens a compradores.

Um **leilão combinatório** (também chamado de mecanismo) é um par (f, p) onde f é uma função de $\mathcal{V}^m$ em $\mathcal{A}$ e p é uma função de $\mathcal{V}^m$ em $\mathbb{R}_+^n$. A função f é chamada de **função de alocação** e define quais itens serão entregues a quais compradores. Já a função p é a **função de precificação**[6] e define o preço de cada item. Isto é, para lances $(\ell_1, \ldots, \ell_m) \in \mathcal{V}^m$, se $f(\ell_1, \ldots, \ell_m) = (A_1, \ldots, A_m)$ então o comprador b recebe os itens do conjunto A_b e se $p(\ell_1, \ldots, \ell_m) = (p_1, \ldots, p_n)$ então o preço do item i é p_i e o comprador b deverá pagar $\sum_{i \in A_b} p_i$.

Existem vários objetivos que podem ser considerados na criação de um leilão. Os mais comuns são maximizar o **bem-estar social**, definido como $\sum_{b \in B} v_b(A_b)$ e maximizar o **lucro do leiloeiro**, definido como $\sum_{b \in B} \sum_{i \in A_b} p_i$.

Através de restrições nas funções de alocação e de preço e também restrições nos lances e valorações dos compradores é possível definir leilões mais específicos como, por exemplo, leilões para a venda de bens digitais. A seguir apresentamos algumas dessas restrições.

Por vezes queremos que um item possa ter várias **cópias**, isto é, queremos que o mesmo item possa ser vendido a vários compradores mas sem diferenciar essas cópias entre si durante o leilão. Por isso, se i e j são cópias do mesmo item, então $p_i = p_j$. Quando temos infinitas cópias do mesmo item (ou cópias suficientes), dizemos que o leilão é de **oferta ilimitada**. Se há uma restrição no número de cópias, dizemos que o leilão é de **oferta limitada**.

Um conjunto de leilões com oferta ilimitada são os **Leilões de Bens Digitais**. Em tais leilões, temos apenas um item com custo marginal de produção zero (e por isso consideramos que existem infinitas cópias do mesmo). Dado os lances dos compradores, escolhemos um preço p para tal item e alocamos as cópias desse item apenas a compradores com lance maior ou igual a p.

Podemos também adicionar outras restrições, principalmente para lidar com o excesso de informação em relação as valorações e aos lances que podem ser funções com imagens de tamanho exponencial em $|I|$. Exemplos incluem considerar que as valorações são modulares, submodulares ou supermodulares.

Um exemplo de tal restrição útil em nosso trabalho é considerar que cada comprador b deseja receber no máximo um item, isto é, para todo conjunto S de itens com $|S| > 1$, temos $v_b(S) = \max\{v_b(\{i\}) : i \in S\}$. Chamamos tais leilões de **Leilões de Demanda Unitária**[7]. É fácil ver que, em tais leilões, o comprador tem interesse em apenas um dos itens do conjunto de itens atribuídos a ele e, por isso, podemos nos restringir a funções de alocação que alocam no máximo um item por comprador.

Os **Leilões de Anúncios** são leilões de demanda unitária usados para a venda de anúncios em sites de busca, Nesses leilões temos m itens diferentes (chamados de *slots* de anúncios). Apesar dos *slots* serem representados por itens diferentes, podemos pensar que eles são variações de qualidade de um mesmo item[8]. Por isso os compradores submetem apenas um lance para todos os *slots* e consideramos que o primeiro *slot* é melhor que o segundo e assim por diante.

1.2.3 Teoria dos Jogos para Leilões

Do ponto de vista da Teoria dos Jogos, podemos considerar que um leilão induz um jogo onde o conjunto de jogadores é B, o conjunto de estratégias de um jogador é $\mathcal{V}$ e que a utilidade de um

[6] Nesse trabalho, consideramos que um mecanismo define um preço para cada item. Outra possibilidade [NRTV07, Cap. 9] seria considerar que mecanismos definem quanto cada comprador deve pagar, isto é, p é uma função de $\mathcal{V}^m$ em $\mathbb{R}_+^m$. Apesar dessa definição alternativa ser mais geral, preferimos manter a coesão com os outros capítulos onde trabalharemos com precificação e portanto consideraremos que cada item recebe um determinado preço.

[7] Também chamados na literatura de Leilões Multi-Item ou Leilões Multi-Produto.

[8] No caso, a variação da qualidade depende do posicionamento do *slot* do anúncio na página de busca.

jogador b é $u_b(\ell) = v(f_b(\ell)) - \sum_{i \in f_b(\ell)} p_i(\ell)$, onde, para um resultado do jogo ℓ (ou seja, os lances dos jogadores), $f_b(\ell)$ é o conjunto de itens alocados a b e $p_i(\ell)$ é o preço do item i.

Dizemos que um leilão combinatório é **compatível com incentivo** (ou à prova de estratégia) se, para todo comprador b, escolher $\ell_b = v_b$ é uma estratégia dominante. Vickery, Clarke e Groves [Vic61, Cla71, Gro73] deram condições suficientes para que um leilão seja compatível com incentivo. Tais leilões são chamados de mecanismos VCG e maximizam o bem-estar social.

Infelizmente, em vários cenários, os mecanismos VCG podem apresentar anomalias como a diminuição do lucro com a entrada de um novo comprador ou então resultar em um lucro ruim para o leiloeiro (inclusive lucro zero no caso de não haver concorrência pelos itens no mercado) [CSS06, Cap. 1]. Por isso, por vezes, é necessário desenvolver outros mecanismos para problemas específicos.

No caso de Leilões de Anúncios, um mecanismo tornou-se bastante popular. O **Leilão de Segundo Preço Generalizado** (GSP) transpõe o conceito do Leilão de Segundo Preço (que é compatível com incentivo) para um leilão de anúncios da seguinte forma. O comprador com maior lance recebe o primeiro *slot* e paga o segundo maior lance, já o comprador com o segundo maior lance recebe o segundo *slot* e paga o terceiro maior lance e assim por diante.

Apesar de tal mecanismo não ser compatível com incentivo, ele é amplamente usado, trazendo grandes lucros para empresas que vendem espaço para anúncios. O GSP é utilizado principalmente pela sua facilidade de entendimento pelos compradores e pela facilidade de implementação, já que os mecanismos VCG são mais complexos.

Uma pergunta natural é quanto um mecanismo que não é compatível com incentivo pode ser prejudicado pelo comportamento estratégico dos compradores. Uma forma de analisar esse impacto é através do **Preço da Anarquia**. O Preço da Anarquia é definido como a razão entre o bem-estar social máximo e o menor bem-estar social obtido por um equilíbrio[9]. Dessa forma, o Preço da Anarquia pode ser utilizado para comparar dois mecanismos diferentes em busca de uma intuição de qual possa ser melhor na prática.

1.3 Algoritmos de Aproximação

Segundo Williamson e Shmoys [WS11], os problemas mais interessante de otimização discreta são NP-difíceis, isto é, para tais problemas não existem algoritmos que executem em tempo polinomial no tamanho da entrada (também chamados de algoritmos eficientes) a não ser que P = NP. 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 .

Dessa forma, podemos considerar três opções. A primeira é desenvolver um algoritmo capaz de encontrar soluções ótimas em tempo polinomial mas para apenas um subconjunto das instâncias. A segunda possibilidade, que inclusive discutiremos na Seção 1.4, é criar algoritmos capazes de encontrar soluções ótimas para qualquer instância mas sem exigir que isso seja feito em tempo polinomial (por exemplo, usando Programação Inteira Mista). Por fim, podemos focar em desenvolver algoritmos que executam em tempo polinomial para todas as instâncias mas que não encontram soluções necessariamente ótimas. Em particular, nessa seção estamos interessados nos chamados **Algoritmos de Aproximação** onde, apesar da solução encontrada não ser necessariamente ótima, temos uma garantia de que tal solução não tem valor maior (no caso de minimização) ou menor (no caso de maximização) de que um determinado fator do valor de uma solução ótima.

Definição. Considere um problema de otimização tal que $\mathcal{I}$ é o seu conjunto de instâncias e $\mathrm{OPT}(I)$ indica o valor de uma solução ótima para uma instância $I \in \mathcal{I}$. Seja $\alpha : \mathcal{I} \to \mathbb{R}_+$, dizemos que um algoritmo é uma **α-aproximação** para tal problema se

- O algoritmo consome tempo polinomial no tamanho da representação de I.

[9] Ou como a razão entre o lucro máximo e o menor lucro obtido por um equilíbrio quando estamos interessados no impacto do comportamento estratégico no lucro do leiloeiro.

- Se $\mathrm{SOL}(I)$ é o valor da solução encontrada pelo algoritmo para a instância $I \in \mathcal{I}$, então

$$\max\left\{\frac{\mathrm{SOL}(I)}{\mathrm{OPT}(I)}, \frac{\mathrm{OPT}(I)}{\mathrm{SOL}(I)}\right\} \leq \alpha(I).$$

Se um algoritmo é uma α-aproximação para um problema de otimização, dizemos que α é a **razão de aproximação** de tal algoritmo.

Existem autores que preferem utilizar uma razão de aproximação maior ou igual a 1 para problemas de minimização e uma razão de aproximação menor ou igual a 1 para problemas de maximização. Mas, pela nossa definição, $\alpha(I) \geq 1$ para todo $I \in \mathcal{I}$ tanto se o problema é de maximização quanto se é de minimização.

Note também que essa definição de razão de aproximação é bem geral, pois α é uma função de $\mathcal{I}$ em $\mathbb{R}_+$. Mas, normalmente, consideramos algo mais simples como α constante ou então α sendo uma função do tamanho da entrada. Por exemplo, considere um problema de minimização com conjunto de instâncias $\mathcal{I}$ e seja $\mathrm{OPT}(I)$ o valor de uma solução ótima para a instância I. Uma α-aproximação para esse problema é tal que, para todo $I \in \mathcal{I}$, $\mathrm{SOL}(I) \leq \alpha(I)\mathrm{OPT}(I)$. Porém, podemos considerar α como uma função de $\mathbb{N}$ em $\mathbb{R}_+$ tal que, para toda instância I de tamanho n, $\mathrm{SOL}(I) \leq \alpha(n)\mathrm{OPT}(I)$. Por outro lado, essa definição nos permite considerar também outras informações da instância além de seu tamanho.

Apresentamos a seguir um conceito bem importante em relação a 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 $\varepsilon > 0$ 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 é 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). Porém, 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 APX-difícil é tal que a existência de um PTAS para esse problema implicaria que $\mathrm{P} = \mathrm{NP}$.

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 $\mathrm{P} = \mathrm{NP}$.

Existem também casos onde a existência de um algoritmo de aproximação com razão constante para um determinado problema de otimização implica que $\mathrm{P} = \mathrm{NP}$. 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 $\mathrm{P} = \mathrm{NP}$.

Os livros de Williamson e Shmoys [WS11] e Vazirani [Vaz01] são excelentes introduções à área de algoritmos de aproximação, abordando em mais detalhes as definições e conceitos apresentados nessa seção.

1.3.1 Algoritmos Online

Por vezes consideramos **Problemas Online**, onde não temos toda a entrada disponível no começo da execução do algoritmo, com o restante da informação chegando aos poucos com o passar do

tempo, e precisamos tomar decisões (normalmente irreversíveis) conforme novas partes da entrada tornam-se disponíveis. As decisões tomadas impactam na qualidade final da solução encontrada e podem basear-se apenas nas informações conhecidas até o momento e nas decisões já tomadas. Chamamos os algoritmos projetados para lidar com tais problemas de **Algoritmos Online**.

Assim como no caso de algoritmos de aproximação, podemos comparar a solução encontrada por um algoritmo online com um algoritmo offline ótimo, isto é, um algoritmo que sabe toda a informação de antemão e toma as decisões que levam a uma solução final ótima. Para realizar tal comparação utilizamos um conceito similar ao conceito de razão de aproximação.

Definição. Considere um problema de otimização online tal que $\mathcal{I}$ é o seu conjunto de instâncias e $\mathrm{OPT}(I)$ indica o valor de uma solução encontrada por um algoritmo offline ótimo para uma instância $I \in \mathcal{I}$. Seja $\alpha : \mathcal{I} \to \mathbb{R}_+$, dizemos que um algoritmo é **α-competitivo** para tal problema se

$$\max\left\{\frac{\mathrm{SOL}(I)}{\mathrm{OPT}(I)}, \frac{\mathrm{OPT}(I)}{\mathrm{SOL}(I)}\right\} \leq \alpha(I),$$

onde $\mathrm{SOL}(I)$ é o valor da solução encontrada pelo algoritmo para a instância $I \in \mathcal{I}$. Se um algoritmo é α-competitivo para um problema de otimização online, dizemos que α é a **razão de competitividade** de tal algoritmo.

Recomendamos o livro de Borodin e El-Yaniv [BEY98] para um aprofundamento no assunto, já que ele é totalmente dedicado a problemas online e a análise de competitividade.

1.4 Programação Linear

Começamos essa seção abordando a **Programação Linear**, cujos conceitos serão utilizados posteriormente para introduzir o assunto da 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

$$\begin{array}{llrcll}
\text{(P)} & \text{maximize} & \displaystyle\sum_{j=1}^{n} c_j x_j & & & \\
& \text{sujeito a} & \displaystyle\sum_{j=1}^{n} A_{ij} x_j & \leq & b_i, & \forall i \in \{1,\ldots,m\} \\
& & x_j & \geq & 0, & \forall j \in \{1,\ldots,n\}.
\end{array}$$

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.

Dois fatos são muito importantes em relação à programação linear. O primeiro 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. O segundo é que, na prática, podemos resolver problemas lineares grandes rapidamente através de vários métodos. O próprio Simplex, o método mais conhecido e bastante usado para a resolução de programas lineares, apesar de não ser polinomial (no pior caso), é muito rápido na prática. Por esses dois fatores, a programação linear tem um importante papel tanto em aspectos teóricos da Otimização Combinatória (em particular, é muito utilizada no projeto de algoritmos de aproximação) quanto em aspectos práticos pois, além da sua importância para a solução de problemas contínuos, é a base da Programação Inteira Mista, como veremos a seguir.

Considere um vetor $x \in \mathbb{Q}^n$ e o programa linear (P) acima. Dizemos que x é uma **solução viável** de (P) se $\sum_{j=1}^{n} A_{ij}x_j \leq b_i$ para todo $i \in \{1,\ldots,m\}$ e $x_j \geq 0$ para todo $j \in \{1,\ldots,n\}$.

Dizemos que x é uma **solução ótima**[10] de (P) se x é uma solução viável e para toda solução viável x' temos que $\sum_{j=1}^{n} c_j x_j \geq \sum_{j=1}^{n} c_j x'_j$. Estendemos esse conceito de forma natural para o contexto de problemas de otimização que não são representados por programas lineares.

Um conceito muito importante para a teoria da Programação Linear é o programa dual. Considere o programa linear (P), que chamaremos de **Programa Primal**, apresentado acima. O **Programa Dual**, representado por (D), consiste em encontrar $y \in \mathbb{Q}^m$ que

$$\begin{array}{llrcll} \text{(D)} & \text{minimize} & \displaystyle\sum_{j=1}^{m} b_i y_i & & & \\ & \text{sujeito a} & \displaystyle\sum_{i=1}^{m} A_{ij} y_i & \geq & c_j, & \forall j \in \{1, \ldots, n\} \\ & & y_i & \geq & 0, & \forall i \in \{1, \ldots, m\}. \end{array}$$

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 uma 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 1.4.1 (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.

Por vezes, queremos considerar programas lineares com uma quantidade exponencial de restrições em relação ao tamanho da entrada original. Por exemplo, imagine que temos um grafo e a partir dele geramos um programa linear com uma restrição para cada subconjunto dos vértices. Nesse caso, gerar todo o programa linear consome tempo exponencial. Felizmente, um resultado de Grötschel, Lovász e Schrijver [GLS88] nos mostra como contornar esse problema.

No que segue, quando escrevemos "polinomial" ou "exponencial" queremos dizer polinomial ou exponencial, respectivamente, no tamanho da representação da entrada original e não no tamanho do programa linear.

Dado um programa linear (P) e um vetor $x \in \mathbb{Q}^n$, o **Problema da Separação** consiste em determinar se x é viável para (P) (mas não necessariamente ótimo) e, caso x não seja viável,

[10] O uso dos termos "solução viável" e "solução ótima" é fruto de discussão. Por um lado, parece redundante chamar uma solução de viável, mas, por outro lado, podemos pensar que a solução de um problema de otimização é uma entidade que atinge o máximo (ou o mínimo, dependendo do objetivo do problema) em vez de algo que apenas satisfaz as restrições do problema (é viável). Assim, optamos por usar os termos solução ótima para indicar explicitamente a otimalidade e solução viável para indicar viabilidade sem implicar em otimalidade.

apresentar uma restrição violada por x. Grötschel et al. provaram que é possível resolver o Problema da Separação de (P) em tempo polinomial se e somente se é possível resolver o problema de encontrar uma solução ótima de (P) em tempo polinomial. Em particular, eles mostraram como resolver um programa linear com um número exponencial de restrições em tempo polinomial, utilizando o Método dos Elipsóides e um oráculo capaz de resolver o problema da separação em tempo polinomial.

O resultado de Grötschel et al. permite também que lidemos com um programa linear que tenha um número exponencial de variáveis. Considere um programa primal (P) com um número exponencial de variáveis e um número polinomial de restrições e seja (D) o seu dual. Note que (D) tem um número polinomial de variáveis e um número exponencial de restrições. Suponha que tenhamos um oráculo polinomial para o problema da separação de (D), então, pelo resultado de Grötschel et al., podemos encontrar uma solução ótima (se existir) para (D) em tempo polinomial. Sejam $\{R_1, \ldots, R_k\}$ as restrições devolvidas por tal oráculo (ao analisar um candidato à solução que, de fato, era inviável) enquanto resolvemos (D). Note que k é limitado polinomialmente, já que encontramos uma solução ótima de (D) em tempo polinomial. Seja (D') o programa linear obtido a partir de (D) considerando apenas as restrições em $\{R_1, \ldots, R_k\}$. Note que o valor de uma solução ótima de (D') é igual ao valor de uma solução ótima de (D). Agora, considere o programa primal (P') construído a partir de (P) considerando apenas as variáveis correspondentes às restrições $\{R_1, \ldots, R_k\}$ (as outras variáveis são consideradas nulas). Note que, (D') é o programa dual de (P'), assim, usando o Teorema Forte da Dualidade e transitividade, temos que o valor de uma solução ótima de (P') é igual ao valor de uma solução ótima de (P). Basta então notar que podemos encontrar uma solução ótima de (P') em tempo polinomial pois (P') tem um número polinomial de restrições e de variáveis (já que k é limitado polinomialmente). Em particular, a solução ótima encontrada tem um número polinomial de variáveis não-nulas e o tamanho de sua representação é limitado polinomialmente pelo tamanho da entrada original.

Tal processo foi utilizado por Karmarkar e Karp [KK82] para encontrar uma solução aproximada do programa linear das configurações viáveis para o Problema do Empacotamento. Recomendamos o livro de Lau et al. [LRS11] que apresenta o resultado de Karmarkar e Karp de maneira bastante didática, juntamente com o processo descrito acima (aplicado ao Problema do Empacotamento).

Indicamos o livro de Chvátal [Chv83] para um aprofundamento no assunto de programação linear pois apresenta uma introdução ao assunto juntamente com os resultados essenciais da área.

1.4.1 Programação Inteira Mista

Na **Programação Inteira Mista**, além das restrições usando desigualdades, 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 inteira mista é NP-difícil [GJ79].

Apesar de não termos a garantia de resolver um **Programa Inteiro Misto** (MIP, *Mixed Integer Program*) em tempo polinomial, na prática MIPs são usados para modelar uma grande quantidade de problemas pois, para muitos casos, existem implementações eficientes do método *Branch-and-Bound* (que normalmente é utilizado para a resolução de MIPs) que obtém ótimos resultados em situações reais. Quando formulamos um determinado problema de otimização utilizando um programa inteiro misto, dizemos que tal programa é uma **Formulação MIP** para tal problema.

Dado um MIP, podemos remover as restrições de integralidade para obter um programa linear. Chamamos esse programa linear de **Relaxação Linear** do MIP em questão. Tal conceito é muito importante para a Programação Inteira Mista e até mesmo para o projeto de algoritmos de aproximação já que a relaxação linear dá um limitante superior para o valor de uma solução ótima do MIP (no caso da maximização; para minimização trata-se de um limitante inferior).

Nemhauser e Wolsey [NW99] apresentam em grandes detalhes diversos aspectos da área de programação inteira mista, inclusive o método de *Branch-and-Bound* mencionado acima.

1.5 Notações e Definições

Nessa seção apresentamos algumas notações e definições que serão úteis ao longo do texto. As notações específicas de cada problema são apresentadas nos seus respectivos capítulos.

Para simplificar a notação, para um inteiro k, denotamos por $[k]$ o conjunto $\{1, \ldots, k\}$. Durante todo o texto consideramos um conjunto I de itens e um conjunto B de compradores. Exceto quando dito o contrário, temos n itens e m compradores de forma que $I = [n]$ e $B = [m]$. Sempre que possível denotamos um elemento de I por i e um elemento de B por b. Por simplicidade, em vez de escrever $i \in I$ ou $b \in B$, por vezes dizemos apenas "um item i" ou "um comprador b".

Além disso, buscamos manter uma notação consistente, denotando o valor do item i para o comprador b por v_{ib} e denotando o preço do item i por p_i. Chamamos v de valoração e p de precificação. Por vezes usamos o termo valoração para representar o valor de um item i para um comprador b, porém deixamos claro pelo contexto que não nos referimos a v e sim a v_{ib}, assim como às vezes utilizamos os termos "valorações de um item" ou "valorações de um comprador" para restringir o foco em um item ou comprador específico.

Durante o texto consideramos grafos bipartidos formados por itens e compradores. Denotamos por (U, V, E) um grafo bipartido onde as partes são U e V e $E \subseteq U \times V$ denota o conjunto de arestas desse grafo. Denotamos uma aresta ligando $u \in U$ a $v \in V$ por (u, v). Note que utilizamos um par ordenado em vez do usual conjunto de tamanho 2 para representar as arestas desse grafo. O objetivo dessa notação é deixar claro que o primeiro elemento pertence a U e o segundo elemento pertence a V. Quando desejável, consideramos o grafo bipartido com pesos nas arestas (I, B, E), onde o peso de uma aresta é dado por v, e $(i, b) \in E$ se e somente se $v_{ib} > 0$, isto é, consideramos apenas as arestas com valoração positiva.

Outro elemento comum aos três problemas são as alocações. Uma alocação, de forma geral, é uma função de B em 2^I, onde 2^I representa o conjunto de todos os subconjuntos de I. Por vezes, consideramos que uma alocação x é representada por uma matriz binária x indexada por $I \times B$, de forma que $x_{ib} = 1$ se e somente se i está alocado para b. Porém, no Capítulo 4 consideramos que uma alocação é um subconjunto das arestas do grafo bipartido descrito acima. Isso decorre da relação entre os problemas discutidos em tal capítulo com o Problema do Emparelhamento Máximo.

Em nossas figuras, utilizamos o grafo bipartido descrito acima (onde apenas arestas com valoração positiva estão presentes) representando I por vértices de cor preta e B por vértices de cor branca. Além disso, representamos um item k por i_k e um comprador k por b_k, para que não haja confusão entre itens e compradores. Usualmente utilizamos arestas azuis grossas para indicar uma alocação. A Figura 1.2 mostra um exemplo de tal convenção.

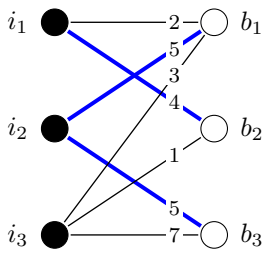


Figura 1.2: *Exemplo das convenções utilizadas nas figuras. O item k é representado por i_k e o comprador k é representado por b_k. As arestas indicam apenas as valorações positivas (com o respectivo valor sobre a aresta). As arestas em azul (grossas) indicam uma alocação de itens a compradores.*

Um conceito muito importante para a Teoria de Grafos e que utilizamos durante o texto é o conceito de emparelhamento.

Definição. Dado um grafo $G = (V, E)$, um conjunto $M \subseteq E$ é um **emparelhamento** se, para todo $v \in V$, existe no máximo uma aresta de M adjacente a v.

Além disso, durante o texto utilizamos uma generalização do conceito de emparelhamento.

Definição. Dado um grafo $G = (V, E)$ e um vetor de inteiros não-negativos ρ indexado por V, um conjunto $M \subseteq E$ é um ρ-**emparelhamento** se, para todo $v \in V$, existem no máximo ρ_v arestas de M adjacentes a v.

Em particular, como estamos interessados em grafos bipartidos (formados por itens e compradores), através do Método Húngaro [Kuh55, Mun57] é possível encontrar um emparelhamento de peso máximo em um grafo bipartido em tempo polinomial. Além disso, através de algumas modificações em tal método, é possível também encontrar um ρ-emparelhamento (para qualquer ρ) em tempo polinomial [Sch03].

1.6 Resultados Obtidos e Organização do Texto

No Capítulo 2, apresentamos o Problema da Compra Máxima, focando no caso de oferta limitada e considerando as versões com e sem escada de preços (uma ordem pré-estabelecida nos preços dos itens). Para a versão sem escada de preços, apresentamos uma $e/(e-1)$-aproximação e para a versão com escada de preços apresentamos uma família de algoritmos parametrizada por um racional positivo ε tal que, para um determinado ε, o algoritmo é uma $(2 + \varepsilon)$-aproximação. Ambos os resultados melhoram as razões de aproximação conhecidas para esses problemas. Esse trabalho, em conjunto com Cristina G. Fernandes, foi aceito para publicação no *11th Latin American Symposium* (LATIN 2014) [FS14a]. Nos próximos meses submeteremos a versão completa desse trabalho para publicação em uma revista da área.

No Capítulo 3, introduzimos o Problema da Precificação Livre de Inveja, apresentamos novas formulações MIP para esse problema e mostramos uma comparação empírica dessas novas formulações com a formulação previamente conhecida na literatura. Pelos nossos resultados, as nossas formulações representam um grande avanço na busca de soluções ótimas para esse problema. Para realizar essa comparações, introduzimos três modelos de geração de instâncias que são baseados em argumentos econômicos na busca de gerar instâncias realísticas e difíceis de serem resolvidas pelas formulações. Os geradores foram implementados e disponibilizados como software livre[11] para que possam ser utilizados pela comunidade acadêmica na geração de instâncias de leilões de demanda unitária. Consideramos também uma variante do Problema da Precificação Livre de Inveja e mostramos que a existência de um algoritmo com razão de aproximação constante para essa variante implicaria em um algoritmo com razão de aproximação constante para o problema original. Esse é um trabalho em conjunto com Cristina G. Fernandes, Carlos E. Ferreira e Álvaro J. Franco e foi aceito para publicação no *3nd International Symposium on Combinatorial Optimization* [FFFS14].

No Capítulo 4, consideramos o Leilão de Anúncios de Segundo Preço com Lances Binários e o Emparelhamento de Segundo Preço em mercados competitivos. Revemos alguns resultados da literatura do ponto de vista de mercados competitivos, mostrando que a competitividade não melhora a razão de aproximação do algoritmo previamente conhecido para o Emparelhamento de Segundo Preço e que o Emparelhamento de Segundo Preço continua APX-difícil mesmo com essa restrição. Em vista disto, apresentamos um novo algoritmo com razão de aproximação melhor do que a previamente conhecida para o Emparelhamento de Segundo Preço em mercados de alta competitividade. Apresentamos também alguns estudos empíricos comparando variantes do nosso algoritmo. Considerando o Leilão de Anúncios de Segundo Preço com Lances Binários, mostramos como usar o nosso algoritmo para o Emparelhamento de Segundo Preço para obter uma aproximação para tal problema. Tal trabalho foi apresentado no *2nd International Symposium on Combinatorial Optimization* com o título *Second-Price Ad Auctions with Binary Bids and Markets with Good Competition* [FS12] e, posteriormente, submetemos uma versão completa do artigo para o periódico *Theoretical Computer Science* que foi aceita e deverá ser publicada em breve [FS14b].

Por fim, no Capítulo 5, apresentamos algumas conclusões relacionadas aos problemas discutidos nessa tese.

[11] Disponível no site https://github.com/schouery/unit-demand-market-models.

Capítulo 2

O Problema da Compra Máxima

Um problema econômico interessante enfrentado por empresas que vendem produtos ou oferecem serviços a consumidores é escolher o preço dos produtos ou serviços com o objetivo de maximizar o lucro. Se os preços forem altos, então alguns consumidores não irão querer (ou não poderão) comprar os produtos, e se os preços forem baixos, então a empresa poderá obter um lucro baixo. Esse é um problema vastamente estudado, com modelos diferentes para situações diferentes e uma grande diversidade de abordagens [OSW84, OSW87, Sen82, Smi86].

Uma forma de abordar tal problema é através de modelagens não-paramétricas [RRG06] onde a empresa coleta as preferências dos grupos de consumidores (por exemplo, usando um site) e otimiza de acordo com algumas hipóteses sobre o comportamento dos consumidores.

Em tal cenário, temos n produtos ou serviços (que chamaremos de itens) e temos m consumidores (que chamaremos de compradores) no mercado. Em um primeiro momento, consideramos que existe uma oferta ilimitada de cada item. A empresa (que chamaremos de leiloeiro) deseja atribuir preços p_i para cada item i com o objetivo de maximizar o seu lucro (a soma dos preços dos itens vendidos considerando multiplicidades). Para isso, o leiloeiro coleta informações sobre as valorações dos compradores, isto é, o maior valor que um comprador b está disposto a pagar pelo item i (que denotamos por v_{ib}).

Os modelos descritos a seguir consideram que o mercado é de demanda unitária, isto é, cada comprador deseja comprar no máximo um item. Ademais, eles consideram que um comprador b comprará um item apenas se ele não for muito caro (isto é, se $p_i \leq v_{ib}$). Chamamos tais itens de **viáveis** para b. Se não existe um item viável para b então b não compra nenhum item.

Três modelos foram introduzidos por Rusmevichientong et al. [RRG06]. No **Problema da Compra Mínima** (*Min-Buying Problem*), considera-se que cada comprador irá comprar um dos itens mais baratos que são viáveis. No **Problema da Compra por Preferência** (*Rank-Buying Problem*), sabe-se também a ordem de preferência dos itens para cada comprador e considera-se que um comprador irá comprar o item preferido dentro dos que são viáveis. Finalmente, no **Problema da Compra Máxima** (*Max-Buying Problem*), considera-se que cada comprador irá comprar um dos itens mais caros entre os que são viáveis.

Existem também algumas variantes dos problemas mencionados acima. Uma delas é considerar que cada item tem um número máximo de cópias que podem ser vendidas, isto é, que temos **Oferta Limitada**. Note que esta é uma generalização do problema original, pois podemos tomar o número de cópias de cada item como o número de compradores e isso corresponde a oferta ilimitada. Por vezes, uma empresa sabe (ou deseja) uma ordenação nos preços dos seus produtos e pode impor uma **Escada de Preços**, isto é, considerar apenas precificações que satisfaçam $p_1 \geq p_2 \geq \cdots \geq p_n$. A motivação para tal restrição é permitir que uma empresa com uma linha de produtos possa definir uma ordem nos preços de seus produtos. Por exemplo, assim é possível garantir que o produto de maior qualidade tenha um preço maior, fazendo com que um produto "*premium*" seja mais caro que um produto "básico". Finalmente, podemos também considerar **Orçamentos Uniformes**, onde cada comprador b tem um valor V_b tal que, para cada item i, $v_{ib} = \{0, V_b\}$.

Rusmevichientong et al. [RRG06] mostraram que, se exigirmos uma escada de preços, então é

possível resolver o Problema da Compra Mínima com Orçamentos Uniformes em tempo polinomial. Posteriormente, Aggarwal et al. [AFMZ04] provaram vários resultados para esses modelos considerando orçamentos não-uniformes. Eles apresentaram um Esquema de Aproximação Polinomial para o Problema da Compra Máxima com Escada de Preços e mostraram como reduzir o Problema da Compra por Preferência com Escada de Preços ao Problema da Compra Máxima com Escada de Preços quando a oferta é ilimitada. Eles também apresentaram uma 4-aproximação para o Problema da Compra Máxima com Oferta Limitada e Escada de Preços. Para o caso onde não exigimos uma escada de preços, Aggarwal et al. apresentaram uma $e/(e-1)$-aproximação para o Problema da Compra Máxima juntamente com um limitante inferior de 16/15, que pode ser melhorado para 8/7 utilizando um limitante inferior apresentado por Guruswami e Khot [GK05] para um caso particular do Problema da 3-Satisfatibilidade Máxima (note que $e/(e-1) \approx 1{,}58$ e $8/7 \approx 1{,}14$), uma $\log(|B|)$-aproximação que pode ser usada para os três modelos e, para uma constante ε, um limitante inferior de $1+\varepsilon$ para o Problema da Compra Mínima, ou seja, não existe PTAS para o Problema da Compra Mínima a não ser que P = NP.

Briest e Krysta [BK07] mostraram que existem constantes positivas ε, ε' e ε'' para as quais o Problema da Compra Mínima (com ou sem escada de preços) não é aproximável por $\mathrm{O}(\log^{\varepsilon}|B|)$ a não ser que $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{O}(\log\log n)})$ e não é aproximável por $\mathrm{O}(|I|^{\varepsilon'})$ e por $\mathrm{O}(\ell^{\varepsilon''})$, onde ℓ é um limitante superior no número de valorações não nulas de cada comprador, a não ser que $\mathrm{NP} \subseteq \mathrm{DTIME}(2^{\mathrm{O}(n^{\delta})})$ para todo $\delta > 0$. Eles mostraram também que o Problema da Compra Máxima com Escada de Preços é fortemente NP-difícil e apresentaram uma 2-aproximação para o Problema da Compra Máxima com Oferta Limitada (sem escada de preços).

Para o Problema da Compra Mínima com Orçamentos Uniformes, Briest [Bri08] mostrou que o problema não é aproximável por $\mathrm{O}(\log^{\varepsilon}|B|)$ para algum $\varepsilon > 0$ se considerarmos algumas hipóteses de dificuldade em relação a refutar instâncias aleatórias do Problema da 3-Satisfatibilidade ou em aproximar o problema do conjunto independente balanceado em grafo bipartidos com grau constante. Posteriormente, Chalermsook et al. [CCKK12] apresentaram limitantes inferiores de $\Omega(\ell^{1/2})$ (a não ser que P = NP) e $\Omega(\log^{1-\varepsilon}(|I|+|B|))$ para qualquer constante positiva ε (a não ser que $\mathrm{NP} \subseteq \mathrm{DTIME}(n^{\mathrm{O}(\log^{\delta} n)})$, onde δ é uma constante que depende de ε). Por fim, Chalermsook et al. [CLN13] apresentaram um limitante inferior de $\Omega(\min(\ell^{1-\varepsilon}, |I|^{1/2-\varepsilon}))$ para qualquer constante positiva ε (usando a Hipótese do Tempo Exponencial).

Enquanto trabalhamos nesse problema, focamos no caso de oferta limitada, que é uma generalização do problema básico. Projetamos dois novos algoritmos de aproximação para o Problema da Compra Máxima com Oferta Limitada, um para o caso geral e outro para o caso com o requisito de escada de preços. Ambos os algoritmos melhoram as razões de aproximação conhecidas para esses problemas.

Para o Problema da Compra Máxima com Oferta Limitada (sem escada de preços), apresentamos uma $e/(e-1)$-aproximação (lembre-se que $e/(e-1) < 1.582$), melhorando o limitante superior de Briest e Krysta [BK07] (uma 2-aproximação). Além disso, esse algoritmo tem a mesma razão de aproximação que o algoritmo para o Problema da Compra Máxima (com Oferta Ilimitada) apresentado por Aggarwal et al. [AFMZ04] embora os dois algoritmos sejam bem diferentes. O algoritmo utiliza a relaxação linear de uma formulação por programação inteira com um número exponencial de variáveis, para fazer um arredondamento probabilístico, e também explora uma estrutura do problema que pode ser útil no desenvolvimento de algoritmos de aproximação para outros dos problemas previamente descritos. Mostramos também como obter um algoritmo determinístico através da desaleatorização do nosso algoritmo usando o Método das Esperanças Condicionais [ES73, Spe87].

Para o Problema da Compra Máxima com Oferta Limitada e Escada de Preços, apresentamos uma família de algoritmos parametrizada por um racional positivo ε tal que o algoritmo é polinomial para ε constante e fornece uma razão de aproximação de $2+\varepsilon$.

2.1 Modelo e Notação

Denotamos por $I = [n]$ o conjunto de itens e por $B = [m]$ o conjunto de compradores.

Definição. Uma **valoração** é uma matriz inteira não-negativa v indexada por $I \times B$.

O número v_{ib} representa o valor do item i para o comprador b.

Definição. Uma **precificação** é um vetor racional não-negativo indexado por I.

Definição. Uma **alocação** é um vetor x indexado por B onde x_b é o conjunto de itens alocados para o comprador b.

Como iremos nos restringir a demanda unitária, temos que $|x_b| \leq 1$ para todo comprador b. Porém, na Seção 2.5 iremos apresentar uma variante do Problema da Compra Máxima com Oferta Limitada e Escada de Preços que permite que os compradores recebam mais de um item. Mesmo considerando a demanda unitária, note que uma alocação não é necessariamente um emparelhamento entre itens e compradores, pois o mesmo item pode ser alocado a mais de um comprador (cada um recebe uma cópia do item).

Definição. Dado uma valoração v e uma precificação p, um item i é **viável** para o comprador b se $v_{ib} \geq p_i$.

Isto é, estamos considerando um leilão individualmente racional. Se o preço do item alocado a um comprador for maior do que a valoração de tal item para o comprador, ele não comprará tal item.

> **Problema da Compra Máxima Limitada**[1]: dados uma valoração v e um vetor C indexado por I, desejamos encontrar uma precificação p e uma alocação x que maximiza o lucro do leiloeiro de forma que cada item i é alocado a no máximo C_i compradores e cada comprador ou não recebe nenhum item ou recebe um item viável.

Assumiremos, sem perda de generalidade, que $C_i \leq |B|$ para todo item i, já que um item não pode ser vendido para mais do que $|B|$ compradores. Se $C_i = |B|$ para todo item i, o problema torna-se equivalente ao Problema da Compra Máxima (com oferta ilimitada).

Observe que, apesar do nome do problema, podemos pensar que numa solução viável, um item alocado a um comprador deve ser viável para esse comprador, mas não necessariamente de preço máximo. No caso de oferta ilimitada, por causa da função objetivo, uma solução ótima sempre alocará itens viáveis de preço máximo. Já no caso de oferta limitada, numa solução ótima os itens viáveis de preço máximo de um comprador podem estar com todas as suas cópias alocadas a outros compradores. Assim, tal comprador pode receber um item viável não necessariamente de preço máximo ou até mesmo não receber nenhum item. A Figura 2.1 apresenta uma instância do problema, juntamente com duas soluções.

A seguir formalizamos a variante do problema onde exigimos uma escada de preços.

> **Problema da Compra Máxima Limitada EP**: variante do Problema da Compra Máxima Limitada em que os preços estão restritos a serem não-crescentes, isto é, $p_1 \geq \cdots \geq p_n$, onde $n = |I|$.

Note que, dada uma precificação p, é fácil encontrar uma alocação x que maximize o lucro sobre todas as alocações onde cada comprador ou não recebe nenhum item ou recebe um item viável. De fato, basta considerar o grafo bipartido $G = (I, B, E)$ onde existe uma aresta $(i, b) \in E$ se e somente se $p_i \leq v_{ib}$ (isto é, i é viável para b) e o peso de uma aresta $(i, b) \in E$ é p_i. Note que x corresponde a um ρ-emparelhamento de peso máximo em G, onde $\rho(i) = C_i$ para todo item i e $\rho(b) = 1$ para todo comprador b e que o lucro obtido por x e p é, precisamente, o peso de tal ρ-emparelhamento. Assim, como é possível encontrar um ρ-emparelhamento de peso máximo em tempo polinomial no tamanho de v [Mun57], é possível encontrar uma tal alocação x em tempo polinomial no tamanho de v.

[1] A partir desse momento, nos referimos ao Problema da Compra Máxima com Oferta Limitada como, simplesmente, Problema da Compra Máxima Limitada.

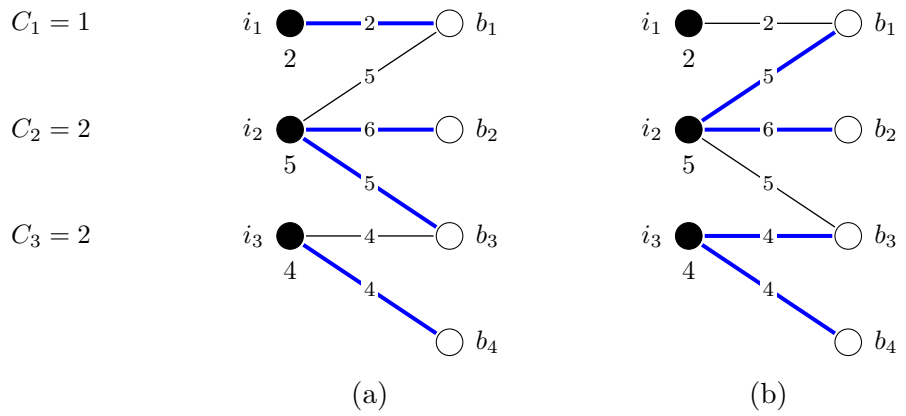


Figura 2.1: *Dois exemplos de alocação e precificação para o Problema da Compra Máxima. Ambas as figuras representam a mesma instância, onde há uma cópia do item* 1 *e duas cópias dos itens* 2 *e* 3*. O valor abaixo do vértice representando o item é o preço escolhido (os dois exemplos têm a mesma precificação). Em* (a)*, o lucro obtido pela alocação dada pelas arestas azuis (grossas) é* 16 *e em* (b) *o lucro é* 18*.*

2.2 Um Limitante Inferior para o Problema da Compra Máxima

Nessa seção apresentamos um resultado de Aggarwal et al. [AFMZ04] que apresenta um limitante inferior para o Problema da Compra Máxima com Orçamentos Uniformes. Começamos apresentando formalmente o problema que consideramos nessa seção.

> **Problema da Compra Máxima com Orçamentos Uniformes**: caso particular do Problema da Compra Máxima Limitada onde $C_i = |B|$ para cada item i (isto é, a oferta é ilimitada) e para cada comprador b, existe um valor V_b tal que $v_{ib} \in \{0, V_b\}$ para todo item i.

Antes de apresentar o resultado para o Problema da Compra Máxima com Orçamentos Uniformes, introduziremos o Problema da Satisfatibilidade Máxima. No que segue, denotamos por $\vee$ a operação "ou", por $\wedge$ a operação "e" e por $\neg$ a operação de negação. Começamos apresentando a estrutura das fórmulas booleanas que consideramos.

Definição. Um **literal** ℓ é uma fórmula booleana de uma única variável X tal que ou $\ell = X$ ou $\ell = \neg X$. Dizemos que ℓ é positivo se $\ell = X$ e dizemos que ℓ é negativo se $\ell = \neg X$.

Definição. Uma **cláusula** C é uma fórmula booleana tal que $C = (\ell_1 \vee \ell_2 \vee \cdots \vee \ell_k)$ onde ℓ_i é um literal para todo $1 \leq i \leq k$.

Definição. Uma cláusula C é **positiva** se C é composta apenas de literais positivos e C é **negativa** se C é composta apenas de literais negativos. Se C não é nem positiva, nem negativa, dizemos que C é **mista**.

Definição. Uma fórmula booleana ϕ está na **forma normal conjuntiva** se $\phi = C_1 \wedge C_2 \wedge \cdots \wedge C_n$ onde C_i é uma cláusula para todo $1 \leq i \leq n$.

Estamos prontos para enunciar o Problema da Satisfatibilidade Máxima.

> **Problema da Satisfatibilidade Máxima** (denotado por Max SAT): dada uma fórmula ϕ na forma normal conjuntiva, desejamos encontrar uma atribuição para as variáveis de ϕ que maximize o número de cláusulas satisfeitas.

Estamos interessados em alguns casos particulares do Max SAT. Denotamos por Max 3SAT o caso particular do Max SAT onde as instâncias são compostas apenas por cláusulas com no máximo 3 literais e por Max E3SAT, o caso particular do Max 3SAT onde as instâncias são

compostas apenas por cláusulas com exatamente 3 literais. Ademais, denotamos por Max NM-3SAT o caso particular do Max 3SAT onde nenhuma cláusula é mista e denotamos por Max NM-E3SAT o caso particular do Max E3SAT onde nenhuma cláusula é mista.

A prova do limitante inferior para o Problema da Compra Máxima com Orçamentos Uniformes apresentada por Aggarwal et al. [AFMZ04] utiliza o Teorema de Håstad (apresentado abaixo) para concluir que não existe um algoritmo para o Max NM-3SAT com razão de aproximação menor do que 16/15 a não ser que P = NP. Utilizando esse resultado, Aggarwal et al. concluem que não existe um algoritmo para o Problema da Compra Máxima com Orçamentos Uniformes com razão de aproximação menor do que 16/15 a não ser que P = NP.

Teorema 2.2.1 (Håstad [Hås01])**.** Não existe um algoritmo para o Max 3SAT com razão de aproximação menor do que 8/7 a não ser que P = NP.

Porém, Guruswami e Khot [GK05] provaram (posteriormente) o seguinte resultado para o Max NM-E3SAT, que pode ser usado para melhorar o limitante inferior para o Problema da Compra Máxima com Orçamentos Uniformes apresentado por Aggarwal et al. [AFMZ04] utilizando a mesma prova.

Teorema 2.2.2 (Guruswami e Khot [GK05])**.** Não existe um algoritmo para o Max NM-E3SAT com razão de aproximação menor do que 8/7 a não ser que P = NP.

A seguir apresentamos o resultado de Aggarwal et al. levemente modificado para utilizar o resultado de Guruswami e Khot.

Teorema 2.2.3. Não existe um algoritmo para o Problema da Compra Máxima com Orçamentos Uniformes com razão de aproximação menor do que 8/7 a não ser que P = NP.

Demonstração. Seja $\varepsilon > 0$. Vamos mostrar como, a partir de uma $(8/7 - \varepsilon)$-aproximação para o Problema da Compra Máxima com Orçamentos Uniformes, podemos obter uma $(8/7 - \varepsilon/7)$-aproximação para o Max NM-E3SAT.

Seja ϕ uma instância do Max NM-E3SAT e seja n o número de cláusulas de ϕ. Vamos construir uma instância $v(\phi)$ do Problema da Compra Máxima com Orçamentos Uniformes a partir de ϕ. O conjunto de itens de $v(\phi)$ é formado pelo conjunto de variáveis de ϕ e o conjunto de compradores de $v(\phi)$ é construído da seguinte forma. Para cada cláusula C positiva de ϕ temos um comprador c tal que, para todo item i, $v_{ic} = n$ se i é um dos literais de C e $v_{ic} = 0$ caso contrário. Para cada cláusula C negativa de ϕ temos n compradores $c_1, \ldots, c_n$ tais que, para todo $1 \leq j \leq n$ e para todo item i, $v_{ic_j} = 1$ se $\neg i$ é um dos literais de C e $v_{ic_j} = 0$, caso contrário. Note que $v(\phi)$ pode ser construída em tempo polinomial no tamanho da representação de ϕ.

Como v é composta apenas pelos valores 0, 1 e n, sem perda de generalidade, supomos que uma precificação p para $v(\phi)$ é tal que $p_i \in \{1, n\}$ para todo item i. Assim, podemos definir uma bijeção f entre as precificações para $v(\phi)$ e as atribuições para as variáveis de ϕ de forma que, para uma precificação p, $f(p)$ é uma atribuição onde cada variável i tem valor "verdadeiro" se $p_i = n$ e valor "falso" se $p_i = 1$.

Seja p uma precificação e seja x uma alocação que aloca um dos itens mais caros entre os que são viáveis para cada comprador. Note que, para uma cláusula C positiva, se existe um item i com preço $p_i = n$ tal que i é um literal de C, então C é satisfeita por $f(p)$ e obtemos lucro n do comprador c e, se não existe tal item, então obtemos lucro 1 do comprador c. Para uma cláusula C negativa, se existe um item i com preço $p_i = 1$ tal que $\neg i$ é um literal de C, então C é satisfeita por $f(p)$ e obtemos lucro n (no total) dos compradores $c_1, \ldots, c_n$ e, se não existe tal item, então obtemos lucro 0 de tais compradores.

Para uma precificação p, denotamos por $k(p)$ o número de cláusulas de ϕ satisfeitas por $f(p)$ e denotamos por $r(p)$ o número de cláusulas positivas de ϕ não satisfeitas por $f(p)$. Temos que o lucro obtido a partir de p e x é $k(p)n + r(p)$.

Seja p^* a precificação de uma solução ótima para $v(\phi)$ e suponha que existe uma $(8/7 - \varepsilon)$-aproximação para o Problema da Compra Máxima com Orçamentos Uniformes para

algum $\varepsilon > 0$. Seja p a precificação encontrada por tal algoritmo. Como $(8/7-\varepsilon)^{-1} = 7/(8-7\varepsilon)$, temos que $k(p)n + r(p) \geq \frac{7}{8-7\varepsilon}(k(p^*)n + r(p^*)) \geq \frac{7}{8-7\varepsilon}k(p^*)n$. Além disso, podemos supor, sem perda de generalidade, que $r(p) \leq k(p)$, pois, caso contrário, a precificação $\tilde{p}$ onde $\tilde{p}_i = n$ para todo item i satisfaz tal propriedade e obtém lucro maior ou igual ao lucro de p. De fato, seja $\overline{r}$ o número de cláusulas positivas de ϕ, se $r(p) > k(p)$, temos que $k(\tilde{p})n + r(\tilde{p}) = \overline{r}n \geq r(p)n \geq (k(p)+1)n \geq k(p)n + r(p)$, de onde concluímos que $\tilde{p}$ obtém um lucro maior ou igual ao lucro de p. Temos que

$$k(p) \geq \frac{7}{8-7\varepsilon}k(p^*) - \frac{r(p)}{n} \geq \frac{7}{8-7\varepsilon}k(p^*) - \frac{k(p)}{n}$$

e portanto

$$k(p) \geq \left(\frac{7}{8-7\varepsilon}\right)\left(\frac{n}{n+1}\right)k(p^*).$$

Para provar o resultado desejado, consideraremos que $n \geq \frac{8-7\varepsilon}{6\varepsilon}$. Note que esse limitante inferior depende apenas da qualidade do algoritmo de aproximação para o Problema da Compra Máxima com Orçamentos Uniformes e não de algum parâmetro da entrada. Ademais, podemos resolver as instâncias com no máximo $\frac{8-7\varepsilon}{6\varepsilon}$ cláusulas de forma exata. Segue que

$$k(p) \geq \left(\frac{7}{8-7\varepsilon}\right)\left(\frac{n}{n+1}\right)k(p^*) \geq \left(\frac{7}{8-7\varepsilon}\right)\left(\frac{\frac{8-7\varepsilon}{6\varepsilon}}{\frac{8-7\varepsilon}{6\varepsilon}+1}\right) = \frac{7}{8-\varepsilon}k^* = \frac{1}{\frac{8}{7}-\frac{\varepsilon}{7}}k^*.$$

Portanto, se existe uma $(8/7-\varepsilon)$-aproximação para o Problema da Compra Máxima com Orçamentos Uniformes para algum $\varepsilon > 0$, então existe uma $(8/7-\varepsilon/7)$-aproximação para o Max NM-E3SAT o que, pelo Teorema 2.2.2 de Guruswami e Khot, implicaria que P = NP, de onde o resultado segue. □

2.3 Um Algoritmo para Oferta Limitada

A seguir, apresentamos um novo algoritmo de aproximação para o Problema da Compra Máxima Limitada cuja razão de aproximação é melhor do que a razão de aproximação do algoritmo apresentado por Briest e Krysta [BK07]. Essa aproximação também pode ser usada para o Problema da Compra Máxima (com oferta ilimitada) e tem a mesma razão de aproximação do algoritmo apresentado por Aggarwal et al. [AFMZ04] para tal problema.

Primeiramente considere uma solução (x, p) do Problema da Compra Máxima Limitada, um item i e o conjunto $S \neq \emptyset$ de compradores que, de acordo com x, compraram o item i. Note que, porque i é viável para todo comprador em S, temos que $p_i \leq \min\{v_{ib} : b \in S\}$. Se $p_i < \min\{v_{ib} : b \in S\}$, então (x, p) não pode ser uma solução ótima porque é possível aumentar o preço de i e obter uma solução estritamente melhor. Concluímos que podemos assumir, sem perda de generalidade, que $p_i = \min\{v_{ib} : b \in S\}$ para todo item i que foi comprado pelo conjunto S de compradores[2]. A Figura 2.2 mostra uma solução que não satisfaz tal propriedade para ilustrar como podemos facilmente aumentar o preço de um item e obter uma solução melhor.

Vamos apresentar uma formulação por Programação Inteira para o Problema da Compra Máxima Limitada que é fortemente baseada na observação acima e nas próximas definições. A partir dessa formulação, projetaremos um algoritmo de aproximação que usa arredondamento probabilístico.

Definição. Para um item i, seja $\mathcal{S}(i) = \{(i, S) : S \subseteq B, |S| \leq C_i\}$. Chamamos $(i, S) \in \mathcal{S}(i)$ de uma **estrela** de 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 $S \subseteq B$ e um item i, $S \cup \{i\}$ induz uma estrela no grafo bipartido completo onde as partes da bipartição são I e B.

[2] Considerando que, para um item i e $S = \emptyset$, temos que $\min\{v_{ib} : b \in S\} = \infty$.

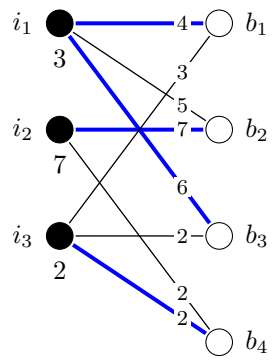


Figura 2.2: *Uma solução para o Problema da Compra Máxima com preço baixo. A alocação é indicada pelas arestas em azul (grossas) e os preços estão abaixo dos vértices que representam os itens. Note que, para a alocação representada acima, podemos aumentar o preço do item i_1 para 4, que é o valor da menor aresta (i_1, b) em que i_1 está alocado a b.*

Definição. Seja (i, S) uma estrela do item i. Denotamos $\min\{v_{ib} : b \in S\}$ por $P_{(i,S)}$, isto é, o preço do item i quando vendido para o conjunto S de compradores.

Note que uma solução viável do Problema da Compra Máxima Limitada pode ser vista como uma coleção de estrelas, uma para cada item, onde cada comprador está em no máximo uma estrela e se a estrela do item i é (i, S), então o preço de tal item é $P_{(i,S)}$.

A seguir apresentamos a nossa formulação, chamada (SF) (de *star formulation*), na qual temos um vetor binário x de variáveis com $|\mathcal{S}|$ posições. Para todo $(i, S) \in \mathcal{S}$, $x_{(i,S)}$ é 1 se e somente se o conjunto de compradores que recebe o item i é precisamente S. O objetivo é determinar x que

$$
\begin{array}{llrcll}
\text{(SF)} & \text{maximize} & \displaystyle\sum_{(i,S)\in\mathcal{S}} |S| P_{(i,S)} x_{(i,S)} & & & \\
& \text{sujeito a} & \displaystyle\sum_{(i,S)\in\mathcal{S}(i)} x_{(i,S)} & = & 1, & \forall i \in I \\
& & \displaystyle\sum_{(i,S)\in\mathcal{S}\colon b\in S} x_{(i,S)} & \leq & 1, & \forall b \in B \\
& & x_{(i,S)} & \in & \{0,1\}, & \forall (i,S) \in \mathcal{S}.
\end{array}
$$

A formulação (SF) pode ser vista como uma redução (exponencial) do nosso problema ao Problema do Empacotamento de Conjuntos [GJ79]. Uma ideia similar foi usada por Hochbaum [Hoc82] para obter uma $\mathrm{O}(\log n)$-aproximação para o Problema da Localização de Facilidades Métrico Sem Capacidades através de uma redução para o Problema da Cobertura de Conjuntos [GJ79], onde n é o número de clientes. No nosso caso, utilizamos a estrutura dos pesos dos conjuntos para obter uma aproximação de fator constante para o Problema da Compra Máxima Limitada.

Essa formulação pode ter $\Omega(|I|2^{|B|})$ variáveis. Mas, felizmente, é possível resolver a relaxação linear em tempo polinomial no tamanho (da representação) de v.

Lema 2.3.1. A relaxação linear da formulação (SF) pode ser resolvida em tempo polinomial no tamanho de v. Ademais, o tamanho da representação da solução ótima encontrada e o número de variáveis não-nulas de tal solução são limitados polinomialmente no tamanho de v.

Demonstração. Primeiramente, note que o dual da relaxação linear de (SF) pode ser descrito da seguinte forma: temos um vetor racional α com $|I|$ posições e um vetor racional não-negativo β com $|B|$ posições e o objetivo é determinar α e β que

$$
\begin{array}{llrcll}
\text{minimize} & \displaystyle\sum_{i\in I} \alpha_i + \sum_{b\in B} \beta_b & & & & \\
\text{sujeito a} & \alpha_i + \displaystyle\sum_{b\in S} \beta_b & \geq & P_{(i,S)}|S|, & & \forall (i,S) \in \mathcal{S} \\
& \alpha_i & \in & \mathbb{Q}, & & \forall i \in I \\
& \beta_b & \geq & 0, & & \forall b \in B.
\end{array}
$$

Iremos provar que é possível resolver o problema da separação para esse programa linear dual em tempo polinomial. A partir disso, e usando o resultado de Grötschel et al. [GLS88], concluiremos que é possível resolver esse programa dual em tempo polinomial no tamanho de v. Utilizando o processo descrito na Seção 1.4, concluiremos que existe um algoritmo polinomial no tamanho de v para encontrar uma solução ótima da relaxação linear de (SF) com a propriedade de que o número de variáveis não-nulas e o tamanho da representação de tal solução são limitados polinomialmente no tamanho de v.

Considere que temos vetores α e β que são candidatos à solução do dual acima. Queremos, em tempo polinomial no tamanho de v, decidir se α e β formam de fato uma solução viável e, caso α e β não formem uma solução viável, queremos apresentar uma restrição violada do dual.

Primeiramente, assumiremos que $\beta_b \geq 0$ para cada comprador b, pois se o comprador b for tal que $\beta_b < 0$, então α e β claramente não formam uma solução viável e podemos devolver a restrição $\beta_b \geq 0$ para provar.

Para $i \in I$, $b \in B$ e $k \in \{1, \ldots, C_i\}$, denote por $\mathcal{S}(i, b, k)$ o conjunto de estrelas (i, S) tais que $P_{(i,S)} = v_{ib}$, $b \in S$ e $|S| = k$. Se $\mathcal{S}(i, b, k) \neq \emptyset$, então existe uma estrela $(i, S) \in \mathcal{S}(i, b, k)$ que minimiza $\sum_{b \in S} \beta_b$. Além disso, note que existe uma estrela $(i, S') \in \mathcal{S}(i, b, k)$ tal que $\alpha_i + \sum_{b \in S'} \beta_b < P_{(i,S')}|S'|$ se e somente se $\alpha_i + \sum_{b \in S} \beta_b < P_{(i,S)}|S|$ já que $P_{(i,S)} = P_{(i,S')} = v_{ib}$ e $|S| = |S'| = k$.

Afirmamos que tal $(i, S) \in \mathcal{S}(i, b, k)$ que minimiza $\sum_{b \in S} \beta_b$ pode ser encontrado (se existir) em tempo polinomial no tamanho de v. Temos apenas que considerar os primeiros $k - 1$ compradores $b' \neq b$ em ordem não-decrescente de $\beta_{b'}$ tais que $v_{ib'} \geq v_{ib}$ (é claro que se existem menos do que $k - 1$ tais compradores, então $\mathcal{S}(i, b, k)$ é vazio). Além disso, podemos decidir se $\alpha_i + \sum_{b \in S} \beta_b \geq P_{(i,S)}|S|$ em tempo polinomial no tamanho de v.

Para concluir nossa prova, note que

$$\mathcal{S} = \bigcup_{\substack{i \in I, b \in B \\ k \in [C_i]}} \mathcal{S}(i, b, k).$$

Isto é, podemos iterar sobre todo $i \in I$, $b \in B$, e $k \in [C_i]$ e decidir se existe uma estrela em $\mathcal{S}(i, b, k)$ que viola uma restrição, de onde segue o resultado. □

Seja (i, S) uma estrela e considere um comprador b. Se $b \in S$, abusamos da notação e dizemos que b **pertence** a (i, S) (analogamente, dizemos que (i, S) contém b) e denotamos esse fato por $b \in (i, S)$.

A seguir mostramos como utilizamos a formulação por programação inteira para desenvolver um algoritmo de aproximação para o Problema da Compra Máxima Limitada usando arredondamento probabilístico.

StarRounding(I, B, v, C)

Seja x uma solução ótima da relaxação linear de (SF) para (I, B, v, C)
para cada item i **em** I
    Escolha uma estrela $S_i \in \mathcal{S}(i)$ com probabilidade $\mathbb{P}(S_i = (i, S)) = x_{(i,S)}$
    Defina o preço de i como P_{S_i}
para cada comprador b **em** B
    se b pertence a alguma das estrelas $\{S_1, \ldots, S_{|I|}\}$
        Seja i um item tal que $b \in S_i$ com P_{S_i} máximo
        Venda o item i para o comprador b
    senão
        O comprador b não recebe nenhum item

A Figura 2.3 apresenta a execução do nosso algoritmo para uma instância do problema.

Lema 2.3.2. StarRounding pode ser implementado de forma a executar em tempo polinomial no tamanho de v.

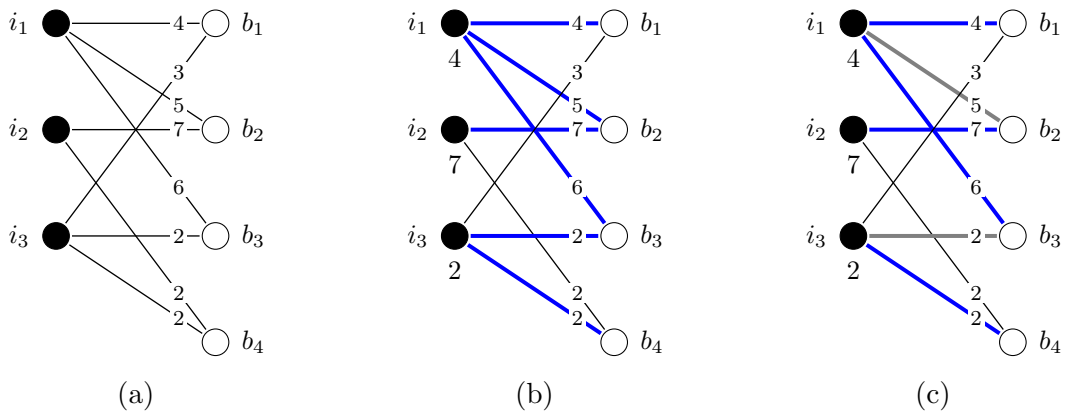


Figura 2.3: *A execução do algoritmo* StarRounding *em uma instância do Problema da Compra Máxima Limitada.* (a) *Uma instância do problema, (b) um possível conjunto de estrelas sorteadas para cada item (indicando pelas arestas grossas azuis) e a precificação encontrada, (c) a solução final encontrada após atribuirmos a cada comprador um dos itens de maior preço dentre aqueles que continham o comprador em sua estrela.*

Demonstração. Em primeiro lugar, na linha 1, pelo Lema 2.3.1, podemos encontrar x em tempo polinomial no tamanho de v tal que o número de variáveis não-nulas de x e o tamanho de sua representação são limitados polinomialmente no tamanho de v. Ademais, considerando o processo utilizado para encontrar x descrito na Seção 1.4, podemos considerar que x é dado apenas por suas entradas não-nulas.

Assim, a linha 3 pode ser implementada para executar em tempo polinomial no tamanho de v e portanto o laço da linha 2 pode ser implementado para executar em tempo polinomial no tamanho de v. É fácil ver que o laço da linha 5 também pode ser implementado para para executar em tempo polinomial. □

Definição. Para uma estrela $T \in \mathcal{S}(i)$, denotamos o item da estrela T por $c(T)$, isto é, $c(T) = i$.

Teorema 2.3.3. StarRounding é uma $\frac{e}{e-1}$-aproximação aleatorizada para o Problema da Compra Máxima Limitada.

Demonstração. Primeiramente, note que a função objetivo da formulação (SF) pode ser reescrita como $\sum_{b \in B} \sum_{(i,S) \in \mathcal{S}: b \in S} P_{(i,S)} x_{(i,S)}$ e que o valor da sua relaxação é uma delimitação superior para o valor de uma solução ótima do problema. Iremos provar que o valor esperado pago pelo comprador b é pelo menos $\frac{e-1}{e} \sum_{(i,S) \in \mathcal{S}: b \in S} P_{(i,S)} x_{(i,S)}$, de onde o resultado seguirá.

Considere um comprador b e 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, assumiremos que $k > 0$.

Denotaremos a ℓ-ésima estrela nessa ordenação simplesmente por ℓ, seu preço por P_ℓ e sua variável primal por x_ℓ. Além disso, definimos $y_\ell = \sum\{x_{\ell'} : \ell' < \ell \text{ e } c(\ell') = c(\ell)\}$. Finalmente, denotamos por E_ℓ o evento no qual a estrela ℓ foi escolhida por StarRounding.

Seja $f(z) = \frac{1-e^{-z}}{z}$. Para algum $1 \leq \ell \leq k$, denotamos $f(\sum_{i=\ell}^{k} x_i)$ simplesmente por f_ℓ. Note que, usando o fato que $1 - z \leq e^{-z}$, concluímos que $f_\ell \leq 1$ para todo $1 \leq \ell \leq k$.

Seja $L(b)$ o lucro que obtivemos a partir do comprador b. Note que, para $1 \leq \ell \leq k$, $\mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_{\ell-1}, E_\ell] = P_\ell$ pois o algoritmo StarRounding aloca $c(\ell)$ (ou outro item com o mesmo preço) para o comprador b já que $c(\ell)$ é um dos itens mais caros que têm b na estrela sorteada. Além disso, note que $\mathbb{P}(E_\ell|\overline{E}_1, \overline{E}_2, \ldots, \overline{E}_{\ell-1}) = \frac{x_\ell}{1-y_\ell}$, pela maneira como o algoritmo StarRounding escolhe as estrelas.

No que segue, usando as observações acima, provaremos, por indução em $k - \ell$, que $\mathbb{E}[L(b)|\overline{E}_1, \overline{E}_2, \ldots, \overline{E}_{\ell-1}] \geq f_\ell \sum_{i=\ell}^{k} P_i x_i$ para todo $1 \leq \ell \leq k$. A partir disso, concluiremos que $\mathbb{E}[L(b)] \geq f_1 \sum_{i=1}^{k} P_i x_i$ e o resultado seguirá, já que $f_1 \geq \frac{e-1}{e}$ (pois f é decrescente).

Se $\ell = k$, então $\mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_{\ell-1}] = P_k x_k/(1-y_k) \geq P_k x_k \geq f_k P_k x_k$. Para $\ell < k$, assuma que o resultado é valido para $\ell + 1$. Temos que

$$\begin{aligned}\mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_{\ell-1}] &= \mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_{\ell-1}, E_\ell]\,\frac{x_\ell}{1-y_\ell} + \mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_{\ell-1}, \overline{E}_\ell]\left(1 - \frac{x_\ell}{1-y_\ell}\right) \\ &= P_\ell\,\frac{x_\ell}{1-y_\ell} + \left(1 - \frac{x_\ell}{1-y_\ell}\right)\mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_\ell].\end{aligned}$$

Usando a hipótese de indução, temos que

$$\begin{aligned}P_\ell\,\frac{x_\ell}{1-y_\ell} + \left(1 - \frac{x_\ell}{1-y_\ell}\right)\mathbb{E}[L(b)|\overline{E}_1, \ldots, \overline{E}_\ell] &\geq P_\ell\,\frac{x_\ell}{1-y_\ell} + \left(1 - \frac{x_\ell}{1-y_\ell}\right) f_{\ell+1}\sum_{i=\ell+1}^{k} P_i x_i \\ &= \frac{x_\ell}{1-y_\ell}\left(P_\ell - f_{\ell+1}\sum_{i=\ell+1}^{k} P_i x_i\right) + f_{\ell+1}\sum_{i=\ell+1}^{k} P_i x_i.\end{aligned}$$

Agora, note que, porque $\sum_{i=\ell+1}^{k} x_i \leq 1$, $P_\ell \geq P_i$ para todo $i > \ell$ e $f_{\ell+1} \leq 1$, temos que $P_\ell \geq P_\ell \sum_{i=\ell+1}^{k} x_i \geq \sum_{i=\ell+1}^{k} P_i x_i \geq f_{\ell+1}\sum_{i=\ell+1}^{k} P_i x_i$. Usando o fato que $\frac{x_\ell}{1-y_\ell} \geq x_\ell$ e que $1 - z \leq e^{-z}$ para todo z, temos que $\frac{x_\ell}{1-y_\ell} \geq x_\ell \geq 1 - e^{-x_\ell}$ e portanto

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

Antes de prosseguirmos, temos que mostrar que $1 - e^{-x_\ell} - e^{-x_\ell} x_\ell f_{\ell+1} \geq 0$. Para $0 < t \leq 1$, seja $h(z) = 1 - e^{-z} - e^{-z} z t$. Note que $h(0) = 0$ e que $h'(z) = e^{-z} + e^{-z} z t - e^{-z} t \geq e^{-z} z t$, isto é, $h(z)$ é não-decrescente para z não-negativo, de onde concluímos que $h(z) \geq 0$ para todo z não-negativo. Combinando isso com o fato que $P_\ell \geq \frac{\sum_{i=\ell}^{k} P_i x_i}{\sum_{i=\ell}^{k} x_i}$ temos que

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

Agora, lembre-se que $f_{\ell+1}\sum_{i=\ell+1}^{k} x_i = 1 - e^{-\sum_{i=\ell+1}^{k} x_i}$, de onde concluímos que

$$\begin{aligned}\left(1 - e^{-x_\ell}\left(1 - f_{\ell+1}\sum_{i=\ell+1}^{k} x_i\right)\right)\frac{\sum_{i=\ell}^{k} P_i x_i}{\sum_{i=\ell}^{k} x_i} &= \left(1 - e^{-x_\ell}\left(1 - 1 + e^{-\sum_{i=\ell+1}^{k} x_i}\right)\right)\frac{\sum_{i=\ell}^{k} P_i x_i}{\sum_{i=\ell}^{k} x_i}\\ &= \left(1 - e^{-\sum_{i=\ell}^{k} x_i}\right)\frac{\sum_{i=\ell}^{k} P_i x_i}{\sum_{i=\ell}^{k} x_i}\\ &= f_\ell \sum_{i=\ell}^{k} P_i x_i.\end{aligned}$$

Isto é, temos que $\mathbb{E}[L(b)|\overline{E}_1, \overline{E}_2, \ldots, \overline{E}_{\ell-1}] \geq f_\ell \sum_{i=\ell}^{k} P_i x_i$, concluindo a prova por indução. A partir disso e do fato que é o mínimo de $f(x) = \frac{1-e^{-x}}{x}$ é atingido em $x = 1$ e $f(1) = \frac{e-1}{e}$, temos que $f(x) \geq \frac{e-1}{e}$ para todo $0 < x \leq 1$, concluímos que

$$\mathbb{E}[L(b)] \geq \frac{1 - e^{-\sum_{i=1}^{k} x_i}}{\sum_{i=1}^{k} x_i}\sum_{i=1}^{k} P_i x_i = f_1 \sum_{i=1}^{k} P_i x_i \geq \frac{e-1}{e}\sum_{i=1}^{k} P_i x_i$$

e o resultado segue. □

Além disso, é fácil provar que a análise é justa, como mostramos no próximo lema.

Lema 2.3.4. Para cada $\varepsilon > 0$, existe uma instância onde o valor de uma solução ótima é 1 e o valor esperado da solução encontrada por StarRounding é menor do que $(e-1)/e + \varepsilon$.

Demonstração. Considere essa simples instância: temos n itens e apenas um comprador b tal que $v_{ib} = 1$ para todo $i \in I$. É fácil ver que uma solução ótima para essa instância tem valor 1. Também é claro que uma solução ótima para a relaxação linear tem valor 1. Uma dessas soluções ótimas é x tal que $x_{(i,\{b\})} = 1/n$ e $x_{(i,\emptyset)} = 1 - 1/n$ para todo item $i \in I$. Note que o comprador b paga 1 se qualquer estrela $(i, \{b\})$ é escolhida e paga 0 (porque que não recebe item) caso contrário. Portanto temos que

$$\mathbb{E}[L(b)] = 1 - \left(1 - \frac{1}{n}\right)^n \xrightarrow{n\to\infty} 1 - \frac{1}{e} = \frac{e-1}{e}$$

e o resultado segue. □

2.4 Desaleatorização do StarRounding

Nessa seção mostramos como obter uma $e/(e-1)$-aproximação determinística para o Problema da Compra Máxima Limitada desaleatorizando o algoritmo StarRounding apresentado na Seção 2.3.

Nosso algoritmo usa esperanças condicionais [ES73, Spe87] para decidir se deve manter uma estrela específica na solução. Para isso, precisamos decidir qual o lucro esperado da solução encontrada por StarRounding sabendo que determinadas estrelas foram escolhidas. Primeiramente apresentamos os algoritmos utilizados e posteriormente provamos os resultados relevantes para esses algoritmos.

Começamos apresentando o algoritmo DeterministicStarRounding que é uma versão desaleatorizada do algoritmo StarRounding. Para isso, consideramos o algoritmo Esperança (apresentado posteriormente) que, dado I, B, v, uma solução x da relaxação linear de (SF) para (I, B, v, C) e um conjunto $\mathbb{X}$ de estrelas onde cada item tem no máximo uma estrela em $\mathbb{X}$ e toda estrela S em $\mathbb{X}$ tem $x_S > 0$, calcula o lucro esperado da solução produzida por StarRounding condicionado que as estrelas em $\mathbb{X}$ sejam escolhidas.

DeterministicStarRounding(I, B, v, C)

Seja x uma solução ótima da relaxação linear de (SF) para (I, B, v, C)
para cada item i **em** I
    Escolha $S_i \in \mathcal{S}(i)$ com $x_{S_i} > 0$ que maximize Esperança$(I, B, v, x, \{S_1, \ldots, S_{i-1}, S_i\})$
    Defina o preço de i como P_{S_i}
para cada comprador b **em** B
    se b pertence a alguma das estrelas $\{S_1, \ldots, S_{|I|}\}$
        Seja i um item tal que $b \in S_i$ com P_{S_i} máximo
        Venda o item i para o comprador b
    senão
        O comprador b não recebe nenhum item

Começamos introduzindo uma notação para representar uma ordenação das estrelas com probabilidade positiva.

Definição. Para uma solução x da relaxação linear de (SF), seja $\mathbb{S} = \{S_1, \ldots, S_N\}$ o conjunto de estrelas tais que $x_S > 0$ ordenadas por P_S de forma não-crescente.

Para calcular as esperanças condicionais, precisaremos representar quais estrelas desejamos considerar como escolhidas. Além disso, precisaremos considerar também que algumas estrelas não podem ser escolhidas (apesar do algoritmo DeterministicStarRounding não utilizar tal artifício). A notação a seguir será útil para representar quais estrelas consideramos como escolhidas e não escolhidas no cálculo das esperanças condicionais.

Definição. Denotamos por $\mathbb{X}$ e por $\overline{\mathbb{X}}$ o conjunto de estrelas que desejamos considerar como escolhidas e o conjunto de estrelas que desejamos considerar como não escolhidas, respectivamente, no cálculo da esperança condicional do lucro. Consideramos que $\mathbb{X} \cup \overline{\mathbb{X}} \subseteq \mathbb{S}$ e $\mathbb{X} \cap \overline{\mathbb{X}} = \emptyset$ e que, se um item i tem uma estrela em $\mathbb{X}$ então toda outra estrela de i em $\mathbb{S}$ está em $\overline{\mathbb{X}}$.

Em particular, note que toda estrela $S \in \mathbb{X} \cup \overline{\mathbb{X}}$ tem probabilidade positiva (de acordo com x) e que um item tem no máximo uma estrela em $\mathbb{X}$, o que condiz com a forma que o algoritmo StarRounding funciona.

A seguir apresentamos o algoritmo Esperança. Para tal, utilizamos o algoritmo Esperança-Comprador que recebe I, B, v, uma solução x da relaxação linear de (SF), $\mathbb{X}$, $\overline{\mathbb{X}}$ e $\mathbb{S}$ como acima e calcula o lucro esperado obtido do comprador b da solução produzida por StarRounding condicionado às estrelas de $\mathbb{X}$ serem escolhidas e as estrelas de $\overline{\mathbb{X}}$ não serem escolhidas.

Esperança$(I, B, v, x, \mathbb{X})$

Seja $\mathbb{S} = \{S_1, \ldots, S_N\}$ uma ordenação não-crescente em P_S das estrelas com $x_S > 0$
Seja $\overline{\mathbb{X}}$ o conjunto de estrelas $S \in \mathbb{S} \setminus \mathbb{X}$ para as quais existe estrela $T \in \mathbb{X}$ com $c(T) = c(S)$
$e = 0$
para cada comprador b **em** B
    $e = e +$ Esperança-Comprador$(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}}, \mathbb{S}, b)$
devolva e

Apesar do algoritmo Esperança construir $\overline{\mathbb{X}}$ a partir de $\mathbb{X}$ simplesmente considerando que se uma estrela S está em $\mathbb{X}$ então toda outra estrela T do mesmo item precisa estar em $\overline{\mathbb{X}}$, o conjunto $\overline{\mathbb{X}}$ terá um papel mais abrangente no algoritmo Esperança-Comprador apresentado abaixo.

ESPERANÇA-COMPRADOR$(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}}, \mathbb{S}, b)$

se toda estrela em $\mathbb{S}$ que contém b pertence a $\overline{\mathbb{X}}$
    devolva 0
senão
    Seja T a primeira estrela em $\mathbb{S} \setminus \overline{\mathbb{X}}$ que contém b
    se T pertence a $\mathbb{X}$
        devolva P_T
    senão
        $q = x_T/(1 - \sum\{x_S : S \in \overline{\mathbb{X}} \text{ e } c(S) = c(T)\})$
        devolva $q \times P_T + (1 - q) \times$ ESPERANÇA-COMPRADOR$(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}} \cup \{T\}, \mathbb{S}, b)$

Como o algoritmo DETERMINISTICSTARROUNDING se baseia no algoritmo ESPERANÇA, que por sua vez se baseia no algoritmo ESPERANÇA-COMPRADOR, começamos apresentando dois lemas relacionados ao algoritmo ESPERANÇA-COMPRADOR, um garantindo sua correção e outro garantindo sua polinomialidade. Antes disso, apresentamos duas definições úteis.

Definição. Denotamos por L a variável aleatória que representa o valor da solução encontrada por STARROUNDING. Além disso, para um comprador b, denotamos por $L(b)$ a variável aleatória que representa o lucro obtido por STARROUNDING a partir do comprador b.

Definição. Seja $S \in \mathcal{S}$. Denotamos por Y_S a variável aleatória binária que indica se o algoritmo STARROUNDING escolhe a estrela S, isto é, $Y_S = 1$ se e somente se STARROUNDING escolhe a estrela S.

Começamos com o resultado de correção para o algoritmo ESPERANÇA-COMPRADOR.

Lema 2.4.1. Dados I, B, v, uma solução x da relaxação linear de (SF), conjuntos $\mathbb{S}$, $\mathbb{X} \subseteq \mathbb{S}$ e $\overline{\mathbb{X}} \subseteq \mathbb{S}$ tais que $\mathbb{X} \cup \overline{\mathbb{X}} \subseteq \mathbb{S}$ e $\mathbb{X} \cap \overline{\mathbb{X}} = \emptyset$ com a propriedade que, se um item i tem uma estrela em $\mathbb{X}$, então toda outra estrela de i em $\mathbb{S}$ está em $\overline{\mathbb{X}}$, juntamente com um comprador b, temos que

$$\text{ESPERANÇA-COMPRADOR}(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}}, \mathbb{S}, b) = \mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}].$$

Demonstração. Em primeiro lugar, suponha que o algoritmo STARROUNDING de fato escolhe as estrelas de $\mathbb{X}$ e não escolhe as estrelas de $\overline{\mathbb{X}}$.

A prova segue por indução em k, o número de estrelas em $\mathbb{S} \setminus \overline{\mathbb{X}}$ que contém b. Note que, se $k = 0$ então toda estrela em $\mathbb{S}$ que contém b pertence $\overline{\mathbb{X}}$ e o comprador b não recebe nenhum item ao final da execução do algoritmo, de onde concluímos que $\mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}] = 0$.

Se $k > 0$, suponha que o resultado é válido para valores menores do que k. Seja T a primeira estrela em $\mathbb{S} \setminus \overline{\mathbb{X}}$ que contém b. Note que toda estrela U que contém b e aparece antes de T em $\mathbb{S}$ pertence a $\overline{\mathbb{X}}$. Assim, se $T \in \mathbb{X}$, temos que $\mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}] = P_T$. Se $T \notin \mathbb{X}$, note que as outras estrelas do item $c(T)$ não pertencem a $\mathbb{X}$ (caso contrário, T pertenceria a $\overline{\mathbb{X}}$) e, portanto, $\mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X} \cup \{T\}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}] = P_T$. Além disso, como escolhemos a estrela de um item independentemente das escolhas das estrelas dos outros itens, temos que $\mathbb{P}(Y_T = 1|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}) = x_T/(1 - \sum\{x_S : S \in \overline{\mathbb{X}} \text{ e } c(S) = c(T)\}$. Seja $q = x_T/(1 - \sum\{x_S : S \in \overline{\mathbb{X}} \text{ e } c(S) = c(T)\}$, temos que

$$\begin{aligned}
&\mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}] \\
&\qquad = q \times \mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X} \cup \{T\}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}] \\
&\qquad\quad + (1 - q) \times \mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}} \cup \{T\}] \\
&\qquad = q \times P_T + (1 - q) \times \text{ESPERANÇA-COMPRADOR}(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}} \cup \{T\}, \mathbb{S}, b),
\end{aligned}$$

onde a última igualdade segue da hipótese de indução. Concluímos que o algoritmo ESPERANÇA-COMPRADOR calcula corretamente $\mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}]$. $\square$

Lema 2.4.2. O algoritmo Esperança-Comprador pode ser implementado de forma a executar em tempo polinomial no tamanho da representação de v e x.

Demonstração. Levando em consideração o fato que os tamanhos das representações de $\mathbb{S}$, $\mathbb{X}$ e $\overline{\mathbb{X}}$ são limitados polinomialmente pelo tamanho da representação de v e x, é fácil ver que as linhas 1 a 8 podem ser executadas em tempo polinomial no tamanho da representação de v e x. Basta então notar que a linha 9 tem uma recursão de cauda, onde o tamanho de $\overline{\mathbb{X}}$ aumenta em 1 a cada passo. Dessa forma, executamos no máximo $|\mathbb{S}\backslash(\mathbb{X}\cup\overline{\mathbb{X}})|$ chamadas recursivas de Esperança-Comprador, de onde o resultado segue. □

Focaremos agora no algoritmo Esperança, mostrando sua correção e polinomialidade.

Lema 2.4.3. Dados I, B, v, uma solução x da relaxação linear de (SF) e um conjunto $\mathbb{X}$ de estrelas onde se $S \in \mathbb{X}$ então $x_S > 0$ e não existe estrela $T \in \mathbb{X}$ tal que $c(T) = c(S)$, temos que

$$\textsc{Esperança}(I, B, v, x, \mathbb{X}) = \mathbb{E}[L|Y_S = 1,\ \forall S \in \mathbb{X}].$$

Demonstração. Sejam $\overline{\mathbb{X}}$ e $\mathbb{S}$ como no algoritmo Esperança. Para todo comprador b, a esperança $\mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}]$ é finita e, pelo Lema 2.4.1, sabemos que $\textsc{Esperança-Comprador}(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}}, \mathbb{S}, b) = \mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}]$. Portanto

$$\begin{aligned}
\textsc{Esperança}(I, B, v, x, \mathbb{X}) &= \sum_{b\in B} \textsc{Esperança-Comprador}(I, B, v, x, \mathbb{X}, \overline{\mathbb{X}}, \mathbb{S}, b) \\
&= \sum_{b\in B} \mathbb{E}[L(b)|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}] \\
&= \mathbb{E}[L|Y_S = 1\,\forall S \in \mathbb{X}, Y_S = 0\,\forall S \in \overline{\mathbb{X}}].
\end{aligned}$$

□

Lema 2.4.4. O algoritmo Esperança pode ser implementado de forma a executar em tempo polinomial no tamanho da representação de v e x.

Demonstração. O resultado segue diretamente do Lema 2.4.2. □

Estamos prontos para provar o principal resultado dessa seção.

Teorema 2.4.5. DeterministicStarRounding é uma $e/(e-1)$-aproximação para o Problema da Compra Máxima Limitada.

Demonstração. Como mencionado na prova do Lema 2.3.2, na linha 1, utilizando o Lema 2.3.1, podemos encontrar x (dado apenas por suas entradas não-nulas) em tempo polinomial no tamanho de v tal que o número de variáveis não-nulas de x e o tamanho de sua representação são limitados polinomialmente no tamanho de v. Pelo Lema 2.4.4, o algoritmo Esperança pode ser implementado de forma a executar em tempo polinomial no tamanho da representação de v e x. Assim, é fácil ver que o algoritmo DeterministicStarRounding pode ser implementado de forma a executar em tempo polinomial no tamanho de v.

É suficiente então provar que a desaleatorização preserva a razão de aproximação. Seja $\mathbb{X}_i = \{S_1, \ldots, S_{i-1}, S_i\}$ para todo $1 \le i \le |I|$. Utilizando o resultado do Lema 2.4.3, temos que $\textsc{Esperança}(I, B, v, x, \mathbb{X}_i) = \mathbb{E}[L|Y_S = 1,\ \forall S \in \mathbb{X}_i]$. Assim, para todo $1 \le i < |I|$, temos que

$$\begin{aligned}
\mathbb{E}[L|Y_S = 1,\ \forall S \in \mathbb{X}_i] &= \sum_{T\in\mathcal{S}(i+1)} \mathbb{E}[L|Y_S = 1,\ \forall S \in \mathbb{X}_i \cup \{T\}]\, \mathbb{P}(Y_T|Y_S = 1,\ \forall S \in \mathbb{X}_i) \\
&\le \mathbb{E}[L|Y_S = 1,\ \forall S \in \mathbb{X}_i \cup \{S_{i+1}\}] \sum_{S\in\mathcal{S}(i+1)} \mathbb{P}(Y_T|Y_S = 1,\ \forall S \in \mathbb{X}_i) \\
&= \mathbb{E}[L|Y_S = 1,\ \forall S \in \mathbb{X}_i \cup \{S_{i+1}\}].
\end{aligned}$$

A partir disso, usando indução e o Teorema 2.3.3, concluímos que

$$\mathbb{E}[L|Y_S = 1, \forall S \in \mathbb{X}_{|I|}] \geq \mathbb{E}[L] \geq \frac{e}{e-1}\mathrm{SOL}(x),$$

onde $\mathrm{SOL}(x)$ indica o valor da solução x da relaxação linear de (SF). Para concluir a prova, note que $\mathbb{E}[L|Y_S = 1, \forall S \in \mathbb{X}_{|I|}]$ é precisamente o valor da solução encontrada por DeterministicStarRounding. □

2.5 Um Algoritmo para Oferta Limitada e Escada de Preços

Nessa seção apresentamos, para cada $\varepsilon > 0$, uma $(2+\varepsilon)$-aproximação para o Problema da Compra Máxima Limitada EP. Usamos as ideias da 4-aproximação para o Problema da Compra Máxima Limitada EP desenvolvida por Aggarwal et al. [AFMZ04], mas de uma forma diferente, para obter uma razão de aproximação melhor.

Sejam α um racional positivo, t um inteiro positivo, v uma valoração indexada por $I \times B$, $V = \max\{v_{ib} : i \in I, b \in B\}$ e, para um inteiro não-negativo k, seja $d_k = V/\alpha^k$.

> O **Problema da Compra Máxima Limitada EP-**(α, t) é uma variante do Problema da Compra Máxima Limitada EP onde cada preço é escolhido do conjunto $\{d_0, d_1, \dots\}$ e cada comprador recebe, para cada r não-negativo, no máximo um item com preço em $\{d_{rt}, d_{rt+1}, \dots, d_{(r+1)t-1}\}$.

Note que, no Problema da Compra Máxima Limitada EP-(α, t), um comprador pode receber mais de um item mas, para cada múltiplo s de t, ele pode receber no máximo um item com preço em $\{d_s, d_{s+1}, \dots, d_{s+t-1}\}$. A Figura 2.4 representa uma possível alocação de itens para um comprador.

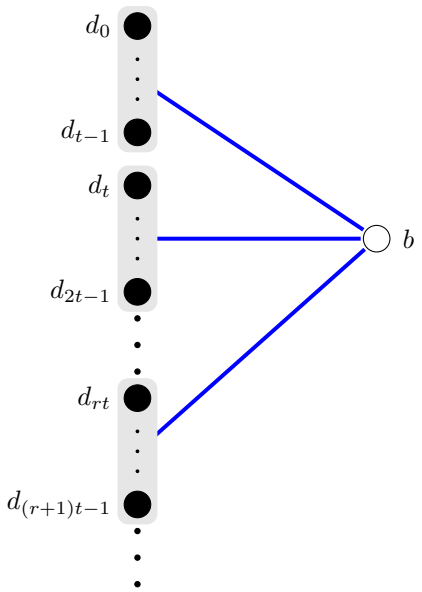


Figura 2.4: *Representação de uma alocação para um comprador no Problema da Compra Máxima Limitada EP-(α, t). O comprador b pode levar no máximo um item com preço d_u e $0 \leq u \leq t-1$, um item com preço d_v e $t \leq v \leq 2t-1$ e assim por diante. Note que podemos ter vários itens com o mesmo preço e que os preços satisfazem a restrição da escada de preços.*

Começamos provando que precisamos considerar apenas um número polinomial de possíveis preços para resolver o Problema da Compra Máxima Limitada EP-(α, t).

Lema 2.5.1. Para toda instância (I, B, v, C) do Problema da Compra Máxima Limitada EP-(α, t), existe uma solução ótima onde o menor preço de um item é d_ℓ, com $\ell = \lceil \log_\alpha V \rceil$.

Demonstração. Pela definição de ℓ, temos que $d_\ell = V/\alpha^{\lceil \log_\alpha V \rceil} \leq V/\alpha^{\log_\alpha V} = 1$. Considere agora uma solução ótima (x, p) do Problema da Compra Máxima Limitada EP-(α, t) e suponha que existe pelo menos um item i de preço $p_i < d_\ell$. Além disso, suponha, sem perda de generalidade, que itens de preço zero não são alocados em (x, p), já que eles não contribuem para o valor da solução. Como v é uma matriz inteira não-negativa, se um item i está alocado, então $p_i \geq 1 \geq d_\ell$ (caso contrário, poderíamos aumentar o preço de i e obter uma solução com um lucro maior), de onde concluímos que os itens com preço menor do que d_ℓ não estão alocados. Portanto, podemos construir outra solução $(x, \tilde{p})$ tal que $\tilde{p}_i = d_\ell$ para cada item i de preço $p_i < d_\ell$ e $\tilde{p}_i = p_i$ para todo item i com $p_i \geq d_\ell$. Note que $\tilde{p}$ respeita a escada de preços e $(x, \tilde{p})$ tem o mesmo valor que (x, p). Concluímos que existe uma solução ótima onde o preço de cada item é pelo menos d_ℓ. □

Provaremos que esse problema pode ser resolvido em tempo polinomial e, posteriormente, que poderá ser usado em um algoritmo de aproximação para o problema original.

Teorema 2.5.2. Para uma instância (I, B, v, C), o Problema da Compra Máxima Limitada EP-(α, t) pode ser resolvido em tempo

$$\mathrm{O}\left(n^{t+3} m (n+m)^4 \frac{\log_\alpha V}{t}\right),$$

que é polinomial no tamanho de v para α e t fixo.

Demonstração. Tome uma instância do Problema da Compra Máxima Limitada EP-(α, t) e considere itens i e j tais que $i \leq j$, e um inteiro não-negativo r. Seja $P(i, j, r)$ o lucro máximo que pode ser atingido vendendo os itens $i, i+1, \ldots, j$ com uma precificação tal que $p_s \in \{d_{rt}, d_{rt+1}, \ldots, d_{(r+1)t-1}\}$ para todo item s em $\{i, \ldots, j\}$ e $p_i \geq p_{i+1} \geq \cdots \geq p_j$ de forma que cada comprador receba no máximo um item em $\{i, \ldots, j\}$. Além disso, seja $F(j, r)$ o lucro máximo que pode ser atingido por uma precificação p onde $p_s \in \{d_0, d_1, \ldots, d_{(r+1)t-1}\}$ para todo item s em $\{i, \ldots, j\}$ considerando apenas itens entre 1 e j e de forma que cada comprador recebe, para cada r' não-negativo, no máximo um item com preço em $\{d_{r't}, d_{r't+1}, \ldots, d_{(r'+1)t-1}\}$. Temos a seguinte recorrência:

$$F(j, r) = \begin{cases} 0, & \text{if } j = 0, \\ P(1, j, r), & \text{if } j > 0 \text{ e } r = 0, \\ \max\limits_{0 \leq i \leq j} \{F(i, r-1) + P(i+1, j, r)\}, & \text{caso contrário.} \end{cases}$$

Seja ℓ como no Lema 2.5.1 e note que $F(|I|, \lceil \ell/t \rceil)$ considera todos os itens e todas as precificação com preços em $\{d_0, \ldots, d_{(\lceil \ell/t \rceil + 1)t-1}\}$. Ademais, note que $(\lceil \ell/t \rceil + 1)t - 1 \geq \lceil \ell/t \rceil t \geq \ell$ e, portanto, $F(|I|, \lceil \ell/t \rceil)$ é o valor de uma solução ótima para o Problema da Compra Máxima Limitada EP-(α, t).

Podemos calcular $P(i, j, r)$ em tempo $\mathrm{O}(n^{t+1} m (n+m)^4)$. Para isso, basta enumerar cada precificação possível (existem não mais do que $(j - i + 2)^t \leq (|I| + 1)^t = \mathrm{O}(n^t)$ tais precificações) e construir um grafo bipartido G com partes $\{i, \ldots, j\}$ e B, onde, para cada item k em $\{i, \ldots, j\}$ e cada comprador $b \in B$, temos uma aresta $\{k, b\} \in E(G)$ de peso p_k se e somente se $p_k \leq v_{kb}$. Basta então encontrar um ρ-emparelhamento de peso máximo em tal grafo, onde cada comprador é emparelhado com no máximo um item e cada item k é emparelhado com no máximo C_k compradores, isto é, para todo $b \in B$, $\rho_b = 1$ e, para todo $k \in \{i, \ldots, j\}$, $\rho_k = C_k$. Utilizando o método húngaro [Kuh55, Mun57], podemos calcular um ρ'-emparelhamento de custo máximo de um grafo bipartido $H = (V_1, V_2, F)$ em tempo $\mathrm{O}((V_1 + V_2)^4 ||\rho'||)$ [Sch03], onde $||\rho'|| = \sum_{v \in V_1 \cup V_2} \rho'_v$. Portanto, podemos calcular um ρ-emparelhamento de custo máximo para G em tempo $\mathrm{O}((n+m)^4 (nm + m)) = \mathrm{O}(nm(n+m)^4)$ já que $||\rho|| = \sum_{b \in B} 1 + \sum_{i \in I} C_i \leq m + nm$.

Note que para calcular uma entrada da matriz F precisamos calcular no máximo n entradas da matriz P. Portanto, podemos calcular uma entrada da matriz F em tempo $\mathrm{O}(n^{t+2} m (n+m)^4)$. Como existem $n \log_\alpha(V)/t$ tais entradas, concluímos que podemos calcular todas as entradas de matriz F em tempo $\mathrm{O}\left(n^{t+3} m (n+m)^4 \log_\alpha(V)/t\right)$. Note que, para α fixo, $\log_\alpha V = \mathrm{O}(\log_2 V)$, e

portanto, $\log_\alpha V$ é polinomial no tamanho da representação da instância (I, B, v, C). Assim, para α e t fixo, o algoritmo descrito acima é polinomial no tamanho de v. □

Iremos agora estabelecer algumas relações envolvendo o valor de uma solução ótima do Problema da Compra Máxima Limitada EP e o valor de uma solução ótima do Problema da Compra Máxima Limitada EP-(α, t).

Lema 2.5.3. Seja OPT o valor de uma solução ótima do Problema da Compra Máxima Limitada EP e OPT$'$ o valor de uma solução ótima para o Problema da Compra Máxima Limitada EP-(α, t). Temos que $\mathrm{OPT}' \geq \mathrm{OPT}/\alpha$.

Demonstração. Considere uma solução ótima do Problema da Compra Máxima Limitada EP. Arredondando para baixo o preço de cada item até o d_k mais próximo, temos uma solução viável do Problema da Compra Máxima Limitada EP-(α, t). Note que $d_{k-1} = \alpha d_k$ para todo inteiro positivo k, isto é, perdemos no máximo um fator de α por causa do arredondamento. Uma solução ótima para o Problema da Compra Máxima Limitada EP-(α, t) tem valor não inferior ao valor dessa solução e, portanto, o resultado segue. □

Lema 2.5.4. Dada uma solução de uma instância (I, B, v, C) para o Problema da Compra Máxima Limitada EP-(α, t) de valor SOL$'$, podemos calcular em tempo polinomial no tamanho de v uma solução do Problema da Compra Máxima Limitada EP de valor pelo menos $\frac{\alpha^t - 1}{\alpha^t - 1 + \alpha^{t-1}}\mathrm{SOL}'$.

Demonstração. Considere uma solução viável do Problema da Compra Máxima Limitada EP-(α, t). Iremos construir uma solução viável para o Problema da Compra Máxima Limitada EP atribuindo a cada comprador b o item mais caro comprado por b (o que pode ser feito em tempo polinomial).

Para um comprador b, seja K o conjunto de inteiros k tais que b comprou um item de preço d_k, e denote por d_i o preço do item mais caro comprado por b. Lembre-se que, para cada r, um comprador pode comprar no máximo um item de preço d_{rt+k} com $0 \leq k < t$. A partir disso concluímos que

$$\sum_{k \in K} d_k \leq d_i + \sum_{k \in K \setminus \{i\}} d_{\lfloor k/t \rfloor t} \leq d_i + \sum_{r \geq 0} d_{i+rt+1}.$$

Note agora que $d_{i+rt+1} = d_i/\alpha^{rt+1}$, de onde concluímos que

$$\sum_{k \in K} d_k \leq d_i + \sum_{r \geq 0} d_{i+rt+1} = d_i \left(1 + \frac{1}{\alpha} \sum_{r \geq 0} \left(\frac{1}{\alpha^t}\right)^r\right) \leq d_i \left(1 + \frac{1}{\alpha}\left(\frac{1}{1 - \frac{1}{\alpha^t}}\right)\right).$$

Note que o lucro obtido a partir de b na solução viável do Problema da Compra Máxima Limitada EP-(α, t) é exatamente $\sum_{k \in K} d_k$ e o lucro obtido a partir de b na solução viável encontrada para o Problema da Compra Máxima Limitada EP é exatamente d_i. Mas concluímos que $d_i \geq \frac{\alpha^t - 1}{\alpha^t - 1 + \alpha^{t-1}} \sum_{k \in K} d_k$, e portanto o resultado segue. □

Combinando os resultados dos Lemas 2.5.3 e 2.5.4, podemos criar um algoritmo de aproximação para o Problema da Compra Máxima Limitada EP.

Teorema 2.5.5. Para todo inteiro positivo t e racional $\alpha > 1$, existe uma $\frac{\alpha(\alpha^t - 1 + \alpha^{t-1})}{\alpha^t - 1}$-aproximação para o Problema da Compra Máxima Limitada EP.

Demonstração. O algoritmo é bem simples: encontre uma solução ótima (x, p) do Problema da Compra Máxima Limitada EP-(α, t) como descrito no Teorema 2.5.2 e devolva $(\tilde{x}, p)$, onde o item alocado ao comprador b em $\tilde{x}$ é o item mais caro alocado a b em (x, p).

Sejam SOL o valor da solução encontrada, OPT o valor da solução ótima do Problema da Compra Máxima Limitada EP e OPT$'$ o valor da solução ótima do Problema da Compra Máxima Limitada EP-(α, t).

Pelo Lema 2.5.3 temos que $\mathrm{OPT}' \geq \mathrm{OPT}/\alpha$ e pelo Lema 2.5.4 temos que $\mathrm{SOL} \geq \frac{\alpha^t-1}{\alpha^t-1+\alpha^{t-1}}\mathrm{OPT}'$. Concluímos que $\mathrm{SOL} \geq \frac{\alpha^t-1}{\alpha(\alpha^t-1+\alpha^{t-1})}\mathrm{OPT}$ de onde obtemos a razão de aproximação desejada. □

Corolário 2.5.6. Para todo $0 < \varepsilon < 1$, existe uma $(2+\varepsilon)$-aproximação para o Problema da Compra Máxima Limitada EP.

Demonstração. Seja $\alpha = 1 + \frac{\varepsilon}{2}$ e $t = \lceil \log_\alpha(\frac{2}{\varepsilon}+1) \rceil$ (note que, porque $\varepsilon < 1$, temos que t é um inteiro positivo). Temos que

$$\frac{\alpha(\alpha^t - 1 + \alpha^{t-1})}{\alpha^t - 1} = 1 + \alpha + \frac{1}{\alpha^t - 1} \leq 1 + \left(1 + \frac{\varepsilon}{2}\right) + \frac{\varepsilon}{2} = 2 + \varepsilon.$$

Isto é, é necessário apenas escolher cuidadosamente α e t e usar o algoritmo do Teorema 2.5.5. □

2.6 O Problema da Compra Máxima Online

Aggarwal et al. [AFMZ04] propuseram uma variante online para o Problema da Compra Máxima Limitada onde, após escolhermos uma precificação, os compradores chegam um a um (em ordem desconhecida) e precisamos escolher o item a ser alocado para tal comprador entre os itens viáveis mais caros ainda disponíveis.

> No **Problema da Compra Máxima Limitada Online**, sabemos I, B, v e C a priori, mas não a ordem de chegada de B (que é revelada de forma online) e escolhemos uma precificação p antes da chegada do primeiro comprador. Quando um comprador chega, precisamos alocar a tal comprador um item que maximize o preço dentro do conjunto de itens viáveis que ainda têm copias restantes. O objetivo é, novamente, maximizar o lucro do leiloeiro.

Ademais, Aggarwal et al. provaram um resultado que limita a perda de lucro por se considerar que os compradores chegam em uma ordem desconhecida. Antes de mostrar tal resultado, apresentamos algumas definições úteis.

Definição. Sejam p uma precificação e σ uma ordem sobre B. Uma alocação x é (p,σ)**-online** se x e p formam uma solução viável para o Problema da Compra Máxima Limitada Online quando a ordem de chegada dos compradores é σ.

Definição. Sejam v uma valoração, p uma precificação e x uma alocação. Se, de acordo com x, todo comprador b que recebe um item, recebe um item viável para b, então a alocação x é **viável**.

Definição. Sejam v uma valoração e p uma precificação. Para uma alocação x viável para p, $L(x,p)$ é o lucro obtido por x considerando que a precificação é p.

Lema 2.6.1 (Aggarwal et al. [AFMZ04])**.** Sejam v uma valoração, p uma precificação e x uma alocação viável para p. Para qualquer ordem σ sobre B e alocação y (p,σ)-online, vale que $L(y,p) \geq \frac{1}{2}L(x,p)$.

Demonstração. Sejam $X(i)$ e $Y(i)$ o conjunto de compradores que recebem i em x e em y, respectivamente. Denotamos por $I_\leq$ o conjunto de itens i para os quais $|X(i)| \leq |Y(i)|$ e denotamos por $I_>$ o conjunto $I \setminus I_\leq$. Note que $\sum_{i\in I_\leq} |X(i)|p_i \leq \sum_{i\in I_\leq} |Y(i)|p_i$. Além disso, para todo item $i \in I_>$, como $|X(i)| > |Y(i)|$ e o número de cópias de i é pelo menos $|X(i)|$, temos que em y existem cópias não alocadas do item i. Portanto, em y, todos os compradores do conjunto $X(i) \setminus Y(i)$ compraram itens com preços pelo menos p_i (caso contrário, até $|X(i)| - |Y(i)|$ desses compradores poderiam ter comprado o item y, pois temos pelo menos tal quantidade de cópias de y disponíveis). Isto é, para

um comprador $b \in X(i) \setminus Y(i)$, $p_i \leq p_{y_b}$, onde y_b indica o item comprado por b em y. Combinando as afirmações acima temos que

$$\begin{aligned}
L(x,p) = \sum_{i \in I} |X(i)| p_i &= \sum_{i \in I_{\leq}} |X(i)| p_i + \sum_{i \in I_{>}} |X(i)| p_i \\
&\leq \sum_{i \in I_{\leq}} |Y(i)| p_i + \sum_{i \in I_{>}} \left(|X(i) \setminus Y(i)| + |X(i) \cap Y(i)| \right) p_i \\
&\leq L(y,p) + \sum_{i \in I_{>}} \sum_{b \in X(i) \setminus Y(i)} p_i + \sum_{i \in I_{>}} \sum_{b \in X(i) \cap Y(i)} p_i \\
&\leq L(y,p) + \sum_{i \in I_{>}} \sum_{b \in X(i) \setminus Y(i)} p_{y_b} + \sum_{i \in I_{>}} \sum_{b \in X(i) \cap Y(i)} p_i \\
&\leq 2L(y,p). \qquad \square
\end{aligned}$$

Note que o resultado é válido mesmo se considerarmos uma escada de preços, já que vale para qualquer precificação.

Em particular, o lema provado por Aggarwal et al. implica que, dada uma solução $(\tilde{x}, \tilde{p})$ tal que $L(\tilde{x}, \tilde{p}) \geq \alpha \mathrm{OPT}$, onde OPT indica o valor de uma solução ótima, temos que, para qualquer ordem σ de chegada do compradores e alocação y construída de forma online utilizando a ordem σ e a precificação $\tilde{p}$, temos que $L(y, \tilde{p}) \geq \frac{1}{2}\mathrm{SOL} \geq \frac{\alpha}{2}\mathrm{OPT}$. Isto é, qualquer β-aproximação para o Problema da Compra Máxima Limitada (com ou sem escada de preços) oferece também um algoritmo 2β-competitivo para o Problema da Compra Máxima Limitada Online proposto por Aggarwal et al. De fato, basta considerar a precificação obtida pelo algoritmo e alocar os compradores arbitrariamente dentro da regra imposta pelo modelo.

Portanto, pela observação acima, temos que o algoritmo apresentado na Seção 2.3 para o Problema da Compra Máxima Limitada pode ser utilizado no projeto de um algoritmo para o Problema da Compra Máxima Limitada Online com razão de competitividade $2e/(e-1)$ e a família de algoritmos (indexada por ε) apresentada na Seção 2.5 para o Problema da Compra Máxima Limitada EP pode ser utilizada no projeto de uma família de algoritmos para o Problema da Compra Máxima Limitada EP Online com razão de competitividade $4 + \varepsilon$.

Capítulo 3

Precificações Livres de Inveja

Nesse capítulo abordamos outro modelo de precificação, o **Problema da Precificação Livre de Inveja**, que foi sugerido por Aggarwal et al. [AFMZ04] e formalizado por Guruswami et al. [GHK+05] para o contexto de leilões combinatórios. Diferentemente do Problema da Compra Máxima considerado no Capítulo 2, no Problema da Precificação Livre de Inveja cada comprador está disposto a comprar um pacote de itens. Sabendo a valoração de cada pacote de itens para cada comprador, o leiloeiro tem que decidir a precificação e a alocação de itens a compradores de forma livre de inveja, isto é, de forma que qualquer comprador esteja pelo menos tão feliz com o pacote de itens atribuído a ele (que pode até mesmo ser vazio) quanto com qualquer outro pacote de itens, considerando a precificação escolhida. Novamente, o objetivo do leiloeiro é maximizar o seu lucro. Obviamente, nessa forma geral, existe um problema na quantidade de informação envolvida, especificamente para as valorações. Portanto é razoável considerar, como fizeram Guruswami et al., casos particulares para evitar esse problema.

Um dos casos particulares considerados por Guruswami et al. foi o Problema da Precificação Livre de Inveja com demandas unitárias[1]. Eles apresentaram uma $(2 \ln m)$-aproximação para esse problema, onde m é o número de compradores, e provaram que esse problema é APX-difícil. Chen e Deng [CD10] mostraram que o Problema da Precificação Livre de Inveja pode ser resolvido em tempo polinomial se cada comprador tem valoração positiva para no máximo dois itens.

Quando consideramos orçamentos uniformes, o Problema da Precificação Livre de Inveja e o Problema da Compra Mínima coincidem. Por isso, os limitantes inferiores apresentados por Briest [Bri08], Chalermsook et al. [CCKK12] e Chalermsook et al. [CLN13], mencionados no Capítulo 2, são válidos também para o Problema da Precificação Livre de Inveja.

Recentemente, Shioda, Tunçel, e Myklebust [STM11] descreveram um modelo levemente mais geral que o definido por Guruswami et al. [GHK+05] e apresentaram uma formulação de Programação Inteira Mista para tal modelo, juntamente com algumas heurística e cortes válidos para a formulação.

Em relação à geração de instâncias aleatórias de leilões para a realização de experimentos computacionais, outro assunto que abordamos nesse capítulo, Leyton-Brown et al. [LBPS00] (veja também [CSS06, Cap. 18]) apresentaram o bem conhecido gerador *Combinatorial Auction Test Suite* (CATS), proposto para criar instâncias de leilões onde os compradores apresentam comportamentos realísticos em seus lances baseando-se em cinco situações reais.

Focando em uma abordagem mais prática, nesse capítulo, apresentamos novas formulações para o Problema da Precificação Livre de Inveja com demanda unitária que podem ser também adaptadas para o modelo descrito por Shioda et al. [STM11] e comparamos essas novas formulações com a apresentada por Shioda et al. através de experimentos computacionais que indicam que as nossas formulações obtém melhores resultados na prática do que a formulação anterior.

Como o CATS produz instâncias onde os compradores estão interessados em comprar um pacote de itens e uma simples adaptação para leilões de demanda unitária não preserva as mo-

[1] Como estamos interessados apenas no caso com demandas unitárias, nesse texto iremos chamar esse problema simplesmente de Problema da Precificação Livre de Inveja.

tivações econômicas utilizadas em sua criação, apresentamos três modelos para gerar instâncias aleatórias diferentes para leilões de demanda unitária. Para os experimentos computacionais, utilizamos seis diferentes conjuntos de instâncias proveniente desses três modelos. Cada um dos três modelos têm uma interpretação econômica interessante e pode ser usado em outros trabalhos em leilões de demanda unitária como por exemplo os experimentos realizados por Myklebust et al. [MST12] e Shioda et al. [STM11]. Acreditamos que esses geradores de instâncias são uma contribuição interessante por si só e, portanto, os disponibilizamos como software livre em https://github.com/schouery/unit-demand-market-models.

O limitante inferior de $\Omega(\ell^{1/2})$ (a não ser que P = NP) de Chalermsook et al. [CCKK12], onde ℓ é um limitante superior no número de valorações não nulas de cada comprador, mostra que o Problema da Precificação Livre de Inveja não está em APX (novamente, a não ser que P = NP). Na busca por uma versão do problema que esteja em APX, consideramos uma variante onde os preços estão restritos a assumirem apenas valores específicos, assim como fizemos para o Problema da Compra Máxima Limitada-EP na Seção 2.5, ou seja, consideramos preços que formam uma série geométrica com razão $1+\varepsilon$ para um $\varepsilon > 0$ fixo. Provamos que qualquer aproximação de razão constante para essa variante é uma aproximação de razão constante (dependendo de ε) para o problema original. Isso implica que essa variante também não está em APX a menos que P = NP. Mesmo assim, a conexão entre esses dois problemas parece interessante, já que um simples arredondamento (como fizemos na Seção 2.5) pode potencialmente resultar em uma grande perda de lucro. Apesar disso, esse resultado garante que a perda (em termos de valores ótimos) está a um fator constante.

3.1 Modelo e Notação

Novamente, denotamos por B o conjunto de compradores e por I o conjunto de itens com oferta ilimitada. Como estamos considerando a versão com demandas unitárias, consideramos apenas alocações onde cada comprador recebe no máximo um item.

Fixe uma valoração v, uma precificação p e uma alocação x. As próximas definições formalizam o conceito de uma alocação livre de inveja.

Definição. A **utilidade** do comprador b, denotada por u_b, é igual a $v_{ib} - p_i$ se o comprador b recebe o item i e é igual a 0 se o comprador b não recebe nenhum item.

Definição. Um comprador b é **livre de inveja** se $u_b \geq 0$ e $u_b \geq v_{ib} - p_i$ para todo item i. Dizemos que x é **livre de inveja** se todo comprador b é livre de inveja.

Em outras palavras, nenhum item daria a algum comprador uma utilidade melhor do que a sua utilidade atual. Estamos agora prontos para enunciar o problema.

> **Problema da Precificação Livre de Inveja**: dado uma valoração v, encontrar uma precificação p e uma alocação livre de inveja x que maximiza o lucro do leiloeiro, isto é, a soma dos preços dos itens vendidos (considerando multiplicidades).

Nesse capítulo representamos uma alocação x por uma matriz binária indexada por $I \times B$ onde $x_{ib} = 1$ se e somente se i está alocado a b.

A Figura 3.1 ilustra uma alocação livre de inveja para a precificação $p = (5, 8, 3)$. O lucro do leiloeiro é, nesse caso, 13. Note que esse não é o melhor lucro possível, pois existe uma alocação livre de inveja para a precificação $p = (6, 6, 3)$ que obtém lucro 21.

Shioda et al. [STM11] consideraram um modelo levemente mais geral para o Problema da Precificação Livre de Inveja. Primeiramente, eles consideraram que cada comprador b tem um valor CS_b que indica a utilidade de b caso ele não receba um item. Isto é, CS_b representa o lucro que o comprador b pode obter comprando um item fora do mercado (de um concorrente, por exemplo).

Eles consideraram também que cada comprador b tem um valor δ_b, tal que se b recebe um item i, então $v_{ib} - p_i \geq v_{i'b} - p_{i'} + \delta_b$ para todo item $i' \neq i$ e $v_{ib} - p_i \geq \mathrm{CS}_b + \delta_b$. A ideia do valor δ_b é se $v_{ib} - p_i \geq v_{i'b} - p_{i'}$ mas $v_{ib} - p_i < v_{i'b} - p_{i'} + \delta_b$, então poderia acontecer do comprador b

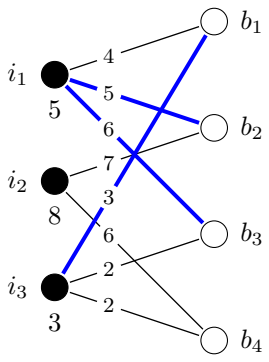


Figura 3.1: *Uma alocação livre de inveja, dada pelas arestas azuis (grossas), para a precificação* $p = (5, 8, 3)$ *(apresentada à esquerda dos respectivos itens).*

comprar o item i' em vez do item i por levar em consideração outros critérios. Assim δ_b obriga um desempate entre os itens para forçar a escolha do comprador ser aquela que desejamos.

Porém, como mostrado por Shioda et al., podemos considerar apenas o problema onde se um comprador b recebe o item i, então $v_{ib} - p_i \geq v_{i'b} - p_{i'} + \delta_b$ para todo item $i' \neq i$, sem exigir que $v_{ib} - p_i \geq \mathrm{CS}_b + \delta_b$. Isso é feito através de um pré-processamento da instância, onde a partir de v, CS e δ, criamos $\tilde{v}$ onde $\tilde{v}_{ib} = \max\{0, v_{ib} - \mathrm{CS}_b - \delta_b\}$ para todo comprador b e todo item i,.

Finalmente, eles consideraram que o conjunto B representa grupos de compradores (cada um com o seu tamanho) em vez de um conjunto de compradores. Para um grupo de compradores b, o valor N_b indica o tamanho de tal grupo. Se o item i é vendido para b, então N_b cópias de i são vendidas, uma para cada comprador no grupo b.

Em nossos experimentos, notamos que o tamanho do grupo dos consumidores não impacta no tempo de execução das formulações. Por isso, consideramos, por simplicidade, que $N_b = 1$ para todo comprador b. Com isso em mente, juntamente com o fato que podemos considerar que $\mathrm{CS}_b = 0$ para todo comprador b e com o fato de que Shioda et al. consideraram apenas instâncias onde $\delta_b = 0$ para todo comprador b, acreditamos termos feito uma comparação justa com o trabalho deles ao focar no Problema da Precificação Livre de Inveja em vez desse problema levemente mais geral.

3.2 Formulações MIP

Primeiramente, vamos introduzir duas notações que serão utilizadas em desigualdades com "M grande" nas formulações apresentadas nessa seção.

Definição. Para um item i, denotamos $\max\{v_{ib} : b \in B\}$ por R_i e, para um comprador b, denotamos $\max\{v_{ib} : i \in I\}$ por S_b.

Note que um item i com preço maior do que R_i não pode ser vendido, já que nenhum comprador pode receber esse item. Além disso, $u_b \leq S_b$ para um comprador b, já que todo item tem preço não negativo.

Começamos apresentando a formulação de Shioda et al. [STM11] já considerando que $\delta_b = 0$ para todo comprador b. Usaremos as seguintes variáveis: uma matriz binária x indexada por $I \times B$ que representa uma alocação, um vetor racional p que representa uma precificação e uma matriz racional $\hat{p}$ indexada por $I \times B$ que representa o preço pago pelo item i para cada comprador b (isto é, $\hat{p}_{ib} = p_i$ se b recebe o item i e $\hat{p}_{ib} = 0$ caso contrário). A formulação, que nomeamos (STM) (as iniciais dos autores), consiste em encontrar x, p, e $\hat{p}$ que

$$
\begin{array}{llrcll}
\text{(STM)} & \text{maximize} & \displaystyle\sum_{b\in B}\sum_{i\in I}\hat{p}_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b\in B \\
& & \displaystyle\sum_{i\in I\setminus\{k\}} (v_{ib}x_{ib}-\hat{p}_{ib}) & \geq & v_{kb}\displaystyle\sum_{i\in I\setminus\{k\}} x_{ib} - p_k, & \forall k\in I, \forall b\in B \quad (1) \\
& & v_{ib}x_{ib}-\hat{p}_{ib} & \geq & 0, & \forall b\in B, \forall i\in I \\
& & \hat{p}_{ib} & \leq & p_i, & \forall b\in B, \forall i\in I \\
& & \hat{p}_{ib} & \geq & p_i - R_i(1-x_{ib}), & \forall b\in B, \forall i\in I \\
& & x_{ib} & \in & \{0,1\}, & \forall b\in B, \forall i\in I \\
& & \hat{p}_{ib} & \geq & 0, & \forall b\in B, \forall i\in I \\
& & p_i & \geq & 0, & \forall i\in I.
\end{array}
$$

A partir da formulação (STM) desenvolvemos uma formulação mais forte mudando as desigualdades (1). Chamamos essa formulação de (A) (da palavra *aprimorada*), apresentada abaixo.

$$
\begin{array}{llrcll}
\text{(A)} & \text{maximize} & \displaystyle\sum_{b\in B}\sum_{i\in I}\hat{p}_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b\in B \quad (2) \\
& & \displaystyle\sum_{i\in I}(v_{ib}x_{ib}-\hat{p}_{ib}) & \geq & v_{kb}-p_k, & \forall k\in I, \forall b\in B \quad (3) \\
& & v_{ib}x_{ib}-\hat{p}_{ib} & \geq & 0, & \forall b\in B, \forall i\in I \quad (4) \\
& & \hat{p}_{ib} & \leq & p_i, & \forall b\in B, \forall i\in I \quad (5) \\
& & \hat{p}_{ib} & \geq & p_i - R_i(1-x_{ib}), & \forall b\in B, \forall i\in I \quad (6) \\
& & x_{ib} & \in & \{0,1\}, & \forall b\in B, \forall i\in I \\
& & \hat{p}_{ib} & \geq & 0, & \forall b\in B, \forall i\in I \\
& & p_i & \geq & 0, & \forall i\in I.
\end{array}
$$

Lema 3.2.1. (A) é uma formulação para o Problema da Precificação Livre de Inveja.

Demonstração. Sejam p uma precificação e x uma alocação livre de inveja para p onde, sem perda de generalidade, $p_i \leq R_i$ para todo item i. Seja $\hat{p}_{ib} = p_i$ se i está alocado a b e $\hat{p}_{ib} = 0$ caso contrário. Primeiramente, vamos mostrar que $(x, p, \hat{p})$ é uma solução viável para (A). As desigualdades (2) e (5) são trivialmente satisfeitas. Agora, note que se $x_{ib} = 1$, então $\sum_{j\in I}(v_{jb}x_{jb} - \hat{p}_{jb}) = v_{ib} - \hat{p}_{ib} = v_{ib} - p_i$ e, como x é uma alocação livre de inveja, $v_{ib} - p_i \geq v_{kb} - p_k$ para todo item k. Por outro lado, se b não recebe item, então $\sum_{j\in I}(v_{jb}x_{jb} - \hat{p}_{jb}) = 0$. Mas como x é uma alocação livre de inveja, $v_{kb} - p_k \leq 0$ para todo item k (caso contrário b teria inveja de k). Concluímos que a desigualdade (3) vale. Para a desigualdade (4), se $x_{ib} = 1$, então como a utilidade de b é necessariamente não negativa e $\hat{p}_{ib} = p_i$, logo $v_{ib}x_{ib} - \hat{p}_{ib} = v_{ib} - p_i = u_b \geq 0$, e se $x_{ib} = 0$, por definição, $\hat{p}_{ib} = 0$ e temos que $v_{ib}x_{ib} - \hat{p}_{ib} \geq 0$. Usando o fato que $p_i \leq R_i$ para todo item i, temos que se $x_{ib} = 0$, então $\hat{p}_{ib} = 0 \geq p_i - R_i = p_i - R_i(1 - x_{ib})$. Se $x_{ib} = 1$, então $\hat{p}_{ib} = p_i = p_i - R_i(1 - x_{ib})$ e, portanto, em ambos os casos (6) é satisfeita. Concluímos que $(x, p, \hat{p})$ é viável em (A) e, além disso, é fácil notar que $\sum_{b\in B}\sum_{i\in I}\hat{p}_{ib}$ é, de fato, o lucro do leiloeiro.

Basta então mostrar que qualquer solução viável $(x, p, \hat{p})$ de (A) é tal que x é uma alocação livre de inveja para p. Por (2), é fácil ver que x é uma alocação de demanda unitária. Além disso, combinando (5) com as desigualdades (4) e (6) respectivamente, temos que $\hat{p}_{ib} = p_i$ se o comprador b recebe o item i e $\hat{p}_{ib} = 0$, caso contrário. Vamos mostrar que os compradores são livres de inveja. Se o comprador b recebe o item i então, para todo item k, temos que $v_{ib} - p_i = \sum_{i\in I}(v_{ib}x_{ib} - \hat{p}_{ib}) \geq v_{kb} - p_k$ e $v_{ib} - p_i = v_{ib}x_{ib} - \hat{p}_{ib} \geq 0$. Se o comprador b não recebe item, então $0 = \sum_{j\in I}(v_{jb}x_{jb} - \hat{p}_{jb}) \geq v_{kb} - p_k$. Isto é, x é uma alocação livre de inveja para p com lucro do leiloeiro precisamente $\sum_{b\in B}\sum_{i\in I}\hat{p}_{ib}$, de onde o resultado segue. □

Uma informação relevante quando comparamos formulações diferentes para o mesmo problema é a qualidade das relaxações lineares de cada formulação. A notação abaixo formaliza tal conceito.

Definição. Para alguma formulação (F) e uma valoração v, denotamos por $\mathrm{RL_F}(v)$ o valor de uma solução ótima da relaxação linear de (F) quando a instância é v.

A seguir relacionamos os valores das relaxações lineares de (A) e (STM).

Lema 3.2.2. Para toda instância v do Problema da Precificação Livre de Inveja, vale que $\mathrm{RL_A}(v) \leq \mathrm{RL_{STM}}(v)$. Além disso, existe uma instância v onde $\mathrm{RL_A}(v) < \mathrm{RL_{STM}}(v)$.

Demonstração. Seja $(x, p, \hat{p})$ uma solução viável para (A). Mostraremos que $(x, p, \hat{p})$ também é viável para (STM). Para obter o resultado desejado, provaremos que, para cada comprador b e item k,

$$\sum_{i \in I \setminus \{k\}} (v_{ib} x_{ib} - \hat{p}_{ib}) \;\geq\; v_{kb} \sum_{i \in I \setminus \{k\}} x_{ib} - p_k.$$

Da viabilidade de $(x, p, \hat{p})$ para (A), temos que $\sum_{i \in I}(v_{ib} x_{ib} - \hat{p}_{ib}) \geq v_{kb} - p_k$, e portanto concluímos que

$$\sum_{i \in I \setminus \{k\}} (v_{ib} x_{ib} - \hat{p}_{ib}) \geq v_{kb} - p_k - v_{kb} x_{kb} + \hat{p}_{kb} \geq v_{kb}(1 - x_{kb}) \;-\; p_k \geq v_{kb} \sum_{i \in I \setminus \{k\}} x_{ib} \;-\; p_k$$

e, portanto, $(x, p, \hat{p})$ uma solução viável para (STM). Além disso, nossos resultados empíricos mostram uma instância onde a desigualdade é estrita. □

Além disso, criamos uma outra formulação que chamamos de (R) (da palavra *relaxada*), que consiste da omissão da desigualdade (5) da formulação (A), pois notamos que a desigualdade (5) é satisfeita por qualquer solução viável da formulação (R). Apresentamos a formulação (R) abaixo.

$$\begin{array}{llrcll}
\text{(R)} & \text{maximize} & \displaystyle\sum_{b \in B} \sum_{i \in I} \hat{p}_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{i \in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & \displaystyle\sum_{i \in I} (v_{ib} x_{ib} - \hat{p}_{ib}) & \geq & v_{kb} - p_k, & \forall k \in I, \forall b \in B \\
& & v_{ib} x_{ib} - \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & p_i - R_i(1 - x_{ib}), & \forall b \in B, \forall i \in I \\
& & x_{ib} & \in & \{0, 1\}, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & p_i & \geq & 0, & \forall i \in I.
\end{array}$$

Lema 3.2.3. Toda solução viável de (R) é uma solução viável de (A).

Demonstração. Basta provar que, dada uma solução viável $(x, p, \hat{p})$ de (R), temos que $\hat{p}_{ib} \leq p_i$ para todo comprador b e item i. Primeiramente, note que para todo item i tal que $x_{ib} = 0$, temos que $\hat{p}_{ib} = 0$. Assim, se $x_{ib} = 0$, temos que $\hat{p}_{ib} = 0$ e $p_i \geq 0$, de onde concluímos que $\hat{p}_{ib} \leq p_i$. Por outro lado, se $x_{ib} = 1$, então como $\sum_{j \in I}(v_{jb} x_{jb} - \hat{p}_{jb}) \geq v_{ib} - p_i$, temos que $v_{ib} x_{ib} - \hat{p}_{ib} \geq v_{ib} - p_i$ e, portanto, $\hat{p}_{ib} \leq p_i$. □

Note que é possível que (R) seja uma versão mais fraca de (A) em termos de garantias da relaxação linear (porém, não observamos esse fato em nossos resultados empíricos).

Lema 3.2.4. Para cada instância v do Problema da Precificação Livre de Inveja, vale que $\mathrm{RL_A}(v) \leq \mathrm{RL_R}(v)$.

Demonstração. O resultado é claro já que o conjunto de desigualdades de (R) é formado a partir de um subconjunto das desigualdades de (A). □

Estudando a formulação (R), desenvolvemos uma nova formulação com variáveis x e p como definidas anteriormente e um vetor racional ℓ indexado por B que representa o lucro obtido a partir do comprador b (isto é, se o comprador b recebe um item i então $\ell_b = p_i$ e $\ell_b = 0$ se b não recebe nenhum item). Isto é, podemos ver ℓ como $\ell_b = \sum_{i \in I} \hat{p}_{ib}$ para todo comprador b. Essa formulação, que nomeamos (L) (da palavra *lucro*), consiste em encontrar x, p, e ℓ que

$$
\begin{array}{llrcll}
\text{(L)} & \text{maximize} & \displaystyle\sum_{b \in B} \ell_b & & & \\
& \text{sujeito a} & \displaystyle\sum_{i \in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & \displaystyle\sum_{i \in I} v_{ib} x_{ib} - \ell_b & \geq & v_{kb} - p_k, & \forall k \in I, \forall b \in B \\
& & \displaystyle\sum_{i \in I} v_{ib} x_{ib} - \ell_b & \geq & 0, & \forall b \in B \\
& & \ell_b & \geq & p_i - R_i(1 - x_{ib}), & \forall b \in B, \forall i \in I \\
& & x_{ib} & \in & \{0, 1\}, & \forall b \in B, \forall i \in I \\
& & p_i & \geq & 0, & \forall i \in I \\
& & \ell_b & \geq & 0, & \forall b \in B.
\end{array}
$$

Lema 3.2.5. (L) é uma formulação para o Problema da Precificação Livre de Inveja.

Demonstração. A prova é muito similar à prova do Lema 3.2.1 e por isso preferimos omiti-la nesse texto. □

Como era de se esperar, (L) é uma versão mais fraca da formulação (R), já que agregamos desigualdades para criar (L).

Lema 3.2.6. Para cada instância v do Problema da Precificação Livre de Inveja, vale que $\mathrm{RL_R}(v) \leq \mathrm{RL_L}(v)$. Além disso, existe uma instância v onde $\mathrm{RL_R}(v) < \mathrm{RL_L}(v)$.

Demonstração. Seja $(x, p, \hat{p})$ uma solução viável da relaxação linear de (R). Provaremos que (x, p, ℓ), onde $\ell_b = \sum_{i \in I} \hat{p}_{ib}$, é uma solução viável da relaxação linear de (R) com mesmo valor que $(x, p, \hat{p})$.

Primeiramente, é claro que, para cada item i, temos que $p_i \geq 0$ e além disso, para cada comprador b, temos que $\sum_{i \in I} x_{ib} \leq 1$. Finalmente, para cada par (i, b) em $I \times B$, temos que $x_{ib} \geq 0$.

Além disso, para um comprador b, usando o fato que $\ell_b = \sum_{i \in I} \hat{p}_{ib}$, é fácil ver que $\ell_b \geq p_i - R_i(1 - x_{ib})$ (porque $\hat{p}_{ib} \geq p_i - R_i(1 - x_{ib})$ para todo item i), $\sum_{i \in I} v_{ib} x_{ib} - \ell_b \geq 0$ (porque $v_{ib} x_{ib} - \hat{p}_{ib} \geq 0$ para todo item i) e $\sum_{i \in I} v_{ib} x_{ib} - \ell_b \geq v_{kb} - p_k$ para todo item k, já que $\sum_{i \in I} (v_{ib} x_{ib} - \hat{p}_{ib}) \geq v_{kb} - p_k$.

Agora, note que (x, p, ℓ) tem o mesmo valor em (L) que $(x, p, \hat{p})$ em (R). Além disso, nossos resultados empíricos mostram uma instância onde a desigualdade é estrita. □

Finalmente, apresentamos a nossa última formulação que foi desenvolvida mudando o foco do preço pago por um comprador para a utilidade do mesmo. Usamos as seguintes variáveis: x e p definidas como antes e um vetor racional u indexado por B que representa as utilidades dos compradores. A formulação, que nomeamos (U) (da palavra *utilidade*), consiste em encontrar x, p, e u que

$$
\begin{array}{llrcll}
\text{(U)} & \text{maximize} & \displaystyle\sum_{i \in I} \sum_{b \in B} v_{ib} x_{ib} - \sum_{b \in B} u_b & & & \\
& \text{sujeito a} & \displaystyle\sum_{i \in I} x_{ib} & \leq & 1, & \forall b \in B \quad (7) \\
& & u_b & \geq & v_{ib} - p_i, & \forall i \in I,\ \forall b \in B \quad (8) \\
& & u_b & \leq & v_{ib} x_{ib} - p_i + (1 - x_{ib})(R_i + S_b), & \forall i \in I,\ \forall b \in B \quad (9) \\
& & u_b & \leq & \displaystyle\sum_{i \in I} v_{ib} x_{ib}, & \forall b \in B \quad (10) \\
& & u_b & \geq & 0, & \forall b \in B \\
& & p_i & \geq & 0, & \forall i \in I \\
& & x_{ib} & \in & \{0, 1\}, & \forall i \in I, \forall b \in B.
\end{array}
$$

Lema 3.2.7. (U) é uma formulação para o Problema da Precificação Livre de Inveja.

Demonstração. Sejam p uma precificação, x uma alocação livre de inveja para p, e u o vetor de utilidades dos compradores. Novamente, supomos que $p_i \leq R_i$ para todo item i. Vamos mostrar que (x, p, u) é viável para (U). Como x é uma alocação livre de inveja para p, temos que (7) e (8) são satisfeitas por definição. Fixe um comprador b e note que $u_b \leq \sum_{i \in I} v_{ib} x_{ib}$ pois se b não recebe item, então $u_b = 0 = \sum_{i \in I} v_{ib} x_{ib}$, e se b recebe o item i, então $u_b = v_{ib} - p_i \leq v_{ib} = \sum_{i \in I} v_{ib} x_{ib}$. Portanto, concluímos que a desigualdade (10) é válida. Por fim, se b recebe o item i, então $u_b = v_{ib} - p_i = v_{ib} x_{ib} - p_i + (1 - x_{ib})(R_i + S_b)$, e se b não recebe o item i, então $v_{ib} x_{ib} - p_i + (1 - x_{ib})(R_i + S_b) = -p_i + R_i + S_b$ e, como $p_i \leq R_i$ e $u_b \leq S_b$, temos que $u_b \leq -p_i + R_i + S_b$.

Agora, vamos mostrar que se (x, p, u) é uma solução viável para (U), então x é uma alocação livre de inveja para p. Por (7), segue que x é uma alocação de demanda unitária. Considere então um comprador b. Se b recebe o item i, então, usando as desigualdades (8) e (9), temos que $u_b = v_{ib} - p_i$. Se b não recebe um item, então por (10) temos que $u_b = 0$. Além disso, $u_b \geq 0$ e por (8) temos que, para todo item k, $u_b \geq v_{kb} - p_k$. Portanto, b é livre de inveja e concluímos que x é uma alocação livre de inveja para p.

Para completar a nossa prova, basta mostrar que o lucro do leiloeiro é igual a $\sum_{i \in I} \sum_{b \in B} v_{ib} x_{ib} - \sum_{b \in B} u_b$. Seja $M \subseteq I \times B$ o conjunto de pares (i, b) tais que, em x, o item i está alocado para o comprador b. Temos que

$$\sum_{i \in I} \sum_{b \in B} v_{ib} x_{ib} - \sum_{b \in B} u_b = \sum_{(i,b) \in M} v_{ib} - \sum_{b \in B} u_b = \sum_{(i,b) \in M} v_{ib} - \sum_{(i,b) \in M} (v_{ib} - p_i) = \sum_{(i,b) \in M} p_i.$$

O resultado segue do fato que $\sum_{(i,b) \in M} p_i$ é precisamente o lucro do leiloeiro. □

Lema 3.2.8. Para cada instância v do Problema da Precificação Livre de Inveja, vale que $\mathrm{RL_L}(v) \leq \mathrm{RL_U}(v)$. Além disso, existe uma instância v onde $\mathrm{RL_L}(v) < \mathrm{RL_U}(v)$.

Demonstração. Seja (x, p, ℓ) uma solução viável da relaxação linear de (L). Provaremos que (x, p, u), onde $u_b = \sum_{i \in I} v_{ib} x_{ib} - \ell_b$, é uma solução viável para a relaxação linear de (U) com o mesmo valor de (x, p, ℓ).

Como (x, p, ℓ) é viável para (L), temos que $\sum_{i \in I} x_{ib} \leq 1$ para todo comprador b, $p_i \geq 0$ para todo item i e, para todo par $(i, b) \in I \times B$, temos que $x_{ib} \geq 0$.

Usando o fato que $\ell_b \geq 0$ para todo comprador b, concluímos que $\sum_{i \in I} v_{ib} x_{ib} - u_b \geq 0$, e pelo fato que $\sum_{i \in I} v_{ib} x_{ib} - \ell_b \geq 0$, concluímos que $u_b \geq 0$. Além disso, temos que $u_b = \sum_{i \in I} v_{ib} x_{ib} - \ell_b \geq v_{kb} - p_k$ para todo item k.

Agora, note que, como $\sum_{i \in I} x_{ib} \leq 1$, para um comprador b, temos que, para cada item i,

$$\begin{aligned} u_b = \sum_{i' \in I} v_{i'b} x_{i'b} - \ell_b &\leq v_{ib} x_{ib} + \sum_{i' \in I \setminus \{i\}} v_{i'b} x_{i'b} - p_i + R_i(1 - x_{ib}) \\ &\leq v_{ib} x_{ib} - p_i + \sum_{i' \in I \setminus \{i\}} S_b x_{i'b} + R_i(1 - x_{ib}) \\ &\leq v_{ib} x_{ib} - p_i + S_b(1 - x_{ib}) + R_i(1 - x_{ib}) \\ &= v_{ib} x_{ib} - p_i + (S_b + R_i)(1 - x_{ib}), \end{aligned}$$

provando que (x, p, u) é viável em (U). É fácil ver que (x, p, u) tem o mesmo valor em (U) que (x, p, ℓ) tem em (L). Além disso, nossos resultados empíricos mostram uma instância onde a desigualdade é estrita. □

Corolário 3.2.9. Para cada instância v do Problema da Precificação Livre de Inveja, vale que $\mathrm{RL_A}(v) \leq \mathrm{RL_{STM}}(v)$ e $\mathrm{RL_A}(v) \leq \mathrm{RL_R}(v) \leq \mathrm{RL_L}(v) \leq \mathrm{RL_U}(v)$. Além disso, existe instâncias onde $\mathrm{RL_A}(v) < \mathrm{RL_{STM}}(v)$ e $\mathrm{RL_A}(v) = \mathrm{RL_R}(v) < \mathrm{RL_L}(v) < \mathrm{RL_U}(v)$.

Demonstração. A primeira parte da prova é dada pelos Lemas 3.2.2, 3.2.4, 3.2.6, e 3.2.8. A segunda parte provêm dos nossos resultados empíricos. Chamamos a atenção para o fato que existe uma instância onde todas as desigualdades são estritas (exceto para a relação entre $\mathrm{RL_A}(v)$ e $\mathrm{RL_R}(v)$). Os dados podem ser encontrados em https://github.com/schouery/unit-demand-market-models. □

No Apêndice A, mostramos como adaptar as formulações apresentadas nessa seção para a generalização do Problema da Precificação Livre de Inveja proposta por Shioda et al. [STM11].

Antes de apresentar e discutir as comparações experimentais dessas formulações, descrevemos os geradores de instâncias que usamos nos experimentos computacionais.

3.3 Geradores de Testes para Leilões de Demanda Unitária

Nessa seção, apresentamos três modelos para gerar instâncias aleatórias para leilões de demanda unitária, cada um deles com uma interpretação econômica interessante.

Nesses três modelos, começamos gerando um grafo bipartido onde uma parte representa os itens e a outra representa os compradores e uma aresta representa que um comprador tem uma valoração positiva para o item correspondente. Posteriormente, decidimos qual o valor de cada aresta desse grafo para obter uma instância para leilões de demanda unitária.

3.3.1 Modelo das Características

No Modelo das Características, consideramos que cada item tem um conjunto de características que são relevantes para determinar se um comprador tem interesse em tal item. Por exemplo, imagine que nossos itens são carros e, por simplicidade, cada carro tem duas características: cor e modelo do motor. Nesse modelo, cada comprador tem interesse em um conjunto de cores e em um conjunto de modelos de motor e, potencialmente, compraria qualquer carro que tiver umas das cores e um dos motores em que tal comprador está interessado.

A ideia desse modelo é criar um grafo onde cada item tem um perfil de características e cada comprador deseja algumas dessas características. Dessa forma, dois itens que têm o mesmo perfil são desejados pelos mesmos compradores e cada comprador deseja itens que não são muito diferentes.

Uma instância do nosso problema, isto é, uma valoração, será representada por um grafo bipartido com pesos como descrito a seguir.

Sejam n o número de itens e m o número de compradores. Denotamos por c o número de características de um item, o o número de opções para cada característica e p o número de opções preferidas por um comprador para cada característica. O conjunto de vértices é $I \cup B$, onde $I = [n]$ e $B = [m]$ e o conjunto de arestas é E como descrito abaixo.

Para cada item i, temos um vetor de tamanho c (as características desse item) onde cada entrada é escolhida independente e uniformemente ao acaso do conjunto $\{1, \ldots, o\}$. Para cada comprador b, construímos a matriz $A \in \{0,1\}^{c \times p}$ onde, para cada linha, escolhemos independente e uniformemente ao acaso p posições para serem definidas como 1, e $o - p$ posições para serem definidas como 0. Para um comprador b e um item i, temos que $(i, b) \in E$ se e somente se a característica do item i coincide com as preferências do comprador b, isto é, se a característica k do item i tem valor v então $A_{kv} = 1$ onde A é a matriz que geramos para o comprador b. Veja um exemplo de tais características e preferências na Figura 3.2.

Para a definição da valoração, é natural pensar que um item tem um valor de mercado e que as valorações para esse item (dos compradores interessados) estão concentradas próximas desse valor. Isto é, as valorações para um item não devem diferir muito entre si e valorações que estão longe desse valor de mercado são raras. Os pesos nas arestas são descritos a seguir.

Sejam ℓ o valor mínimo de mercado de um item, h o valor máximo de mercado de um item, e d a porcentagem de desvio usada. Para cada item i, definimos $\bar{p}_i$ como o preço de mercado do item i, escolhidos independente e uniformemente ao acaso do intervalo $[\ell, h]$. Para cada (i, b) em E, escolhemos v_{ib} de $1 + \mathcal{N}(\bar{p}_i, (\bar{p}_i d)^2)$, onde $\mathcal{N}(\mu, \sigma^2)$ denota a distribuição Gaussiana com média μ e desvio padrão σ.

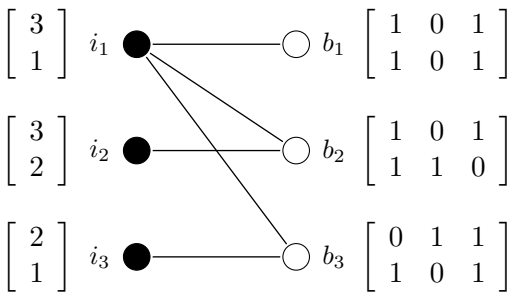


Figura 3.2: *A construção do grafo do Modelo das Características. Consideramos duas características, cada uma com três opções. Cada comprador deseja itens que tenham uma de duas opções (dadas por suas matrizes) em cada característica.*

3.3.2 Modelo da Vizinhança

No Modelo da Vizinhança, consideramos que tanto os itens quanto os compradores estão localizados no plano e que cada comprador tem interesse por itens que são próximos de sua localidade. Imagine, por exemplo, que cada comprador tem um local de preferência para morar em uma cidade e que os compradores estão comprando casas. Nesse modelo, um determinado comprador não comprará uma casa muito distante do seu ponto de preferência e dará mais valor para casas mais próximas se comparadas com casas mais distantes.

Como esse modelo seleciona as valorações baseado em uma relação geométrica, adaptamos o processo aleatório primeiramente introduzido por Gilbert [Gil61] para gerar Grafos Geométricos Aleatórios.

A ideia é distribuir os itens e os compradores em um quadrado 1×1 em $\mathbb{Q}^2$ e definir as valorações de acordo com a distâncias entre um comprador e um item.

Denotamos o quadro 1×1 em $\mathbb{Q}^2$ por W e denotamos o disco com centro na posição (x, y) e raio r por $D(x, y, r)$.

Sejam n o número de itens, m o número de compradores e r o raio. Para construir a valoração, construímos um grafo $G = (I, B, E)$ tal que $I = [n]$ e $B = [m]$. Primeiramente, atribuímos um ponto em W independente e uniformemente ao acaso para cada item e para cada comprador. Isto é, o item i e o comprador b são representados em W pelos pontos (x_i, y_i) e (x_b, y_b) em $\mathbb{Q}^2$, respectivamente. Se $(x_i, y_i) \in D(x_b, y_b, r)$ para um item i e um comprador b, então existe a aresta (i, b) em E, isto é, b está interessando em i se a distância entre os dois é no máximo r. A Figura 3.3 mostra um exemplo dessa construção.

Finalmente, para definir a valoração, consideramos dois valores h e M e para cada comprador b, escolhemos uniformemente ao acaso um multiplicador k_b em $[1, h]$ e se $(i, b) \in E$, então $v_{ib} = 1 + Mk_b/d(i, b)$ onde $d(i, b)$ denota a distância (Euclidiana) entre i e b.

Dessa forma, um comprador dá valor positivo apenas para itens suficientemente próximos de sua localização e o valor diminui conforme a distância aumenta. Dois compradores que estão à mesma distância de um item ainda podem avaliar o item diferentemente porque eles podem ter diferentes multiplicadores k_b.

3.3.3 Modelo da Popularidade

Em um mercado onde os itens são parecidos, as preferências dos compradores podem se basear em quão popular é um determinado item. No nosso caso, medimos a popularidade de um item pela quantidade de compradores que o desejam. Para isso, consideramos um processo onde um item popular fica cada vez mais popular. Além disso, consideramos que cada item tem uma qualidade e que o preço de mercado do item é diretamente proporcional à sua qualidade e inversamente proporcional à sua popularidade (já que uma empresa pode vender esse item mais barato se ele for popular).

Sejam n o número de itens, m o número de compradores e e o número de arestas desejadas no

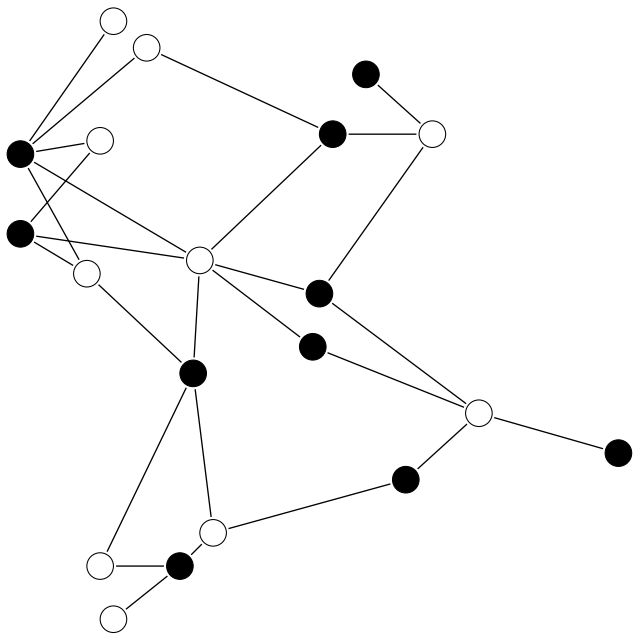


Figura 3.3: *A construção do grafo utilizado no Modelo da Vizinhança com* 10 *itens,* 10 *compradores e raio* 0.33.

grafo. Construímos um grafo bipartido como descrito a seguir. Começamos com um grafo bipartido vazio com partes I e B tais que $I = [n]$ e $B = [m]$, e adicionamos arestas aleatoriamente (como descrito abaixo) até que o grafo tenha e arestas.

Para adicionar uma aresta, escolhemos um comprador uniformemente ao acaso e escolhemos um item priorizando os itens de maior grau. Isto é, um item i de grau (atual) d_i tem peso $d_i + 1$ (somamos 1 para permitir que itens de grau zero também sejam sorteados). Dessa forma, itens que são "populares", quando adicionamos uma aresta, têm uma chance maior de serem escolhidos do que itens "impopulares". O resultado final é um grafo com um pequeno número de itens de grau grande e um grande número de itens de grau pequeno. Esse processo é semelhante ao processo de *Preferential Attachment*, utilizado por Barabási e Albert [BA99] para gerar grafos com propriedades que aparecem na Internet e em grafos que surgem da Biologia como, por exemplo, as interações entre células do cérebro.

Utilizamos também a informação do grau (final) de um item para decidir as valorações. Sejam Q a qualidade máxima de um item e d a porcentagem de desvio. Assim como no Modelo das Características, um item i tem preço de mercado $\bar{p}_i$, e escolhemos a valoração v_{ib} de acordo com $1 + \mathcal{N}(\bar{p}_i, (\bar{p}_i d)^2)$. Mas, diferentemente do Modelo das Características, não escolhemos o valor de mercado uniformemente ao acaso em um intervalo. Para cada item i, escolhemos uma qualidade aleatória q_i em $(0, Q]$ e definimos $\bar{p}_i$ como q_i/d_i. Dessa forma, o preço de mercado de um item cresce com a sua qualidade e decresce com a sua popularidade. Isto é, um item é muito interessante se é "barato" ou se tem boa qualidade. Veja a Figura 3.4.

3.4 Resultados Empíricos

Nessa seção apresentamos alguns resultados empíricos envolvendo as formulações apresentadas na Seção 3.2. Em nossos experimentos, as novas formulações produziram resultados melhores que a formulação (STM), que é a formulação previamente conhecida na literatura. Em particular, a formulação (R) foi a melhor para instâncias pequenas e a formulação (U) foi a melhor para instâncias grandes (obtendo um gap pequeno).

Para os nossos testes, utilizamos um computador com dois processadores Intel Xeon E5620 2,40 GHz, com 64GB de RAM rodando o Gentoo Linux 64bit. Os MIPs foram resolvidos usando CPLEX 12.1.0 em modo *single-thread*, com memória limitada a 4GB. Os testes foram executados

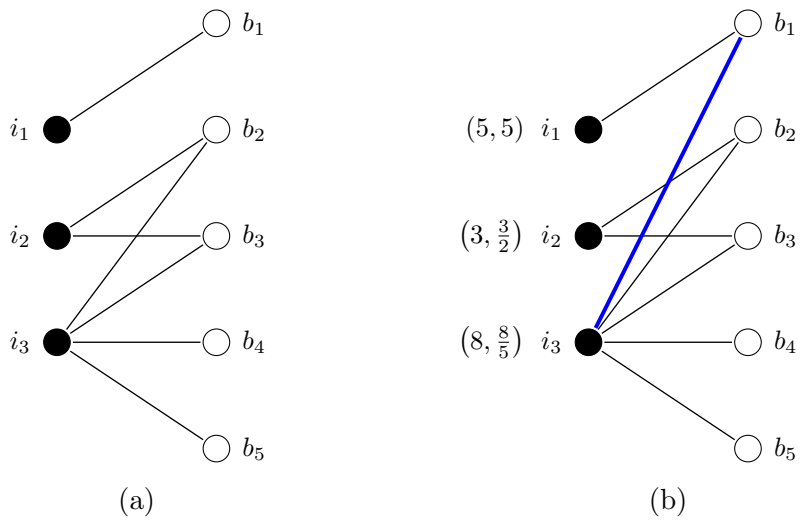


Figura 3.4: *De* (a) *para* (b)*, a oitava aresta* (i_3, b_1) *foi adicionada. Em* (a)*, as probabilidades de escolher os itens* i_1*,* i_2 *e* i_3 *são* 2/10*,* 3/10 *e* 5/10*, respectivamente. O par em parênteses em* (b) *representa* $(q_i, \bar{p}_i)$*.*

usando um limite de uma hora de tempo de CPU.

Geramos seis conjuntos de testes, dois para cada um dos nossos três modelos: um com 20 instâncias de 50, 100, 150, 200, 250, e 300 itens e a mesma quantidade de compradores (que chamamos de "instâncias pequenas") e outro com 20 instâncias de 500, 1000, 1500, 2000, 2500, e 3000 itens e a mesma quantidade de compradores (que chamamos de "instâncias grandes"). A seguir descrevemos os parâmetros utilizados nos modelos e denotamos por n o número de itens e compradores.

Para o Modelo das Características, escolhemos o número de opções para uma característica como $o = 8$, o número de opções preferidas por um comprador para qualquer característica como $p = 7$, o número de características de um item como $\lceil \log(8/n)/\log(p/o) \rceil$. Assim, o grau médio de um comprador está entre 7 e 8. Além disso, escolhemos o valor mínimo de mercado de um item como $\ell = 1$, o valor máximo de mercado de um item como $h = 100$, e a porcentagem de desvio como $d = 0{,}25$. Para o Modelo da Vizinhança, escolhemos o multiplicador máximo de um comprador como $h = 3$, o raio como $\sqrt{8/(n\pi)}$ (de forma que o grau médio de um comprador seja próximo de 8) e o fator de escala $M = 10$. Finalmente, para o Modelo da Popularidade, escolhemos o número de arestas como $e = 8n$, a qualidade máxima de um item como $Q = 200$ e a porcentagem de desvio como $d = 0{,}25$.

Note que todas as nossas instâncias são esparsas, já que acreditamos que, na prática, instâncias densas sejam raras. Isto é, esperamos que, em um mercado com muitos itens, cada comprador tenha, usualmente, valoração positiva para apenas alguns itens.

Em nossos experimentos, o CPLEX teve uma melhor performance com a formulação (R) do que com qualquer outra formulação para instâncias pequenas. Primeiramente, usando a formulação (R), o CPLEX resolveu mais instâncias pequenas do que as outras formulações. Por exemplo, com (STM), o CPLEX não pode resolver (em uma hora) qualquer instância com 100 itens ou mais no Modelo das Características. Mas, com (R), o CPLEX pode resolver todas as instâncias com 100 itens, 13 instâncias com 150 itens, 8 instâncias com 200 itens e 1 instância com 250 itens. As Figuras 3.8, 3.9 e 3.10 a seguir e a Tabela B.1 no Apêndice B apresentam o número de soluções encontradas por cada formulação para instâncias pequenas dos três modelos. Como nenhuma instância com 300 itens foi resolvida, omitimos essa informação das figuras.

Além disso, a Figuras B.1 e a Tabela B.2 no Apêndice B mostram que (STM) e (L) apresentaram um gap final médio muito pior que os obtidos por (U), (A) e (R) para as instâncias pequenas que não foram resolvidas em uma hora (pelo menos o dobro do gap final médio dessas formulações).

Mesmo com a formulação (R) sendo melhor do que as outras, notamos que as formulações (U)

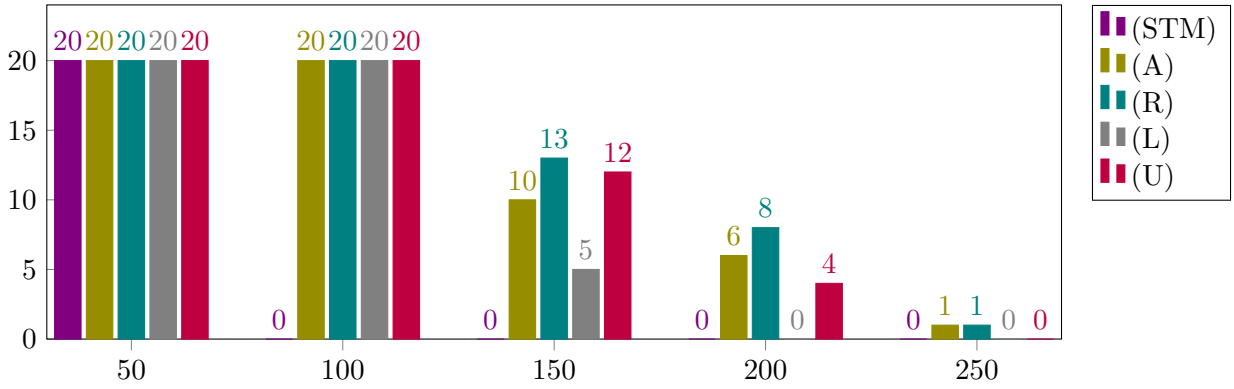


Figura 3.5: *Número de soluções encontradas para o Modelo das Características.*

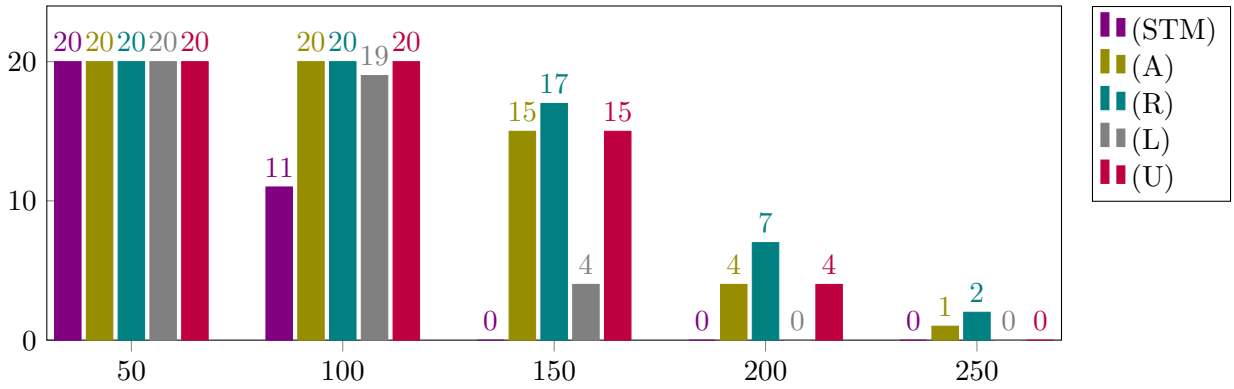


Figura 3.6: *Número de soluções encontradas para o Modelo da Vizinhança.*

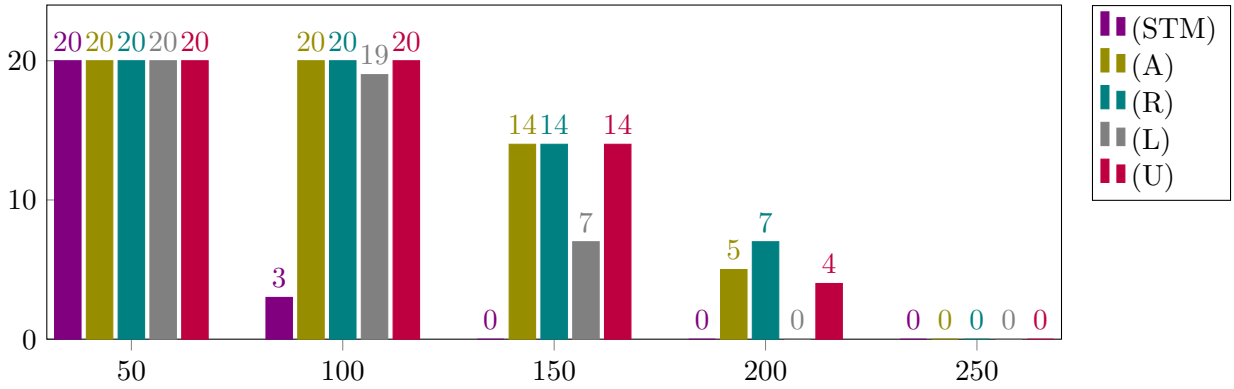


Figura 3.7: *Número de soluções encontradas para o Modelo da Popularidade.*

e (A) não estão muito atrás da formulação (R) como é possível observar nas Figuras 3.5, 3.6 e 3.7 (nessa seção) e B.1 (no Apêndice B) e a Tabela B.2 (também no Apêndice B).

Para instâncias grandes, quando analisamos o gap final médio, concluímos que a formulação (U) é melhor do que as outras. De fato, (U) foi capaz de manter um gap final médio pequeno mesmo para instâncias com 3000 itens (2% para o Modelo das Características, 3% para o Modelo da Vizinhança e 17% para o Modelo da Popularidade). As Figuras 3.8, 3.9 e 3.10 (nessa seção) e a Tabela B.3 (no Apêndice B) apresentam essa informação em mais detalhes.

Acreditamos que esse resultado é impressionante porque, teoricamente (como provado no Teorema 3.2.9), (U) é a mais fraca das novas formulações (note que não temos um resultado teórico que compare (U) e (STM)). Apesar disso, (U) é uma formulação pequena quando comparada com (STM), (A) e (R) e, por causa disso, a relaxação linear pode ser computada mais rapidamente para essa formulação (e também para a formulação (L)) como é possível ver na Figura B.2 e na

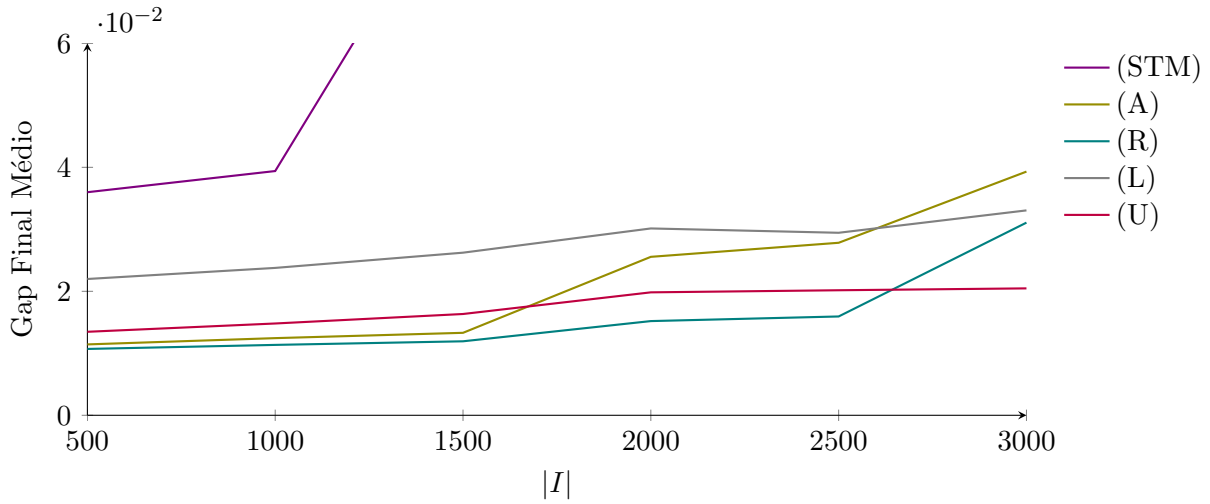


Figura 3.8: *Gap final médio para instâncias grandes para o Modelo das Características.*

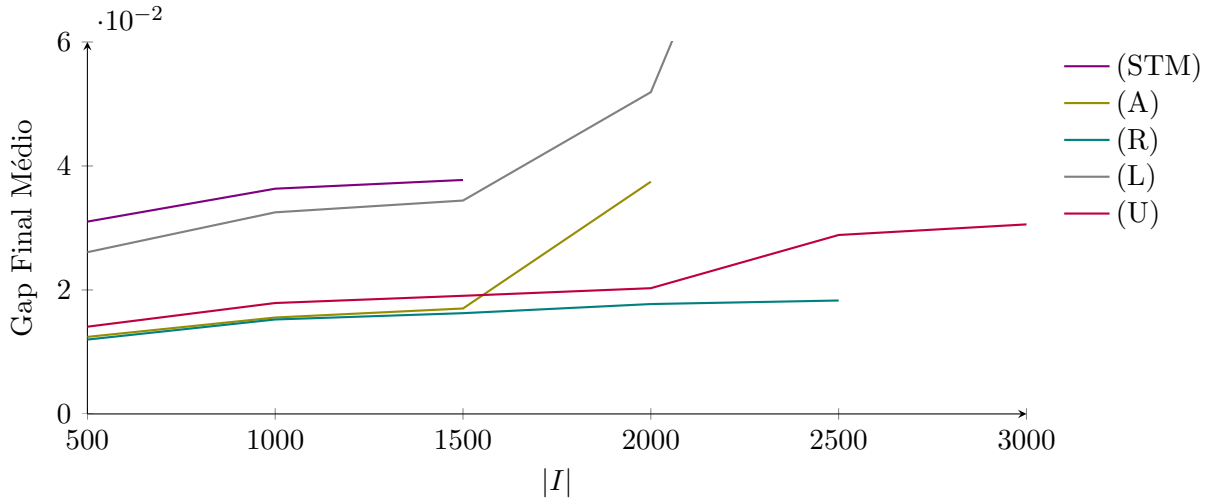


Figura 3.9: *Gap final médio para instâncias grandes para o Modelo da Vizinhança.*

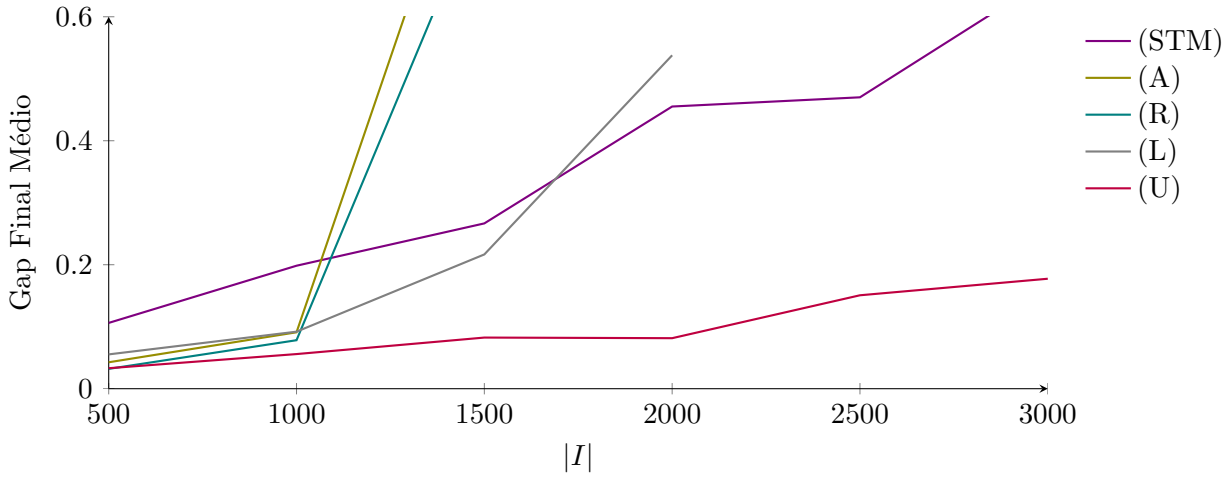


Figura 3.10: *Gap final médio para instâncias grandes para o Modelo da Popularidade.*

Tabela B.4 no Apêndice B. Como (L) é uma formulação mais forte do que (U) e elas têm o mesmo tamanho, acreditamos que a formulação (U) superou a formulação (L) por causa das heurísticas usadas pelo CPLEX para encontrar boas soluções inteiras que, aparentemente, foi mais eficiente

para (U) do que para (L).

3.5 A Dificuldade de Aproximar Preços Geométricos

O resultado de Chalermsook et al. [CCKK12] mostra quão difícil é aproximar o Problema da Precificação Livre de Inveja. Enquanto buscávamos uma variante do problema para a qual pudéssemos projetar um algoritmo polinomial ou uma aproximação de razão constante e inspirados por [AFMZ04], consideramos o caso onde os preços são restritos a elementos de uma série geométrica. Algo semelhante ao que fizemos na Seção 2.5.

Note que isso reduz drasticamente o espaço de preços viáveis e poderia, potencialmente, levar a soluções com valor muito baixo se comparada com o valor de uma solução ótima. Esse não é o caso. Como provamos abaixo, o valor de uma solução ótima para essa variante está a um fator constante do valor de uma solução ótima para o problema original. Antes de apresentar o resultado, introduzimos algumas notações úteis.

Definição. Para uma valoração v, um racional positivo ε e um inteiro positivo k, seja $d_k = \max\{v_{ib} : i \in I, b \in B\}/(1+\varepsilon)^k$. Denotamos por $\mathcal{D}_{1+\varepsilon}$ o conjunto $\{d_0, d_1, \ldots\}$.

Definição. Para um real positivo y e um conjunto $\mathcal{D}_{1+\varepsilon}$, seja $\lfloor y \rfloor_{1+\varepsilon} = d_r$ onde r é um inteiro tal que $d_r \leq y < d_{r-1}$.

Dada uma valoração v e uma precificação p, é fácil computar uma alocação livre de inveja x_p que maximiza o lucro do leiloeiro. De fato, para cada comprador b, atribua a b um item i que maximize $v_{ib} - p_i$ (se esse valor for não-negativo), resolvendo empates escolhendo o item de maior preço (se o empate persistir, existe mais de uma tal alocação).

Definição. Seja $\mathrm{SOL}(p)$ o lucro do leiloeiro para a precificação p e x_p.

Primeiramente consideramos $\varepsilon = 1$, isto é, que os preços estão restritos a $\mathcal{D}_2$. Mostraremos que o lucro máximo atingido por uma precificação restrita a $\mathcal{D}_2$ é pelo menos $1/4$ do valor de uma solução ótima para o Problema da Precificação Livre de Inveja.

Teorema 3.5.1. Para toda precificação p, existe uma precificação $\tilde{p}$ consistindo de preços apenas em $\mathcal{D}_2$ tal que $\mathrm{SOL}(\tilde{p}) \geq \mathrm{SOL}(p)/4$.

Demonstração. Obtemos $\tilde{p}$ arredondando os componentes do vetor $2p/3$. Isto é, $\tilde{p}_i = \lfloor 2p_i/3 \rfloor_2$ para todo item i. Primeiramente, note que $p_i/3 \leq \tilde{p}_i \leq 2p_i/3$. Vamos mostrar que $\mathrm{SOL}(\tilde{p}) \geq \mathrm{SOL}(p)/4$.

Como $\tilde{p}_i \leq p_i$ para todo item i, cada comprador que recebe um item em x_p também recebe um item em $x_{\tilde{p}}$. Agora, suponha que o comprador b recebe um item i em x_p e um item k em $x_{\tilde{p}}$. Se $\tilde{p}_k \geq \tilde{p}_i$, então o lucro obtido a partir de b é pelo menos $\tilde{p}_i \geq p_i/3$. Portanto, de agora em diante, assumimos que $\tilde{p}_k < \tilde{p}_i$.

Como b recebe i e não k em x_p, temos que $v_{ib} - p_i \geq v_{kb} - p_k$. Além disso, se b recebe k e não i em $x_{\tilde{p}}$, temos que $v_{kb} - \tilde{p}_k \geq v_{ib} - \tilde{p}_i$, o que implica que $p_k - \tilde{p}_k \geq p_i - \tilde{p}_i$. Pela regra de quebra de empates usada na definição de x_p, a desigualdade é, de fato, estrita, já que $\tilde{p}_k < \tilde{p}_i$.

Agora, suponha que $\tilde{p}_k \leq \tilde{p}_i/4$. Como $\tilde{p}_i \leq 2p_i/3$, temos que

$$p_i - \tilde{p}_i \;\geq\; \frac{3\tilde{p}_i}{2} - \tilde{p}_i \;=\; \frac{\tilde{p}_i}{2} \;\geq\; 2\tilde{p}_k \;=\; 3\tilde{p}_k - \tilde{p}_k \;\geq\; p_k - \tilde{p}_k,$$

contradizendo o fato que $p_k - \tilde{p}_k > p_i - \tilde{p}_i$. Portanto $\tilde{p}_k > \tilde{p}_i/4$, mas como $\tilde{p}_k$ e $\tilde{p}_i$ são preços em $\mathcal{D}_2$, concluímos que $\tilde{p}_k \geq \tilde{p}_i/2$. Temos que,

$$\tilde{p}_k \;\geq\; \frac{\tilde{p}_i}{2} \;>\; \frac{p_i + \tilde{p}_k - p_k}{2} \;\geq\; \frac{p_i + \tilde{p}_k - 3\tilde{p}_k}{2},$$

isto é, $\tilde{p}_k > p_i/4$. Concluímos que $\mathrm{SOL}(\tilde{p}) \geq \mathrm{SOL}(p)/4$. □

É possível provar um resultado similar para precificações que são restritas a $\mathcal{D}_{1+\varepsilon}$, para um $\varepsilon > 0$ arbitrário. A razão obtida para tais precificações com respeito ao valor ótimo aproxima-se de 1 quando ε tende a 0. Esse fato é apresentado no próximo teorema.

Teorema 3.5.2. Seja p uma precificação. Para cada real ε tal que $0 < \varepsilon < 1$, existe uma precificação $\tilde{p}^\varepsilon$ consistindo apenas de preços em $\mathcal{D}_{1+\varepsilon}$ tal que $\mathrm{SOL}(\tilde{p}^\varepsilon)$ tende a $\mathrm{SOL}(p)$ conforme ε tende a 0.

Demonstração. Para cada item i, seja $\tilde{p}_i^\varepsilon = \lfloor p_i/r \rfloor_{1+\varepsilon}$, onde r é uma constante que escolheremos depois. Note que $\frac{p_i}{(1+\varepsilon)r} \leq \tilde{p}_i^\varepsilon \leq \frac{p_i}{r}$.

Como na prova do Teorema 3.5.1, cada comprador que recebe um item em x_p também recebe um item em $x_{\tilde{p}^\varepsilon}$, e se o comprador b troca o item i pelo item k quando a precificação muda para $\tilde{p}^\varepsilon$, então $p_k - \tilde{p}_k^\varepsilon > p_i - \tilde{p}_i^\varepsilon$ a não ser que $\tilde{p}_k^\varepsilon \geq \tilde{p}_i^\varepsilon$.

Se $\tilde{p}_k^\varepsilon \geq \tilde{p}_i^\varepsilon$, então $\tilde{p}_k^\varepsilon \geq \frac{p_i}{(1+\varepsilon)r}$. Caso contrário,

$$\tilde{p}_k^\varepsilon \geq \frac{p_k}{(1+\varepsilon)r} > \frac{\tilde{p}_k^\varepsilon + p_i - \tilde{p}_i^\varepsilon}{(1+\varepsilon)r} \geq \frac{\tilde{p}_k^\varepsilon + p_i - \frac{p_i}{r}}{(1+\varepsilon)r},$$

o que implica que

$$\tilde{p}_k^\varepsilon \geq \frac{r-1}{(1+\varepsilon)r^2 - r} p_i.$$

Pode ser mostrado que o máximo no lado direito é alcançado quando $r = 1 + \frac{\sqrt{\varepsilon}}{\sqrt{1+\varepsilon}}$. Usando esse valor de r, temos que

$$\tilde{p}_k^\varepsilon \geq \frac{1}{2\sqrt{\varepsilon(1+\varepsilon)} + 2\varepsilon + 1} p_i.$$

Conforme ε tende a 0, o valor escolhido para r tende a 1, e $\mathrm{SOL}(\tilde{p}^\varepsilon)$ tende a $\mathrm{SOL}(p)$. □

Uma consequência desse resultado é o seguinte. Se existir uma aproximação de razão constante para o Problema da Precificação Livre de Inveja com os preços restritos a $\mathcal{D}_{1+\varepsilon}$ para algum $\varepsilon > 0$, então existiria uma aproximação de razão constante para a versão irrestrita do problema. Isso é impossível a menos que P = NP, como mostrado por Chalermsook et al. [CCKK12]. Portanto, mesmo para essa variante, não é possível encontrar uma aproximação de razão constante a menos que P = NP. Por outro lado, um algoritmo empiricamente rápido capaz de resolver essa variante encontraria boas soluções para o problema original.

Capítulo 4

Leilões de Anúncios

Um relatório anual da Econsultancy e SEMPO [ES10] estimou que a industria norte-americana de propaganda de mecanismos de busca valia US$14,6 bilhões em 2009 com um crescimento de anual de 8%. De fato, as buscas patrocinadas são uma das formas mais rentáveis de propaganda e geraram aproximadamente US$7 bilhões em 2005 [NRTV07, Cap. 28]. Assim, o problema de alocar propagandas de anunciantes em *slots* de propaganda em buscas online de forma a maximizar o lucro da ferramenta de busca é de grande importância econômica para empresas como Google e Microsoft.

Motivados pela grande importância do mercado de anúncios online, muitos artigos recentes apresentaram modelos e resultados para leilões relacionados a anúncios online [AMT07, ABK+08, BJN07, CG08, DH09, GMNS08, GM08, MSVV05, Sri08]. Um dos modelos considerados na literatura é o **Leilão de Anúncios de Primeiro Preço** (*First-Price Ad Auctions*) onde cada busca por uma palavra-chave tem apenas um slot, cada anunciante tem um orçamento e faz um lance para ser anunciado junto com o resultado da busca dessa palavra-chave. A cada busca realizada é necessário escolher um dos anunciantes interessados nessa palavra-chave. Esse anunciante terá sua propaganda exibida e irá pagar o valor de seu lance. Existem vários resultados para esse modelo, tanto para a versão offline quanto para a versão online (quando não sabemos quais serão as próximas palavras-chave e quais serão os lances dos compradores) [LLN06, MSVV05, BJN07, DH09, CG08, Sri08].

Para a versão online do Leilão de Anúncios de Primeiro Preço, a melhor razão de competitividade conhecida é 2 [LLN06] quando não há restrições nos valores dos lances relativamente aos orçamentos. Quando os lances são muito menores que os orçamentos, Mehta et al. [MSVV05] e Buchbinder et al. [BJN07] apresentaram um algoritmo que alcança a razão de competitividade ótima de $e/(e-1)$. Devanur e Hayes [DH09] mostraram, para qualquer $\epsilon > 0$, sob certas hipóteses, um algoritmo $(1+\epsilon)$-competitivo para o caso em que as palavras-chaves chegam em uma ordem aleatória. Para a versão offline do Leilão de Anúncios de Primeiro Preço, o melhor resultado conhecido é de Chakrabarty e Goel [CG08] e Srinivasan [Sri08]. Eles apresentaram uma 4/3-aproximação e mostraram que é NP-difícil aproximar essa versão por um fator de 16/15.

Azar et al. [ABKN09] consideraram então o **Leilão de Anúncios de Segundo Preço** (*Second-Price Ad Auctions*). Nesse modelo, quando uma busca é realizada, é necessário escolher dois compradores e entre esses dois anunciar aquele com maior lance cobrando o lance do outro comprador. (Isto é equivalente a fazer um leilão de segundo preço pelo slot entre apenas esses dois compradores.) Tal modelo busca capturar o fato de que os mecanismos de buscas normalmente se utilizam do Leilão de Segundo Preço Generalizado [EOS07, Var07] para decidir os vencedores e o preço dos slots para uma determinada busca. De fato, Goel et al. [GMNS08] consideraram anteriormente uma variante online do Leilão de Anúncios de Segundo Preço. O modelo deles difere do modelo proposto por Azar et al. porque permite que os compradores façam lances acima do orçamento restante mesmo sendo cobrado no máximo o seu orçamento restante.

Azar et al. [ABKN09] apresentaram uma (n/c)-aproximação para a versão offline do problema do Leilão de Anúncios de Segundo Preço onde n é o número de palavras-chaves (itens) e c é a menor razão entre o orçamento e o maior lance de um comprador. Eles também provaram que é

NP-difícil aproximar a versão offline do problema por uma razão $o(n)$. Eles consideraram também um problema mais simples, chamado **Emparelhamento de Segundo Preço** (2PM, *Second-Price Matching*), onde todos os compradores são restritos a lances binários e têm orçamento unitário, isto é, um comprador pode ter o seu anúncio exibido no máximo uma vez. Eles provaram que a versão offline do 2PM é APX-difícil e apresentaram uma 2-aproximação para esse caso. Para a versão online, eles provaram um limitante inferior de n na razão de competitividade de qualquer algoritmo determinístico (juntamente com um algoritmo determinístico n-competitivo), um limitante inferior de 2 para a razão de competitividade de qualquer algoritmo aleatorizado e apresentaram um algoritmo aleatorizado 5,083-competitivo para o problema.

Nesse capítulo iremos considerar um caso particular do Leilão de Anúncios de Segundo Preço onde cada lance é 0 ou 1. Nós o denominamos **Leilão de Anúncios de Segundo Preço com Lances Binários** (B2PAA). Note que o B2PAA é uma generalização do 2PM onde cada comprador tem um orçamento arbitrário. Primeiramente nos concentramos na versão offline desses problemas. Mostraremos como usar um algoritmo para o 2PM que encontra uma solução a partir da remoção de arestas de um emparelhamento máximo no grafo bipartido de compradores e itens para obter uma aproximação para o B2PAA. Usando a 2-aproximação de Azar et al. [ABKN09] para o 2PM, obtemos uma 2-aproximação para o B2PAA.

A seguir focamos em instâncias de leilões de anúncios com boa competitividade. Chamamos de **grau** de i o número de compradores interessados em um item i . Consideramos um mercado δ-**competitivo** se o grau mínimo de um item é pelo menos um inteiro fixo δ. Isto é, existem pelo menos δ compradores interessados em cada item. De forma geral, dizemos que um mercado é competitivo se existe $\delta \geq 3$ tal que o mercado é δ-competitivo. Primeiramente parametrizamos a análise anterior e os resultados para o 2PM em termos de δ.

Adaptamos a prova de complexidade de Azar et al. e mostramos que 2PM continua APX-difícil para um $\delta > 2$ fixo. Porém, é interessante notar que o limitante inferior se enfraquece (isto é, aproxima-se de 1) quando δ cresce. Além disso, mostramos que a 2-aproximação de Azar et al. [ABKN09] para o 2PM não obtém razão melhor em mercados competitivos.

Apresentamos então uma aproximação para o 2PM cuja razão é melhor do que 2 para $\delta \geq 4$. A razão de nosso algoritmo se aproxima de 1.5 conforme δ cresce. Para $\delta \leq 3$, a razão do nosso algoritmo é 2, e portanto é, de forma geral, tão bom quanto o algoritmo de Azar et al. e melhor para mercados mais competitivos. Nosso algoritmo, assim como o de Azar et al., usa um emparelhamento máximo e pode ser usado para obter uma solução do B2PAA, sendo que a razão para mercados competitivos é mantida.

Na busca de algoritmos melhores para o 2PM, implementamos o nosso algoritmo juntamente com algumas variantes e realizamos alguns estudos empíricos. Pelo que observamos em nossos experimentos, sugerimos uma variante do nosso algoritmo que funciona empiricamente melhor. Essa variante em particular utiliza um emparelhamento máximo. Naturalmente, buscamos provar uma razão de aproximação melhor para essa variante mas fomos capazes de provar que, infelizmente, qualquer algoritmo que produza um emparelhamento de segundo preço contido em um emparelhamento máximo não é capaz de obter uma razão de aproximação melhor do que 2 para o caso geral do 2PM.

Finalmente consideramos a versão online do 2PM e mostramos que o algoritmo RankingSimulate [ABKN09] é $(2^{\delta-1}/(2^{\delta-1}-1))(\sqrt{e}/(\sqrt{e}-1))$-competitivo, o que é melhor do que 5,083 (o resultado previamente conhecido) para $\delta \geq 3$.

4.1 Modelo e Notação

Para manter uma notação uniforme com o resto do texto, chamamos os anunciantes de compradores e as palavras-chaves de itens. Como nos capítulos anteriores, consideramos um conjunto $I = [n]$ de itens e um conjunto $B = [m]$ de compradores, bem como uma valoração v onde v_{ib} representa o valor do item i para o comprador b. Para cada comprador b, temos um orçamento $O_b > 0$ que indica quanto pode ser gasto pelo comprador b. Denotamos por $\tilde{O}_b$ o orçamento restante do comprador b

em um dado momento, isto é, seu orçamento menos qualquer valor que tenha pago na compra de itens até esse dado momento.

> No **Leilão de Anúncios de Segundo Preço** são dados B e O *a priori* e I e v de forma online. Os itens chegam um a um (em ordem), juntamente com a valoração de tal item para cada comprador. Quando um item i chega, precisamos escolher dois compradores b e b' de forma que $\tilde{v}_{ib} \geq \tilde{v}_{ib'}$, onde $\tilde{v}_{ib} = \min\{v_{ib}, \tilde{O}_b\}$ (idem para $\tilde{v}_{ib'}$), isto é, o valor que b dá para o item i nesse momento é o mínimo entre o seu real valor para o item e o seu orçamento restante. Escolhidos b e b', o comprador b recebe o item i e paga $\tilde{v}_{ib'}$. O objetivo é maximizar o lucro do leiloeiro.

Note que quando um item chega, fazemos um leilão de segundo preço entre os dois compradores escolhidos para tal item i (e por isso o problema recebe esse nome). A Figura 4.1 exemplifica um tal leilão.

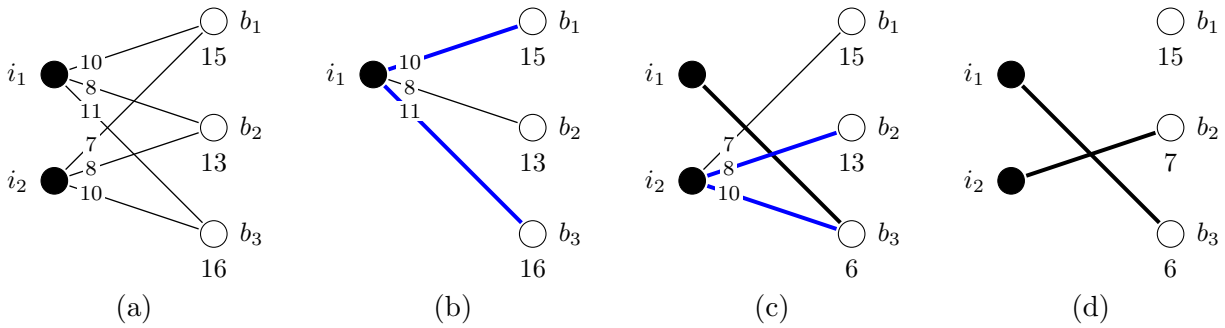


Figura 4.1: *Um exemplo do Leilão de Anúncios de Segundo Preço. Na figura* (a) *temos uma instância do problema. Na figura* (b) *escolhemos os compradores* b_1 *e* b_3 *para competir pelo item* i_1*. Como* b_3 *dá mais valor para esse item, ele receberá* i_1 *e o seu orçamento diminuirá para* 6*. Na figura* (c) *escolhemos os compradores* b_2 *e* b_3 *para competir pelo item* i_2*. Apesar de* b_3 *originalmente dar valor* 10 *para esse item, o seu orçamento restante é de apenas* 6 *e, portanto,* b_2 *recebe o item* i_2*. A figura* (d) *representa a situação final do mercado com as alocações realizadas e os orçamentos finais dos compradores.*

Apresentamos a seguir o caso particular do Leilão de Anúncios de Segundo Preço no qual estamos interessados nesse capítulo.

> **Leilão de Anúncios de Segundo Preço com Lances Binários** (também denotado por B2PAA): caso particular do Leilão de Anúncios de Segundo Preço onde para todo comprador b e item i temos que $v_{ib} \in \{0, 1\}$.

Dada uma valoração v onde para todo comprador b e item i temos que $v_{ib} \in \{0, 1\}$, podemos considerar o grafo bipartido $G = (I, B, E)$ onde $(i, b) \in E$ se e somente se $v_{ib} = 1$. Se para um comprador b e um item i temos que $v_{ib} = 1$, dizemos que b está **interessado** em i.

A seguir introduzimos um conceito útil para caracterizar as soluções viáveis do Leilão de Anúncios de Segundo Preço com Lances Binários.

Definição. Dizemos que um O-emparelhamento[1] M de G é um **O-emparelhamento de segundo preço** se, para toda aresta (i, b) em M, existe uma aresta (i, b') em E tal que menos do que $O_{b'}$ arestas (i', b') com $i' < i$ pertencem a M.

Isso significa que, ao emparelhar i com b existia um outro comprador b' interessado em i que ainda não havia esgotado o seu orçamento naquele momento. Chamamos esse comprador b' de **comprador de segundo preço** para i (note que b' não é necessariamente único). A Figura 4.2 apresenta uma instância do B2PAA juntamente com um O-emparelhamento de segundo preço.

[1] Abusamos um pouco da notação chamando de O-emparelhamento o ρ-emparelhamento onde, para todo item i, $\rho_i = 1$ e, para todo comprador b, $\rho_b = O_b$.

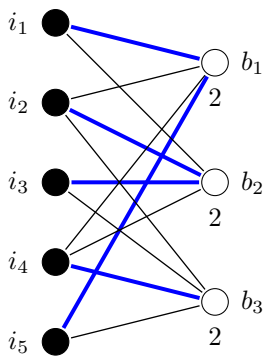


Figura 4.2: *Um exemplo de um O-emparelhamento de segundo preço, denotado pelas arestas azuis (grossas). Note como b_2 não é um comprador de segundo preço para o item i_4 (vendido para o comprador b_3) pois seu orçamento está esgotado no momento da venda de i_4. Note também que b_1 é um comprador de segundo preço para i_4 porque, apesar de receber dois itens, no momento da venda de i_4, b_1 havia comprado apenas o item i_1.*

Dada a correspondência entre uma atribuição de itens a compradores onde obtemos lucro estritamente positivo de todos os itens e um O-emparelhamento de segundo preço de G, consideramos que a entrada do B2PAA consiste de um grafo bipartido $G = (I, B, E)$ e um vetor de orçamentos O indexado por B. O objetivo é encontrar um O-emparelhamento de segundo preço em G de tamanho máximo. Na versão offline do problema G e O são dados na íntegra no começo do algoritmo. Na versão online, B e O são conhecidos inicialmente mas os elementos de I são revelados em ordem, um por vez, de forma online, juntamente com as arestas incidentes ao novo item revelado e devemos então decidir quem irá ser emparelhado com esse item.

Quando $O_b = 1$ para todo comprador b, dizemos que um O-emparelhamento de segundo preço é um **emparelhamento de segundo preço**.

> **Problema do Emparelhamento de Segundo Preço** (também denotado por 2PM): dado um grafo $G = (I, B, E)$, desejamos encontrar um emparelhamento de segundo preço de cardinalidade máxima em G.

Apesar de serem grandes simplificações do problema original, o Leilão de Anúncios de Segundo Preço com Lances Binários e o Problema do Emparelhamento de Segundo Preço ainda mantém uma característica interessante do problema. Para vender um item i para um comprador b ainda é necessário que exista um comprador b' interessado em i e que ainda não tenha exaurido o seu orçamento para servir como um comprador de segundo preço. Além disso, a escolha do comprador que receberá um determinado item é bastante importante, já que após exaurido o seu orçamento, um comprador não pode mais servir como comprador de segundo preço. De fato, é interessante notar que o caso particular do Leilão de Anúncios de Primeiro Preço onde as valorações são binárias e os orçamentos são todos iguais a 1 (como no Problema do Emparelhamento de Segundo Preço) é equivalente ao Problema do Emparelhamento Máximo que pode ser resolvido em tempo polinomial [Sch03] enquanto que o Problema do Emparelhamento de Segundo Preço é APX-difícil [ABK+08].

O **grau** de um item i é o número $|\{b \in B : (i, b) \in E\}|$, isto é, o número de compradores interessados em i. Denotamos por B2PAA(δ) a restrição do B2PAA a grafos onde o grau de todo item é pelo menos δ. Podemos assumir que $\delta \geq 2$, já que um item com grau menor que 2 não pode ser emparelhado em um O-emparelhamento de segundo preço. De forma análoga, definimos 2PM(δ).

Para um emparelhamento M e um item i tal que $(i, b) \in M$ para algum b em B, escrevemos $M(i) = b$ e $M^{-1}(b) = i$. Dizemos que b é o comprador de i em M e i é o item de b em M. Se M é um emparelhamento de segundo preço em G, dizemos também que, se $(i, b) \in M$, o item i é vendido para o comprador b.

Eventualmente abusamos da notação e utilizamos outro conjunto além de $[n]$ como o conjunto I de itens. Quando isso acontecer, definiremos explicitamente uma ordem sobre esse conjunto. Além

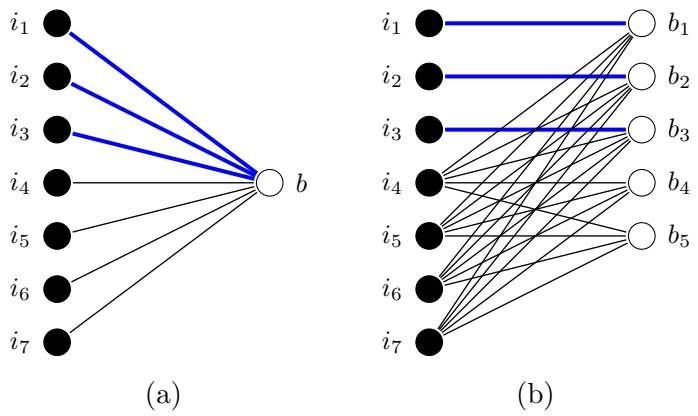


Figura 4.3: *A construção de G_O^M para um comprador b com $O_b = 5$. Na figura (a), as arestas azuis (grossas) denotam M e na figura (b) denotam M'.*

disso, no que segue, consideramos que $B = [m]$ para algum inteiro positivo m. Mas note que a ordem dos compradores não importa para esse problema. Além disso, de agora em diante, nos referiremos à versão offline do problema simplesmente como B2PAA ou 2PM.

4.2 Leilões de Anúncios de Segundo Preço com Lances Binários

Dado um grafo bipartido G e um vetor de orçamentos O, seja G_O o grafo bipartido obtido a partir de G adicionando $O_b - 1$ novas cópias de cada vértice b em B. Isto é, para cada vértice b em B, G_O tem O_b vértices com a mesma vizinhança, composta pelos itens que são adjacentes a b em G. Um O-emparelhamento M em G corresponde a um emparelhamento M' em G_O e vice-versa. De fato, para cada (i, b) em M, seja $(i, b') \in M'$ onde b' é uma das cópias de b em G_O. Escolha b' tal que duas arestas (i, b) e (i', b) em M correspondam a arestas (i, b') e (i', b'') com $b' \neq b''$. Seja G_O^M o subgrafo de G_O obtido pela remoção, para cada aresta (i, b) em M, das arestas $(i, b') \notin M'$. Note que M' é um emparelhamento de G_O^M. A Figura 4.3 mostra um exemplo de tal construção. Observe que, se o grau mínimo em G de um item é pelo menos δ então também o é em G_O^M.

Seja Separar um algoritmo polinomial tal que, dado um grafo bipartido G, um vetor de orçamentos O e um O-emparelhamento M de G, devolve um grafo bipartido G_O^M e um emparelhamento M' de G_O^M como descrito acima. Seja Juntar um algoritmo polinomial que, dado G_O^M, um vetor de orçamentos O e um emparelhamento de segundo preço S em G_O^M, devolve um O-emparelhamento de G, onde o item i está casado com o comprador b se i está casado em S com uma cópia de b. Finalmente, seja O-EmparelhamentoMáximo um algoritmo polinomial que recebe um grafo bipartido G e um vetor de orçamentos O e devolve um O-emparelhamento de cardinalidade máxima em G [Mun57, Sch03].

Definição. Chamamos um algoritmo polinomial de **especial** se, dado um grafo G' e um emparelhamento M' em G', esse algoritmo devolve um emparelhamento de segundo preço contido em M'. Dizemos que ele é α-**especial** para o 2PM(δ) se a saída tem tamanho pelo menos $\alpha|M'|$.

Note que qualquer algoritmo α-especial para o 2PM(δ) pode ser adaptado para se tornar uma α-aproximação para o 2PM(δ). De fato, basta aplicar tal algoritmo em um emparelhamento máximo do grafo de entrada que, por sua vez, pode ser encontrado em tempo polinomial. A seguir apresentamos um algoritmo O-Alg que utiliza um algoritmo especial Alg.

O-ALG(n, m, E, O)
1 M = O-EMPARELHAMENTOMÁXIMO(n, m, E, O)
2 G_O^M, M' = SEPARAR(n, m, E, O, M)
3 $m' = \sum_{b=1}^{m} O_b$
4 S = ALG$(n, m', E(G_O^M), M')$
5 **devolva** JUNTAR(G_O^M, O, S)

Teorema 4.2.1. Se ALG é α-especial para o 2PM(δ), então o algoritmo O-ALG é uma α-aproximação para o B2PAA(δ).

Demonstração. Note que M e M' têm o mesmo número de arestas. Além disso, o tamanho de um O-emparelhamento de cardinalidade máxima é um limitante superior no tamanho de uma solução ótima do B2PAA e o O-emparelhamento encontrado por JUNTAR(G_O^M, O, S) tem o mesmo tamanho que S. Basta, então, mostrar que M'' = JUNTAR(G_O^M, O, S) é, de fato, um O-emparelhamento de segundo preço.

Seja (i, b) uma aresta em M''. Então, por construção, $(i, b') \in S$ para alguma cópia b' de b em G_O^M. Como S é um emparelhamento de segundo preço, existe um comprador c' em G_O^M (e, por construção, c' não é uma cópia de b) tal que $(i, b') \in E(G_O^M)$ e c' não está emparelhado com um item $i' < i$. Seja c o comprador do qual c' foi copiado. É fácil ver que o número de itens $i' < i$ emparelhados com c em M'' é menor do que O_c porque c' não está emparelhado com nenhum desses itens em S. Concluímos que M'' é um O-emparelhamento de segundo preço para $G = ([n], [m], E)$. □

4.3 Resultados de Complexidade para Mercados Competitivos

Nessa seção mostramos que 2PM(δ) é APX-difícil para $\delta \geq 3$. Esse resultado é um aprimoramento do resultado de Azar et al. [ABKN09] que 2PM(2) é APX-difícil.

Começamos apresentando uma redução a partir da Cobertura por Vértices para o 2PM criada por Azar et al. [ABKN09].

Dado um grafo H, denotamos por $f(H)$ a seguinte instância do 2PM. Para cada aresta e de H, temos um item e, para cada vértice v de H, temos um comprador. Temos uma aresta (e, v) se v é uma das duas pontas de e. Para cada e em H, temos um comprador x_e e uma aresta (e, x_e). Finalmente, para cada vértice v em H, temos um *gadget* contendo dois itens h_v e ℓ_v e dois compradores y_v e z_v e temos as arestas (h_v, v), (h_v, y_v), (ℓ_v, y_v), e (ℓ_v, z_v). A ordem em I é tal que, para cada vértice v de H, h_v aparece antes do que ℓ_v e os itens que representam arestas aparecem depois de todos os h_v's e ℓ_v's. A Figura 4.4 representa tal construção para um grafo composto por uma única aresta.

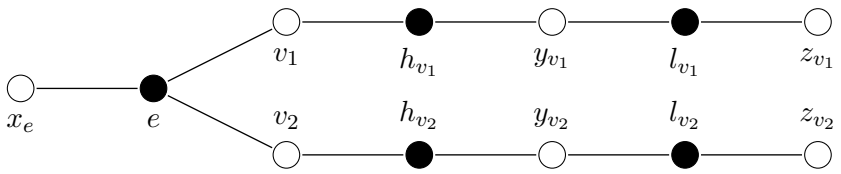


Figura 4.4: *Uma instância de $f(H)$ onde H é o grafo com vértices v_1 e v_2 e uma aresta $e = \{v_1, v_2\}$.*

Mostraremos agora como estender $f(H)$ para uma instância do 2PM(δ). Dado um grafo H e um $\delta \geq 3$, denotamos por $f_\delta(H)$ a seguinte instância do 2PM(δ). A instância estende $f(H)$ de forma que cada item tenha grau δ.

Para cada item que representa uma aresta e, primeiramente temos $\delta - 3$ novos compradores, conectados a e, denotados por $d_e^1, \ldots, d_e^{\delta-3}$. Temos então $\delta - 3$ cópias de e (itens com a mesma vizinhança de e). Dessa forma, e e todas as suas cópias têm grau δ.

Para cada vértice v, temos $\delta - 2$ novos compradores, denotados por $d^1_{h_v}, \ldots, d^{\delta-2}_{h_v}$, conectados com h_v e $\delta - 2$ novos compradores, denotados por $d^1_{\ell_v}, \ldots, d^{\delta-2}_{\ell_v}$, conectados com ℓ_v. Finalmente, temos $\delta - 2$ cópias de h_v e $\delta - 2$ cópias de ℓ_v. Dessa forma, h_v, suas cópias, ℓ_v e suas cópias têm todos grau δ.

Chamamos os itens e compradores de $f(H)$ de **itens originais** e **compradores originais**. Chamamos os outros itens e compradores de **cópias de itens** e **novos compradores**.

Ordenamos I de forma que todas as cópias aparecem antes de todos os itens originais e, para os itens originais, seguimos a ordem de $f(H)$. A Figura 4.5 mostra um exemplo para $f_4(H)$.

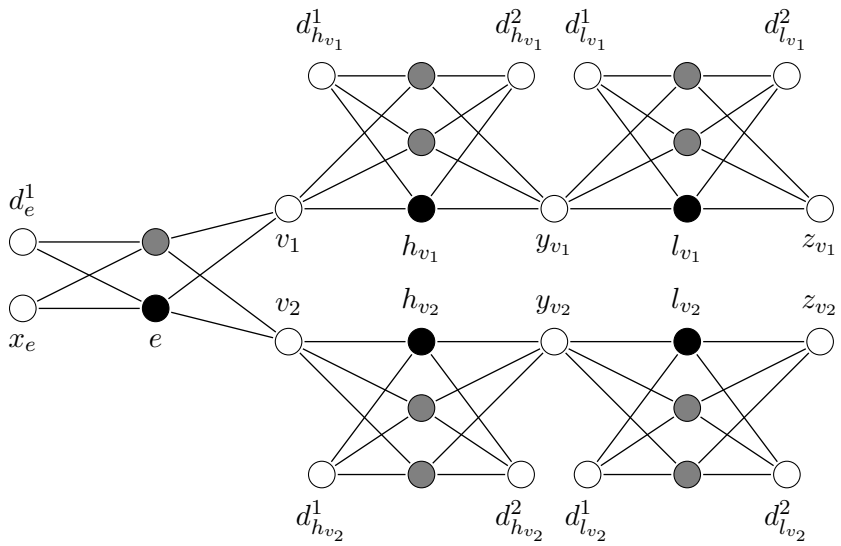


Figura 4.5: *Uma instância de $f_4(H)$ onde H é o grafo com vértices v_1 e v_2 e uma aresta $e = \{v_1, v_2\}$. Os vértices de cor preta representam os itens em $f(H)$ e os vértices de cor cinza representam as cópias desses itens.*

Definição. Denotamos por $\mathrm{OPT}_{2P}(G)$ o valor de uma solução ótima da instância G do 2PM e denotamos por $\mathrm{OPT}_{VC}(H)$ o valor de uma solução ótima da instância H da Cobertura por Vértices.

Podemos agora relacionar o valor de uma solução ótima de $f(H)$ com o valor de uma solução ótima de $f_\delta(H)$.

Lema 4.3.1. Seja M uma solução para $f_\delta(H)$. Então existe uma solução M' onde todas as cópias de itens estão emparelhadas com novos compradores e $|M'| \geq |M|$.

Demonstração. Suponha que, para uma aresta e em H, existe uma cópia i de e que não está emparelhada com um novo comprador em M. Se existe um novo comprador não emparelhado vizinho de i, basta emparelhar tal comprador com i (desemparelhando i e x_e, eventualmente). Se todo novo comprador vizinho de i está emparelhado então, pelo Princípio da Casa dos Pombos, e está emparelhado com um novo comprador d^j_e (para algum $1 \leq j \leq \delta - 3$). Basta então trocar os papéis de i e e, isto é, se i não está emparelhado, podemos emparelhar o comprador de e com i e deixar e desemparelhado e se i está emparelhado, podemos trocar os compradores de e e i. O mesmo é válido para h_v e l_v para todo vértice v em H. Note que, por esse processo, o tamanho do emparelhamento não diminui. Portanto, podemos repetir esse processo e obter M' com a propriedade desejada e tal que $|M'| \geq |M|$. □

Corolário 4.3.2. $\mathrm{OPT}_{2P}(f_\delta(H)) = (\delta - 3)|E| + 2(\delta - 2)|V| + \mathrm{OPT}_{2P}(f(H))$.

Demonstração. Primeiramente, note que $\mathrm{OPT}_{2P}(f_\delta(H)) \geq (\delta - 3)|E| + 2(\delta - 2)|V|$, já que podemos emparelhar todas as cópias de itens com novos compradores e obter uma solução de tamanho $(\delta - 3)|E| + 2(\delta - 2)|V|$. Seja M uma solução ótima para $f_\delta(H)$ e assuma (pelo Lema 4.3.1)

que em M todas as cópias dos itens estão emparelhadas com novos compradores. Então M consiste de um emparelhamento perfeito entre novos compradores e cópias de itens, e uma solução para $f(H)$, que tem que ser ótima pois M é solução ótima para $f_\delta(H)$. □

Iremos agora começar a relacionar o 2PM(δ) com a Cobertura por Vértices.

Lema 4.3.3 (Azar et al. [ABKN09])**.** Seja H um grafo e $k \geq |V(H)| + |E(H)|$ o tamanho de uma solução viável para $f(H)$. Então existe uma cobertura por vértices T de H tal que $|T| = 2|V(H)| + |E(H)| - k$. Em particular, $\mathrm{OPT}_{VC}(H) = 2|V(H)|+|E(H)|-\mathrm{OPT}_{2P}(f(H))$.

É claro que $f(H)$ tem uma solução de tamanho $|V(H)| + |E(H)|$: basta vender cada ℓ_v a z_v e cada e a x_e. Um dos extremos de e pode servir de comprador de segundo preço para e enquanto y_v pode servir de comprador de segundo preço para ℓ_v. A ideia do lema acima é vender h_v para v exceto para os vértices de uma cobertura por vértices. Toda aresta então terá um extremo na cobertura que poderá servir como seu comprador de segundo preço e y_v pode continuar servido de comprador de segundo preço para ℓ_v.

Corolário 4.3.4. Para $\delta \geq 3$, sejam H um grafo e r o tamanho de uma solução viável para $f_\delta(H)$ de forma que $r \geq (\delta-3)|E(H)| + 2(\delta-2)|V(H)| + |V(H)| + |E(H)|$. Então existe uma cobertura por vértices T de H tal que $|T| = 2(\delta-1)|V(H)| + (\delta-2)|E(H)| - r$. Em particular, $\mathrm{OPT}_{VC}(H) = 2(\delta-1)|V(H)| + (\delta-2)|E(H)| - \mathrm{OPT}_{2P}(f_\delta(H))$.

Demonstração. Pelo Lema 4.3.1, existe uma solução de tamanho r para $f_\delta(H)$ onde todas as cópias de itens são emparelhadas com novos compradores. Portanto, existe uma solução para $f(H)$ de tamanho $k = r - (\delta-3)|E(H)| - 2(\delta-2)|V(H)|$. Pelo Lema 4.3.3, existe uma cobertura por vértices T de H tal que $|T| = 2|V(H)| + |E(H)| - k$, de onde o resultado segue. □

Também é fácil ver que $f_\delta(H)$ tem uma solução de tamanho $(\delta-3)|E(H)| + 2(\delta-2)|V(H)| + |V(H)| + |E(H)|$. É necessário apenas vender todas as cópias de itens para novos compradores e então vender cada l_v para z_v e cada e para x_e.

Usamos o seguinte teorema para obter o resultado de inaproximabilidade.

Teorema 4.3.5 (Chlebík e Chlebíková [CC06])**.** É NP-difícil aproximar a Cobertura por Vértices em grafos 4-regulares por um fator de 53/52.

Teorema 4.3.6. É NP-difícil aproximar 2PM(δ) para $\delta \geq 3$ por um fator de $\frac{8\delta-13}{8\delta-13-1/52}$.

Demonstração. Suponha que temos uma α-aproximação para o 2PM(δ) com $\alpha = \frac{8\delta-13}{8\delta-13-1/52}$ e seja $H = (V, E)$ um grafo 4-regular. Observe que $|E| = 2|V|$, e portanto $2(\delta-1)|V| + (\delta-2)|E| = (2\delta-3)|E|$. Logo, pelo Corolário 4.3.4, existe uma cobertura por vértices T tal que

$$\begin{aligned} |T| &\leq (2\delta-3)|E| - \frac{\mathrm{OPT}_{2P}(f_\delta(H))}{\alpha} \\ &= (2\delta-3)|E| - \frac{(2\delta-3)|E| - \mathrm{OPT}_{VC}(H)}{\alpha} \\ &= \Big(\frac{\alpha-1}{\alpha}\Big)(2\delta-3)|E| + \frac{\mathrm{OPT}_{VC}(H)}{\alpha}. \end{aligned}$$

Além disso, como H é 4-regular, $\mathrm{OPT}_{VC}(H) \geq |E|/4$. Temos que

$$\begin{aligned} |T| &\leq 4\Big(\frac{\alpha-1}{\alpha}\Big)(2\delta-3)\mathrm{OPT}_{VC}(H) + \frac{\mathrm{OPT}_{VC}(H)}{\alpha} \\ &= \frac{4(\alpha-1)(2\delta-3)+1}{\alpha}\,\mathrm{OPT}_{VC}(H). \end{aligned}$$

Substituindo α, concluímos que, se tal α-aproximação existe, então podemos aproximar a Cobertura por Vértices em grafos 4-regulares por um fator de 53/52, o que implica que P = NP. □

4.4 O Algoritmo ReverseMatching em Mercados Competitivos

Nessa seção provaremos que o algoritmo proposto por Azar et al. [ABKN09] chamado ReverseMatching tem uma razão de aproximação de pelo menos 2 em mercados competitivos. Isto é, a razão de aproximação não melhora em mercados competitivos.

Começamos com algumas definições. Seja $G = (I, B, E)$ um grafo bipartido com $I = [n]$ e $B = [m]$.

Definição ([ABKN09])**.** Seja M um emparelhamento em G. Para um item i, dizemos que uma aresta $e = (i, b')$ é uma **aresta para cima** se existe uma aresta $f = (i', b')$ em M com $i' < i$. Se existe uma aresta $f = (i', b')$ em M com $i' > i$ ou se não existe aresta em M adjacente a b' então e é uma **aresta para baixo**.

Figura 4.6: *Na figura* (a)*, a aresta* (i', b) *é uma aresta para cima e na figura* (b) *a aresta* (i', b) *é uma aresta para baixo.*

O algoritmo encontra um emparelhamento máximo M em G e elimina algumas arestas de M para obter um emparelhamento de segundo-preço. Se o algoritmo decide manter uma aresta (i, b) de M, duas coisas podem acontecer. Se existe uma aresta para baixo (i, b'), então ele pode manter (i, b) em M já que b' pode servir como um comprador de segundo preço para i. Caso contrário, toda aresta (i, b') com $b' \neq b$ é uma aresta para cima e o algoritmo escolhe uma aresta para cima (i, b') e remove $(M^{-1}(b'), b')$ de M. Dessa forma, b' pode servir como um comprador de segundo preço para i. A próxima definição captura a relação entre a aresta $(M^{-1}(b'), b')$ e a aresta (i, b) nesse caso.

Definição. Seja M um emparelhamento em G e $e = (i, b) \in M$. Uma aresta $f = (i', b')$ em M é um **competidor superior** de e se $i' < i$ e $(i, b') \in E$, isto é, se (i, b') é uma aresta para cima. Se f é um **competidor superior** de e, então e é um **competidor inferior** de f.

O algoritmo proposto por Azar et al. [ABKN09] para o 2PM offline é muito simples: encontre um emparelhamento máximo e itere em ordem inversa nos itens emparelhados (isto é, dos itens de maior índice para os de menor índice). Se um item tem uma aresta para baixo então não temos nada a fazer. Mas se um item tem apenas arestas para cima, então selecione um de seus competidores superiores e remova-o do emparelhamento. A seguir apresentamos o algoritmo ReverseMatching.

```
ReverseMatching(G)
M = Emparelhamento-Máximo(G)
para i = n decrescendo até 1
    se i está emparelhado em M
        e = (i, M(i))
        se e não tem uma aresta para baixo
            Remova de M um competidor superior de e
devolva M
```

Azar et al. mostraram que o ReverseMatching é uma 2-aproximação para o 2PM(2). A seguir mostramos que essa razão é justa para o 2PM(δ) para todo $\delta \geq 2$.

Lema 4.4.1. Para cada $\delta \geq 2$, ReverseMatching não é uma $(2-\epsilon)$-aproximação para o 2PM(δ) para qualquer $\epsilon > 0$.

Demonstração. Descreveremos uma instância $G = (I, B, E)$ para o 2PM(δ). Fixe um inteiro $s > 0$ e seja $I = [\delta + s]$, $B = [\delta + s + 1]$ e E definido a seguir. Os itens $1, \ldots, \delta$ são todos adjacentes aos compradores $1, \ldots, \delta$ e todo item i para $\delta + 1 \leq i \leq \delta + s$ é adjacente aos compradores $1, \ldots, \delta$ e também aos compradores i e $i + 1$. Note que os itens $1, \ldots, \delta$ têm grau δ e os itens $\delta + 1, \ldots, \delta + s$ têm grau $\delta + 1$, portanto G é, de fato, uma instância do 2PM(δ). A figura 4.7 mostra um exemplo de tal G para $\delta = 3$ e $s = 3$.

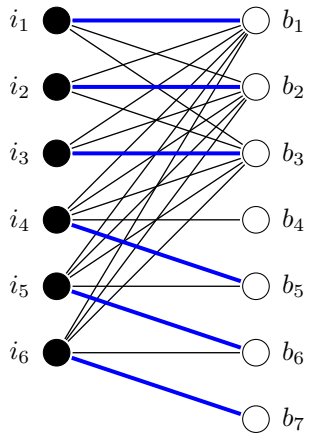


Figura 4.7: *O grafo G da prova do Lema 4.4.1 para $\delta = s = 3$. O k-ésimo item é indicado por i_k e o k-ésimo comprador por b_k. As arestas azuis (grossas) indicam um emparelhamento máximo em G.*

Primeiramente, vamos mostrar que existe um emparelhamento de segundo preço em G de tamanho $s + \delta - 1$. Considere o emparelhamento onde o item 1 não está emparelhado e o item i, para $2 \leq i \leq s + \delta$, está emparelhado com o comprador i. É fácil ver que, como todo item é adjacente ao comprador 1, este é um emparelhamento de segundo preço com tamanho $s + \delta - 1$.

Vamos mostrar que o algoritmo pode produzir um emparelhamento de segundo preço de tamanho $\delta - 1 + \lceil \frac{s}{2} \rceil$. Suponha que o emparelhamento máximo usado pelo algoritmo é o seguinte: itens $1, \ldots, \delta$ são emparelhados com os compradores $1, \ldots, \delta$ respectivamente e cada item i para $\delta + 1 \leq i \leq \delta + s$ é emparelhado com o comprador $i + 1$. O algoritmo irá iniciar no item $\delta + s$ e irá remove uma aresta da solução, já que o item $\delta + s$ tem apenas arestas para cima. Suponha que o algoritmo remova a aresta $(\delta + s - 1, \delta + s)$ do emparelhamento. Então o algoritmo prosegue para o item $\delta + s - 2$ e novamente precisa remover uma aresta do emparelhamento. De fato, para todo item i tal que $\delta + 1 \leq i \leq \delta + s$ e i tem a mesma paridade que $\delta + s$, considere que para manter a aresta $(i, i + 1)$, o algoritmo remova a aresta $(i - 1, i)$ da solução. Quando o algoritmo atinge o item δ, ele remove a aresta $(1, 1)$ e após isso não é necessário remover nenhuma outra aresta, já que todo item i para $2 \leq i \leq \delta - 1$ é adjacente ao comprador 1. Concluímos que a solução produzida para G pode ter tamanho $\delta - 1 + \lceil \frac{s}{2} \rceil$.

Como a razão

$$\frac{\delta - 1 + \lceil \frac{s}{2} \rceil}{s + \delta - 1}$$

tende a $\frac{1}{2}$ para δ fixo quando s cresce, o resultado segue. □

4.5 Novos Resultados para Mercados Competitivos

Na Seção 4.4 provamos que a razão do algoritmo ReverseMatching projetado por Azar et al. [ABKN09] é pelo menos 2 mesmo para o 2PM(δ) onde δ é grande (porém, fixo). A seguir apresentamos uma aproximação que tem sua razão aprimorada conforme δ cresce.

O algoritmo recebe um emparelhamento máximo M em G e elimina algumas das arestas de M para obter um emparelhamento de segundo preço, isto é, ele é especial.

Note que é possível que uma aresta e de M tenha apenas arestas adjacentes para cima e, para manter e em M, é necessário remover de M pelo menos um dos competidores superiores de M.

Da mesma forma, se e é removido de M, então podemos manter em M todos os seus competidores inferiores.

Definição. Seja M um emparelhamento em G e $e \in M$. Uma aresta f de M é o **competidor do topo** de e se f é o competidor superior de e com o menor item. Isto é, se $f = (i, b)$ e (i', b') é outro competidor superior de e, então $i < i'$.

Nosso algoritmo, chamado Marcar-e-Matar, utiliza a ideia de cancelar vendas que bloqueiem pelo menos outras duas vendas. Ele recebe o número n de itens, o número m de compradores, e o conjunto E de arestas e um emparelhamento máximo M em $G = ([n], [m], E)$ e devolve um emparelhamento de segundo preço no grafo G contido em M.

O algoritmo Marcar-e-Matar, apresentado a seguir, itera sobre as arestas de M em ordem reversa de I, dividindo M em três: as arestas pretas, as arestas cinzas e as arestas mortas. As aresta pretas são arestas de M que têm uma aresta para baixo. Elas obviamente podem ser mantidas na solução. Cada aresta cinza tem apenas arestas para cima e requer que um desses competidores superiores seja descartado (morto) para que ela possa ser mantida na solução. Toda aresta cinza é primeiramente marcada como vermelha (de forma que toda aresta vermelha também tem apenas arestas para cima). Para uma aresta vermelha se tornar cinza, um de seus competidores superiores precisa ser descartado. Cada aresta vermelha marca todos os seus competidores superiores. Quando o algoritmo processa uma aresta de M com pelo menos duas marcas, essa aresta é descartada e todos os seus competidores inferiores vermelhos se tornam cinza e as marcas que eles criaram são apagadas. Algumas vezes, o algoritmo processa o competidor do topo de uma aresta vermelha f. Nesse caso, se esse competidor do topo tiver uma única marca ele será descartado para que f se torne cinza.

Os conjuntos de arestas pretas e vermelhas são denotados por P e V, respectivamente. Para o propósito da análise do algoritmo, as arestas cinza e as arestas mortas são divididas em dois conjuntos: C_1 e C_2, e R_1 e R_2, respectivamente. Esses conjuntos são usados para distinguir entre os dois casos descritos acima (uma aresta descartada com pelo menos duas marcas ou com apenas uma marca). Após o pseudocódigo, apresentamos alguns invariantes relacionados aos conjuntos P, C_1, C_2, V, R_1 e R_2 que ajudam a entender melhor o algoritmo.

```
Marcar-e-Matar (n, m, E, M)
V = ∅,   P = ∅,   C1 = ∅,   C2 = ∅,   R1 = ∅,   R2 = ∅
para cada e em M
    marcas(e) = 0
para i = n decrescendo até 1
    se i está emparelhado em M
        e = (i, M(i))
        se marcas(e) ≥ 2
            R2 = R2 ∪ {e}
            para cada competidor inferior f de e
                se f ∈ V
                    V = V \ {f},   C2 = C2 ∪ {f}
                    para cada competidor superior h de f
                        marcas(h) = marcas(h) − 1
        senão se e é o competidor do topo de uma aresta f em V
            R1 = R1 ∪ {e},   V = V \ {f},   C1 = C1 ∪ {f}
        senão se e tem uma aresta para baixo
            P = P ∪ {e}
        senão
            V = V ∪ {e}
            para cada competidor superior h de e
                marcas(h) = marcas(h) + 1
devolva P ∪ C1 ∪ C2
```

Na linha 4 do algoritmo, valem os seguintes invariantes:

(i) $\{P, V, C_1, C_2, R_1, R_2\}$ é uma partição das arestas de M com item em $\{i+1, \ldots, n\}$.

(ii) As arestas de P têm uma aresta para baixo.

(iii) As arestas de C_i têm um competidor superior em R_i, para $i \in \{1, 2\}$.

(iv) $P \cup C_1 \cup C_2$ é um emparelhamento de segundo preço para $([n], [m], E)$.

(v) As arestas em V têm seus competidores do topo em $M \setminus (P \cup C_1 \cup C_2 \cup V \cup R_1 \cup R_2)$.

Além disso, ao final do algoritmo, V está vazio, já que cada aresta que estava em V em alguma iteração será salva (movida para C_1 ou C_2) já que um de seus competidores superiores será morto (seja por ter pelo menos duas marcas ou por ser o competidor do topo com apenas uma marca). A Figura 4.8 mostra uma simulação da execução do algoritmo Marcar-e-Matar.

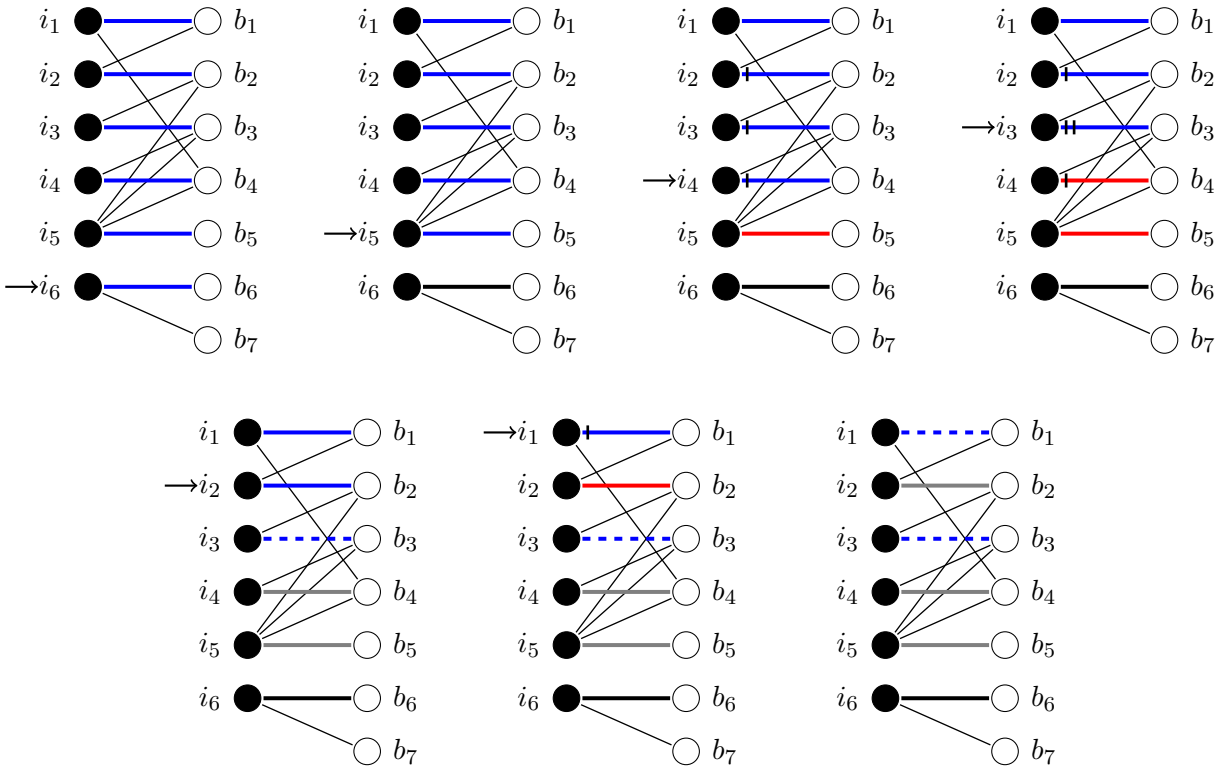


Figura 4.8: *Simulação do Algoritmo* Marcar-e-Matar. *As arestas azuis indicam arestas do emparelhamento M passado para o algoritmo que ainda não foram processadas pelo algoritmo. As arestas pretas finas indicam as outras arestas da instância. As arestas marcadas como pretas, vermelhas e cinzas são representadas na figura por arestas grossas de mesma cor. As arestas mortas são indicadas por arestas azuis grossas tracejadas. A seta representa o item que está sendo processado pelo algoritmo e, finalmente, as marcas são representadas pela quantidade de traços verticais em cada aresta.*

Os dois próximos lemas relacionam o tamanho do emparelhamento M com o tamanho dos conjuntos R_1 e R_2, apresentando um limitante inferior para o tamanho de M. Tais resultados serão úteis para limitar o número de arestas removidas de M pelo Marcar-e-Matar e permitirão obter a razão de aproximação do algoritmo. Começamos com um resultado que não depende da competitividade do mercado.

Lema 4.5.1. $|M| \geq 2|R_1| + 3|R_2|$.

Demonstração. A linha 15 é o único ponto do algoritmo que modifica R_1 e C_1. Portanto cada aresta e de R_1 está associada a uma aresta f de C_1 e vice-versa, implicando que $|R_1| = |C_1|$. O

conjunto R_2 é modificado apenas na linha 8, onde uma aresta e é incluída em R_2. Nesse ponto, existem pelo menos duas marcas em e, vindas de arestas de V. Nessa mesma iteração, na linha 11, essas arestas de V que marcam e são movidas para C_2 e, portanto, $|C_2| \geq 2|R_2|$. Concluímos que $|M| \geq |R_1| + |C_1| + |R_2| + |C_2| \geq 2|R_1| + 3|R_2|$. A primeira desigualdade vale porque R_1, C_1, R_2 e C_2 são subconjuntos de M dois a dois disjuntos (veja o invariante (i)). □

Lema 4.5.2. $|M| \geq (\delta - 1)|R_1| + |R_2|$, onde δ é o menor grau de um item.

Demonstração. Para cada aresta e em R_1, considere a iteração de Marcar-e-Matar na qual e entrou em R_1 e a aresta f que foi movida de V para C_1 nessa iteração (linha 15). Considere o conjunto $K_e = \{e' : e' \text{ é um competidor superior de } f\}$ e note que $|K_e| \geq \delta - 1$. (Isso é válido porque as arestas em V não têm arestas para baixo.) Note que K_e é disjunto de R_2. De fato, toda vez que uma aresta e' entra em R_2 (linha 8), todos os seus competidores inferiores em V são movidos para C_2 (linhas 9-13). Portanto, se $e' \in K_e$, a aresta f teria sido movida para C_2 quando e' entrasse em R_2.

Agora, note que os conjuntos K_e's são subconjuntos de M e são dois a dois disjuntos. Portanto, $|M| \geq \sum_{e \in R_1} |K_e| + |R_2| \geq (\delta - 1)|R_1| + |R_2|$ e o resultado segue. □

Teorema 4.5.3. Se M é um emparelhamento máximo, então Marcar-e-Matar é uma 2-aproximação para o 2PM(2) e uma $\frac{3\delta-5}{2\delta-4}$-aproximação para o 2PM($\delta$) para $\delta \geq 3$.

Demonstração. Primeiramente, note que um emparelhamento de segundo preço produzido por Marcar-e-Matar tem tamanho $|M| - |R_1| - |R_2|$ e que o tamanho de um emparelhamento máximo em G (isto é, $|M|$) é um limitante superior no tamanho de um emparelhamento de segundo preço máximo em G.

Pelo Lema 4.5.1, temos que $|M| \geq 2|R_1| + 3|R_2| \geq 2(|R_1| + |R_2|)$, e portanto $|M| - |R_1| - |R_2| \geq |M|/2$. Concluímos que Marcar-e-Matar é uma 2-aproximação para o 2PM(2).

Agora, pelos Lemas 4.5.1 e 4.5.2, temos que

$$\frac{|M| - |R_1| - |R_2|}{|M|} \geq \max\left\{1 - \frac{|R_1| + |R_2|}{2|R_1| + 3|R_2|}, 1 - \frac{|R_1| + |R_2|}{(\delta - 1)|R_1| + |R_2|}\right\}.$$

O valor mínimo na direita é atingido quando as duas expressões coincidem. Isso acontece, para $\delta \geq 3$, quando $2|R_1| + 3|R_2| = (\delta - 1)|R_1| + |R_2|$, isto é, quando $|R_2| = (\delta - 3)|R_1|/2$. A partir disso, concluímos que

$$\frac{|M| - |R_1| - |R_2|}{|M|} \geq 1 - \frac{(\delta - 1)|R_1|}{2(\delta - 1)|R_1| + (\delta - 3)|R_1|} = 1 - \frac{(\delta - 1)}{3\delta - 5} = \frac{2\delta - 4}{3\delta - 5},$$

e o resultado segue. □

Adiante no Teorema 4.6.1 mostramos que a análise é justa.

A tabela a seguir mostra a razão de aproximação específica para alguns valores de δ.

δ	2	3	4	5	6
razão	2	2	1,75	1,66	1,625

Observe que, conforme δ cresce, a razão de aproximação se aproxima de 1,5.

Como Marcar-e-Matar é um algoritmo especial, pelos Teoremas 4.2.1 e 4.5.3, concluímos o principal resultado desse capítulo.

Corolário 4.5.4. O algoritmo O-Marcar-e-Matar é uma 2-aproximação para o B2PAA(2) e uma $\frac{3\delta-5}{2\delta-4}$-aproximação para o B2PAA(δ) para $\delta \geq 3$.

4.6 A Análise do Marcar-e-Matar é Justa

Nessa seção provamos que a análise do algoritmo Marcar-e-Matar apresentada na Seção 4.5 é justa.

Teorema 4.6.1. Para cada $\delta \geq 3$, existe uma família $\{G_s \colon s = 1, \ldots\}$ de instâncias do 2PM(δ) e um emparelhamento de segundo preço E_s em G_s que pode ser produzido pelo algoritmo Marcar-e-Matar tal que a razão entre $|E_s|$ e o valor ótimo para G_s tende a $\frac{2\delta-4}{3\delta-5}$ quando s cresce.

Demonstração. Fixe um $\delta \geq 3$. Antes de descrever G_s, iremos descrever uma árvore T_s que está contida em G_s. A árvore T_s é definida recursivamente em s. Seu conjunto de vértices é bipartido nos conjuntos em I_s e B_s e o seu conjunto de itens que são folhas em I_s será denotado por F_s. Seu tamanho depende de s e δ.

A árvore T_1 consiste de apenas uma aresta, conectando os vértices i_1 e b_1. Para T_1, temos que $I_1 = F_1 = \{i_1\}$ e $B_1 = \{b_1\}$. Para $s > 1$, a árvore T_s contém $\delta - 2$ cópias disjuntas de T_{s-1}. (Veja a Figura 4.9(a).) O conjunto I_s consiste da união das $\delta - 2$ cópias de I_{s-1}, juntamente com dois novos vértices i_s e i'_s. O conjunto B_s consiste da união das $\delta - 2$ cópias de B_{s-1} juntamente com um vértice b_s e δ vértices $b'_{s,1}, \ldots, b'_{s,\delta}$. Além das arestas contidas nas cópias de T_{s-1}, temos que i_s é adjacente a b_s, a $b'_{s,1}$ e ao vértice b_{s-1} de cada cópia de T_{s-1}. Além disso, o vértice i'_s é adjacente a $b'_{s,1}, \ldots, b'_{s,\delta}$. O conjunto F_s é a união dos conjuntos F_{s-1} para as $\delta - 2$ cópias de T_{s-1}. Observe que o grau de cada vértice em $I_s \setminus F_s$ é δ em T_s.

Agora estamos prontos para descrever o grafo $G_s = (I, B, E)$ (veja a Figura 4.9(b)). O conjunto I consiste do conjunto I_s adicionado de δ novos vértices, formando o conjunto I' e $\lfloor |F_s|/2 \rfloor$ novos vértices, formando o conjunto I''. O conjunto B consiste de B_s adicionado de δ novos vértices, formando o conjunto B' e $\lfloor |F_s|/2 \rfloor$ novos vértices, formando o conjunto B''. Vamos definir as arestas de G_s. O subgrafo de G_s induzido por $I' \cup B'$ é um grafo bipartido completo em (I', B'). O subgrafo induzido por $I'' \cup B''$ é um emparelhamento perfeito M'' com arestas entre I'' e B''. Finalmente, cada vértice em I'' é adjacente a todos os vértices em B', cada vértice em B'' é adjacente a duas folhas em F_s e cada folha em F_s é adjacente a no máximo um vértice em B'' e a todos os vértices em B'. Completamos então a descrição de G_s, exceto pela ordem no conjunto I. Note que todos os vértices em I têm grau pelo menos δ em G_s.

A ordem nos vértices de I é apresentada a seguir. Primeiramente os vértices de I', seguido dos vértices de I''. Seguidos pelos vértices de I_s ordenados pelo nível (da direita para a esquerda) que eles aparecem na Figura 4.9(b), com os vértices i'_k aparecendo antes de todos os outros do mesmo nível.

Observe que existe um emparelhamento M_s em T_s cobrindo todos os vértices em I_s. De fato, M_s pode também ser definido de forma indutiva. O emparelhamento consiste da união dos emparelhamentos M_{s-1} dentro das cópias de T_{s-1} e das arestas (i_s, b_s) e $(i'_s, b'_{s,1})$ (as arestas azuis na Figura 4.9(a)). Além disso, existe um emparelhamento M em G_s que cobre todos os vértices em I. Basta tomar o emparelhamento M_s juntamente com o emparelhamento perfeito M'' (entre I'' e B'') e um emparelhamento perfeito no grafo bipartido completo com vértices (I', B'). (Veja as arestas azuis na Figura 4.9(b).) Note que, na ordenação escolhida, não existe aresta para baixo incidente nos vértices em $I \setminus I'$.

Suponha que Marcar-e-Matar receba G_s e M como entrada. Note que, nesse caso, cada arestas $(i'_k, b'_{k,1})$ irá para R_1, para todo k e todas as cópias de T_k. De fato, essa aresta é um competidor do topo da aresta (i_k, b_k). Além disso, todas as arestas no emparelhamento M'' irão para o conjunto R_2. De fato, cada aresta desta tem duas marcas, feitas pelas arestas em M incidentes às folhas de T_s. A aresta de M incidente ao $(\delta - 2)$-ésimo vértice de U' também termina em R_2, por causa das marcas feitas pelas outras aresta de M incidentes aos vértices de I'.

Para calcular o tamanho de um emparelhamento de segundo preço produzido pelo algoritmo, vamos contar o número de arestas que terminaram em R_1. Para isso, é possível mostrar por indução que, em I_s, para $\delta > 3$, existem $\frac{(\delta-2)^{s-1}-1}{\delta-3}$ vértices i'_k para todo k e para todas as cópias de T_k

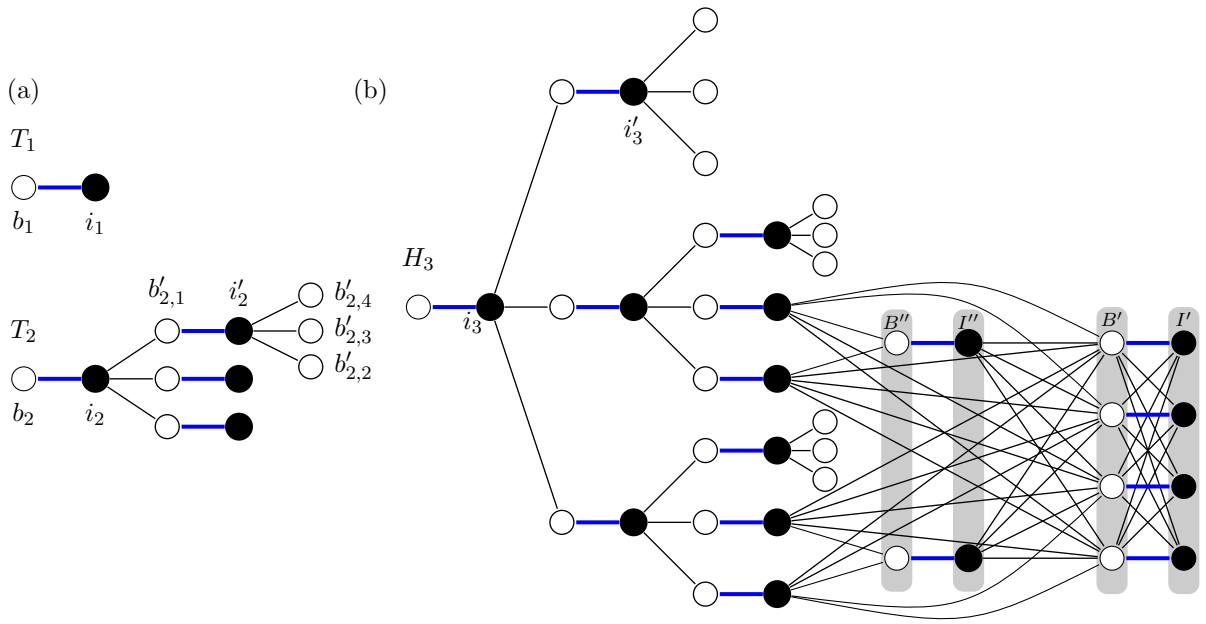


Figura 4.9: *A construção de uma instância que mostra um limitante inferior da razão de aproximação do algoritmo* Marcar-e-Matar. *(a) Árvores T_1 e T_2 para $\delta = 4$. Os conjuntos I_i e B_i são indicados pelos vértices pretos e brancos respectivamente. As arestas azuis (grossas) formam um emparelhamento em T_s que cobre I_s. (b) O grafo G_3 para $\delta = 4$, com I indicado pelos vértices pretos e B indicado pelos vértices brancos.*

e para $\delta = 3$ existem $s - 1$ tais vértices. De forma similar, podemos demonstrar que o número de vértices em F_s é $(\delta - 2)^{s-1}$ e que o número de arestas em M é $\frac{2(\delta-2)^{s-1}-2}{\delta-3} + \lfloor \frac{3(\delta-2)^{s-1}}{2} \rfloor + \delta$ se $\delta > 3$ e $2(s-1) + 1 + \delta$ se $\delta = 3$. Portanto o emparelhamento de segundo preço produzido por Marcar-e-Matar tem tamanho

$$|M| - \left(\frac{(\delta-2)^{s-1}-1}{\delta-3}\right) - \left\lfloor \frac{(\delta-2)^{s-1}}{2} \right\rfloor - 1 = \frac{(\delta-2)^{s-1}-1}{\delta-3} + (\delta-2)^{s-1} + \delta - 1$$

se $\delta > 3$ e

$$|M| - (s-1) - 1 = s - 1 + \delta$$

se $\delta = 3$.

Por outro lado, seja M^* um emparelhamento obtido de M removendo a aresta incidente no primeiro vértice de I' e substituindo cada aresta $(i'_k, b'_{k,1})$ pela aresta $(i'_k, b'_{i,2})$ e a aresta (i_k, b_k) pela aresta $(i_k, b'_{k,1})$. Obviamente, $|M^*| = |M| - 1$. Além disso, M^* é um emparelhamento de segundo preço. Portanto, a razão obtida por Marcar-e-Matar para essa instância é no máximo

$$\frac{\frac{(\delta-2)^{s-1}-1}{\delta-3} + (\delta-2)^{s-1} + \delta - 1}{\frac{2(\delta-2)^{s-1}-2}{\delta-3} + \lfloor \frac{3(\delta-2)^{s-1}}{2} \rfloor + \delta - 1} \leq \frac{2\delta - 4 + \frac{2(\delta-1)(\delta-3)-2}{(\delta-2)^{s-1}}}{3\delta - 5 + \frac{2(\delta-1)(\delta-3)-4}{(\delta-2)^{s-1}}}$$

se $\delta > 3$, que tende para $\frac{2\delta-4}{3\delta-5}$ quando s cresce, e

$$\frac{s - 1 + \delta}{2(s-1) + \delta}$$

para $\delta = 3$, que tende para $1/2$ quando s cresce, de onde o resultado segue. □

4.7 Análise do Uso de Emparelhamentos Máximos

A natureza do 2PM nos fez considerar duas variantes do Marcar-e-Matar que utilizam emparelhamentos iniciais diferentes. A primeira variante utiliza um emparelhamento máximo M arbitrário. A ideia por trás da outra variante é tentar fazer o algoritmo que escolhe M incluir arestas "melhores" em M. Isso foi feito atribuindo pesos a cada aresta do grafo de entrada $G = (I, B, E)$ do 2PM, de forma que o peso de uma resta (i, b) é maior do que o peso de uma aresta (i', b) se $i > i'$. Especificamente, para cada comprador b, se os itens adjacentes a b em ordem crescente são $i_1, \ldots, i_d$ então atribuímos o peso $w(i_k, b) = k$ para $k = 1, \ldots, d$. A Figura 4.10 mostra tal atribuição para uma determinada instância do 2PM. A escolha desses pesos tenta capturar a impressão que, se existe um comprador b interessado em muitos itens, se atribuirmos b para o item de índice maior possível, b está disponível para ser um comprador de segundo preço para mais itens. Essa variante do Marcar-e-Matar usa essa atribuição de pesos selecionando M como um emparelhamento máximo em G de peso máximo. A razão de aproximação das duas variantes é dada pelo Teorema 4.5.3, já que elas utilizam um emparelhamento máximo.

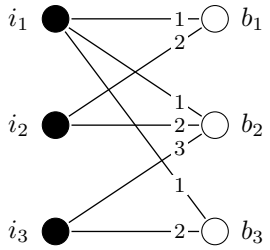


Figura 4.10: *Heurística de atribuição de pesos para instâncias do* 2PM.

Para avaliar as duas variantes do Marcar-e-Matar, nós as implementamos, juntamente com o algoritmo ReverseMatching, e as testamos em instâncias geradas aleatoriamente.

Definição. Seja n um inteiro positivo e δ tal que $0 \leq \delta \leq n$. Denotamos por $D_{n,\delta} = (I, B, E)$ um grafo aleatório com $|I| = |B| = n$ onde o grau de cada item i em I é δ e a vizinhança de i é escolhida independente e uniformemente ao acaso entre todos os subconjuntos de B de tamanho δ.

Referimo-nos à primeira variante simplesmente por Marcar-e-Matar e à segunda como Marcar-e-Matar$_w$. Denotamos por $\mathrm{MK}(G)$ (e $\mathrm{MK}_w(G)$ e $\mathrm{RM}(G)$) a razão entre o tamanho de um emparelhamento de segundo preço produzido por Marcar-e-Matar (Marcar-e-Matar$_w$ e o algoritmo ReverseMatching apresentado por Azar et al. [ABKN09], respectivamente) e o tamanho de uma solução ótima para G.

As soluções ótimas para as instâncias geradas foram encontradas utilizando o CPLEX com a seguinte formulação inteira para o 2PM. Na formulação (F2PM), temos um vetor binário x de variáveis com $|E|$ posições. Para todo $(i, b) \in E$, x_{ib} é 1 se e somente se o item i é alocado ao comprador b. Além disso, para todo $i \in I$, denotamos por $N(i)$ o conjunto de vizinhos de i em G. O objetivo é determinar x que

$$
\begin{array}{llrcll}
\text{(F2PM)} & \text{maximize} & \displaystyle\sum_{(i,b)\in E} x_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{(i,b)\in E} x_{ib} & \leq & 1, & \forall b \in B \quad (11) \\
& & \displaystyle\sum_{(i,b)\in E} x_{ib} & \leq & 1, & \forall i \in I \quad (12) \\
& & \displaystyle\sum_{\substack{(i',b)\in E \\ i'<i \,\wedge\, (i,b)\in E}} x_{i'b} & \leq & |N(i)| - 2\displaystyle\sum_{(i,b)\in E} x_{ib}, & \forall i \in I \quad (13) \\
& & x_{ib} & \in & \{0,1\}, & \forall (i,b) \in E. \quad (14)
\end{array}
$$

Lema 4.7.1. (F2PM) é uma formulação para o 2PM.

Demonstração. Fixe uma instância G do 2PM e seja S um emparelhamento de segundo preço para G. Defina, para todo $(i,b) \in E$, $x_{ib} = 1$ se $(i,b) \in S$ e $x_{ib} = 0$ caso contrário. Afirmamos que x é uma solução viável para (F2PM) de valor $|S|$. De fato, basta provar que a restrição (13) está satisfeita para todo item i. Se o item i não é vendido em S, então a restrição (13) está trivialmente satisfeita já que $|\{(i',b) \in S : i' < i \ \wedge \ (i,b) \in E\}| \leq |N(i)|$. Por outro lado, se o item i é vendido em S, então existem pelo menos dois compradores interessados em i que não receberam um item $i' < i$ (o comprador de i e o comprador de segundo preço para i). Portanto, existem no máximo $|N(i)| - 2$ compradores interessados em i que compraram um item $i' < i$.

Considere agora uma solução x do (F2PM). Note que o conjunto $S = \{(i,b) \in E : x_{ib} = 1\}$ é um emparelhamento de segundo preço de tamanho $\sum_{ib \in E} x_{ib}$. Isso ocorre porque as desigualdades (11) e (12), juntamente com (14), garantem que S é um emparelhamento em G e a desigualdade (13) garante que existe (pelo menos) um comprador de segundo preço para todo item vendido. □

Geramos 50 instâncias do $D_{n,\delta}$ com n variando entre 100 e 1000 em passos de tamanho 100 e usamos $\delta = 3$ e $\delta = 10$. Nas Tabelas 4.1 e 4.2, apresentamos o valor médio de MK(·) e das razões $\text{MK}_w(\cdot)/\text{MK}(\cdot)$ e MK(·)/RM(·) e, entre colchetes, o valor mínimo e o valor máximo de MK(·) e das razões. Note que, pelo Teorema 4.5.3, $\text{MK}(D_{n,\delta})$ e $\text{MK}_w(D_{n,\delta})$ é pelo menos 0,5 para $\delta = 3$ e 0,64 para $\delta = 10$, e $\text{RM}(D_{n,\delta})$ é pelo menos 0,5 para qualquer $\delta \geq 2$ [ABK+08]. Na Figura 4.11, representamos essas médias em um gráfico.

Tamanho	$\text{MK}(D_{n,3})$	$\text{MK}_w(D_{n,3})/\text{MK}(D_{n,3})$	$\text{MK}(D_{n,3})/\text{RM}(D_{n,3})$
100	0,887 [0,835 - 0,928]	1,042 [0,987 - 1,096]	1,188 [1,110 - 1,291]
200	0,883 [0,851 - 0,920]	1,038 [1,006 - 1,077]	1,186 [1,127 - 1,254]
300	0,879 [0,840 - 0,901]	1,045 [1,017 - 1,086]	1,190 [1,131 - 1,263]
400	0,882 [0,855 - 0,900]	1,044 [1,010 - 1,064]	1,185 [1,126 - 1,237]
500	0,876 [0,856 - 0,902]	1,051 [1,028 - 1,074]	1,186 [1,142 - 1,253]
600	0,878 [0,859 - 0,898]	1,047 [1,015 - 1,077]	1,187 [1,153 - 1,236]
700	0,877 [0,858 - 0,895]	1,046 [1,028 - 1,075]	1,193 [1,162 - 1,235]
800	0,880 [0,865 - 0,891]	1,042 [1,019 - 1,062]	1,187 [1,141 - 1,229]
900	0,879 [0,863 - 0,903]	1,043 [1,025 - 1,065]	1,185 [1,152 - 1,207]
1000	0,874 [0,856 - 0,887]	1,051 [1,030 - 1,071]	1,189 [1,167 - 1,216]

Tabela 4.1: *Resultados para $D_{n,3}$.*

Para os valores de δ usados, Marcar-e-Matar$_w$ perde apenas para o Marcar-e-Matar em pequenas instâncias e por, no máximo, 1,3%. Em média, Marcar-e-Matar$_w$ foi melhor para todo n, produzindo emparelhamentos de segundo preço que eram aproximadamente 4,5% maiores do que aqueles produzidos pelo Marcar-e-Matar para $\delta = 3$ e aproximadamente 2,6% maiores para $\delta = 10$. Até esse momento, não sabemos se Marcar-e-Matar$_w$ tem uma razão de aproximação melhor do que Marcar-e-Matar para 2PM(δ) com $\delta > 2$. Ambas as variantes tiveram resultados melhores do que o algoritmo ReverseMatching proposto por Azar et al. [ABKN09] em todas as instâncias que testamos.[2]

A seguir provaremos que, escolher um emparelhamento de segundo preço contido em um emparelhamento máximo como solução não pode levar a uma α-aproximação para o 2PM(2) onde $\alpha < 2$. Portanto, todo algoritmo especial, em particular o Marcar-e-Matar e o Marcar-e-Matar$_w$, não tem uma razão de aproximação melhor do que 2 para o 2PM(2).

[2]Note que o algoritmo ReverseMatching não foi projetado para explorar mercados δ-competitivos e, portanto, é necessário levar esse fato em consideração ao analisar os resultados empíricos apresentados nessa seção.

Tamanho	$\mathrm{MK}(D_{n,10})$	$\mathrm{MK}_w(D_{n,10})/\mathrm{MK}(D_{n,10})$	$\mathrm{MK}(D_{n,10})/\mathrm{RM}(D_{n,10})$
100	0,961 [0,929 - 0,990]	1,022 [0,990 - 1,054]	1,231 [1,134 - 1,329]
200	0,957 [0,944 - 0,975]	1,027 [1,010 - 1,048]	1,244 [1,140 - 1,394]
300	0,956 [0,939 - 0,970]	1,029 [1,007 - 1,050]	1,222 [1,114 - 1,311]
400	0,956 [0,944 - 0,972]	1,027 [1,010 - 1,043]	1,223 [1,131 - 1,305]
500	0,956 [0,942 - 0,966]	1,026 [1,013 - 1,039]	1,218 [1,151 - 1,331]
600	0,957 [0,946 - 0,968]	1,026 [1,014 - 1,043]	1,225 [1,157 - 1,299]
700	0,955 [0,947 - 0,963]	1,027 [1,016 - 1,036]	1,212 [1,155 - 1,277]
800	0,955 [0,948 - 0,965]	1,027 [1,017 - 1,036]	1,221 [1,155 - 1,298]
900	0,956 [0,950 - 0,962]	1,026 [1,017 - 1,035]	1,216 [1,157 - 1,271]
1000	0,955 [0,950 - 0,965]	1,027 [1,018 - 1,035]	1,219 [1,151 - 1,290]

Tabela 4.2: *Resultados para* $D_{n,10}$.

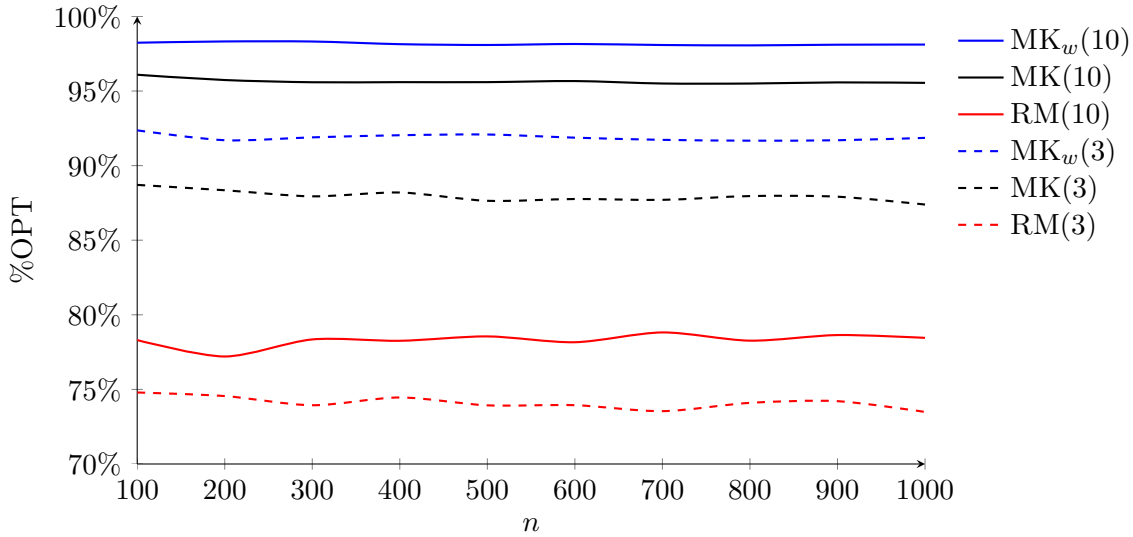


Figura 4.11: *Comparação para* $D_{n,\delta}$. *As linhas* $\mathrm{MK}(\delta)$, $\mathrm{MK}_w(\delta)$ *e* $\mathrm{RM}(\delta)$ *denotam a média de* $\mathrm{MK}(D_{n,\delta})$, $\mathrm{MK}_w(D_{n,\delta})$ *e* $\mathrm{RM}(D_{n,\delta})$, *respectivamente.*

Teorema 4.7.2. Para todo $s > 0$, existe uma instância do 2PM(2) com $s + 3$ itens onde o emparelhamento de segundo preço máximo tem tamanho $s+1$ e todo emparelhamento de segundo preço contido em um emparelhamento máximo tem tamanho no máximo $\lceil \frac{s}{2} \rceil + 2$. Isto é, toda α-aproximação para o 2PM(2) que devolve um emparelhamento de segundo preço contido em um emparelhamento máximo tem $\alpha \geq 2$.

Demonstração. Para cada $s > 0$, construímos uma instância $G = (I, B, E)$ com $I = \{1, \ldots, s + 3\}$ e $B = \{1, \ldots, s + 3\}$. Para $1 \leq i \leq s$, temos que o item i é adjacente apenas aos compradores i e $i + 1$, o item $s + 1$ é adjacente aos compradores 1 e $s + 2$ e os itens $s + 2$ e $s + 3$ são ambos adjacentes aos compradores $s + 2$ e $s + 3$. A Figura 4.12 mostra tal instância para $s = 2$.

Primeiramente, note que, em um emparelhamento de segundo preço, os itens $s + 2$ e $s + 3$ não podem estar ambos emparelhados. Considere agora um emparelhamento de segundo preço onde o item $s+1$ está emparelhado. Então o comprador 1 não está emparelhado com o item 1. Se o item 1 está emparelhado, então precisa estar emparelhado com o comprador 2. Nesse caso, o item 2 não pode estar emparelhado. Concluímos que, se o item $s+1$ está emparelhado então os itens 1 e 2 não podem estar ambos emparelhados. Isso implica que um emparelhamento de segundo preço em G tem tamanho no máximo $s + 1$. Agora, note que o emparelhamento que emparelha o item i com

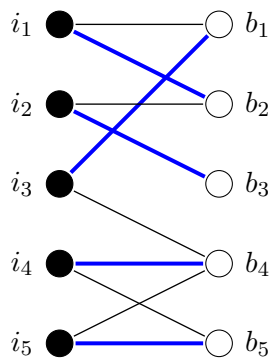


Figura 4.12: *O grafo G da prova do Teorema 4.7.2 para $s = 2$. As arestas azuis (grossas) indicam um dos dois emparelhamentos máximos de G. O outro emparelhamento máximo emparelha i_4 com b_5 e i_5 com b_4.*

o comprador i para $1 \leq i \leq s$ e o item $s + 2$ com o comprador $s + 2$ é um emparelhamento de segundo preço máximo já que tem tamanho $s + 1$.

Agora note que existem dois emparelhamentos máximos em G. Um deles emparelha o item i ao comprador $i + 1$ para $1 \leq i \leq s$, emparelha o item $s + 1$ ao comprador 1 e emparelha $s + 2$ e $s + 3$ aos compradores $s + 2$ e $s + 3$, respectivamente. O outro emparelhamento é idêntico para os itens em $\{1, \ldots, s + 1\}$, mas emparelha o item $s + 2$ com o comprador $s + 3$ e o item $s + 3$ com o comprador $s + 2$.

Concluímos a prova apontando o fato de que se o item i é emparelhado com o comprador $i + 1$ em um emparelhamento de segundo preço, para $2 \leq i \leq s$, então o item $i - 1$ não pode estar emparelhado. Portanto, concluímos que tal emparelhamento de segundo preço emparelha apenas $\lceil \frac{s}{2} \rceil$ dos itens em $\{1, \ldots, s\}$. Se o item $s+2$ não está emparelhado, então os itens $s+1$ e $s+3$ podem ser emparelhados aos compradores 1 e $s+3$, porque ambos têm o comprador $s+2$ como um comprador de segundo preço. Se o item $s + 2$ está emparelhado, o item $s + 3$ não pode estar. Um argumento similar funciona para o outro emparelhamento máximo. Portanto, a partir desses emparelhamentos máximos, obtemos emparelhamentos de segundo preço de tamanho no máximo $\lceil \frac{s}{2} \rceil + 2$.

A razão entre o tamanho de um emparelhamento de segundo preço máximo para G e qualquer emparelhamento de segundo preço para G contido em um emparelhamento máximo (de G) é no máximo $(\lceil \frac{s}{2} \rceil + 2)/(s + 1)$, o que tende a $1/2$ quando s cresce. A partir disso, concluímos que toda α-aproximação para o 2PM(2) que devolve um emparelhamento de segundo preço contido em um emparelhamento máximo tem $\alpha \geq 2$. □

Em vista do resultado do Teorema 4.7.2, analisamos uma terceira variante do Marcar-e-Matar que utiliza como emparelhamento inicial um emparelhamento de custo máximo que não é, necessariamente, de cardinalidade máxima. Isto é, essa variante não esbarra na dificuldade apresentada pelo Teorema 4.7.2, mas, por outro lado, não se beneficia do Teorema 4.5.3. Infelizmente, essa terceira variante perdeu em comparação com as outras duas nos testes empíricos. Além disso, como mostrado a seguir no Lema 4.7.3, existe uma família de instâncias onde existe um emparelhamento de custo máximo tal que a cardinalidade de tal emparelhamento tende a metade da cardinalidade de um emparelhamento de segundo preço ótimo e, portanto, essa variante não pode ter uma razão de aproximação melhor do que 2 para o 2PM(2).

Lema 4.7.3. Existe uma família $\{G_k : k = 1, \ldots\}$ de instâncias do 2PM tal que existe um emparelhamento de custo máximo de G_k de tamanho k e um emparelhamento de segundo preço de G_k de tamanho $2k - 1$.

Demonstração. Para cada inteiro positivo k, seja $G_k = (I_k, B_k, E_k)$ tal que $I_k = \{1, \ldots, 2k - 1\}$, $B_k = \{b_j^i : 1 \leq j \leq 3 \text{ e } k \leq 2k - 1\}$ e E_k é descrito a seguir. Para cada item i tal que $1 \leq i \leq k-1$, temos que i é adjacente a b_1^{k+i-1} e b_1^{k+i} e, para cada item i tal que $k \leq i \leq 2k - 1$, temos que i é adjacente a b_1^i, b_2^i e b_3^i. A Figura 4.13 exemplifica tal construção para $k = 4$.

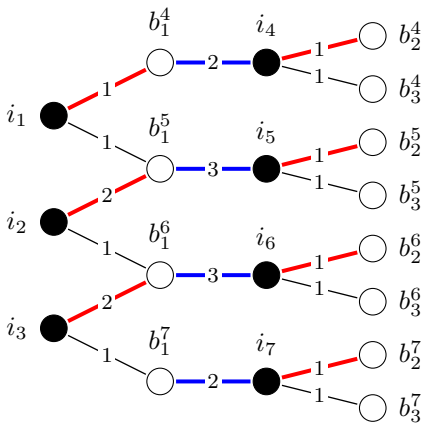


Figura 4.13: *Uma instância onde um emparelhamento de custo máximo não é um bom emparelhamento de segundo preço. As arestas em azul (grossas) indicam um emparelhamento de custo máximo que também é um emparelhamento de segundo preço. As arestas em vermelho (grossas) indicam um emparelhamento de segundo preço de tamanho máximo.*

Em primeiro lugar, considere o emparelhamento onde um item i está emparelhado com o comprador b_1^{k+i-1} se $1 \leq i \leq k-1$ e i está emparelhado com o comprador b_2^i se $k \leq i \leq 2k-1$. Note que tal emparelhamento (indicado pelas arestas vermelhas da Figura 4.13) é um emparelhamento de segundo preço de cardinalidade $2k-1$.

Em segundo lugar, note que o emparelhamento M onde o item i, para $k \leq i \leq 2k-1$, está emparelhado com o comprador b_1^i (e os outros itens não estão emparelhados) é um emparelhamento de segundo preço de cardinalidade k. Tal emparelhamento é indicado pelas arestas azuis da Figura 4.13.

Afirmamos que M é um emparelhamento de custo máximo em G_k. De fato, considere um emparelhamento de custo máximo M^* com o maior número de aresta em comum com M e suponha que exista um item i, como $k \leq i \leq 2k-1$ tal que (i, b_1^i) não pertença a M^*. Note que podemos remover as arestas de M^* adjacentes a i e a b_1^i e adicionar a aresta (i, b_1^i) e obter um emparelhamento de custo igual ao custo de M^* com uma aresta em comum a mais com M, contradizendo a escolha de M^*. Portanto, M é um emparelhamento de custo máximo em G_k. □

4.8 Emparelhamento de Segundo Preço Online

Azar et al. [ABKN09] propuseram um algoritmo aleatorizado para o 2PM denominado RankingSimulate. Nessa seção, revisamos a análise de tal algoritmo para derivar uma razão em termos de δ para o 2PM(δ) online.

Primeiramente, apresentamos o algoritmo RankingSimulate. Novamente, denotamos por $N(i)$ o conjunto de vizinhos de i em G.

```
RankingSimulate(G)
1  M = ∅
2  R = ∅
3  Escolha uma permutação aleatória σ de B
4  Quando o item i chegar:
5      Seja V o conjunto de vizinhos de i que não estão em M ou R
6      se |V| = 1
7          Seja b o (único) comprador em N(i)
8          Escolha uniformemente ao acaso entre as linhas 9 e 10 e execute:
9              Venda i para b e adicione b em M
10             Adicione b em R
11     senão se |V| ≥ 2
12         Sejam b1 e b2 dois compradores em N(i) com σ mínimo
13         Escolha uniformemente ao acaso entre as linhas 14 e 15 e execute:
14             Venda i para b1 e adicione b1 em M e b2 em R
15             Venda i para b2 e adicione b2 em M e b1 em R
```

Para a conveniência de sua análise, RankingSimulate produz um emparelhamento que não necessariamente é um emparelhamento de segundo preço. Dizemos que um comprador emparelhado em um emparelhamento encontrado pelo RankingSimulate fornece lucro 1 se o comprador é emparelhado com um item que tem um comprador de segundo preço. Caso contrário, dizemos que o comprador fornece lucro 0.

Azar et al. [ABKN09] provaram que RankingSimulate é $2\sqrt{e}/(\sqrt{e}-1)$-competitivo. A análise feita por eles pode ser fortalecida para o 2PM(δ) online com $\delta > 2$. De fato, na prova, ele mostraram que a probabilidade de um comprador emparelhado obter lucro é pelo menos $\frac{1}{2}$. Para isso, eles utilizaram apenas o fato de que cada item tem pelo menos dois compradores adjacentes. Usando o limitante δ no grau dos itens, obtemos o seguinte.

Lema 4.8.1. Seja P_b o lucro obtido pelo comprador b em um emparelhamento produzido pelo RankingSimulate(G), onde G é uma instância do 2PM(δ). Vale que

$$\mathbb{E}(P_b | b \text{ está emparelhado}) \geq 1 - \left(\frac{1}{2}\right)^{\delta-1}.$$

Demonstração. Suponha que o comprador b está emparelhado com o item i. Existem $\delta - 1$ outros compradores adjacentes a i e b fornece lucro 1 se pelo menos um deles não está emparelhado. Note que RankingSimulate emparelha um comprador com probabilidade no máximo $\frac{1}{2}$. Portanto, a probabilidade de que esses $\delta - 1$ compradores estejam todos emparelhados é no máximo $\left(\frac{1}{2}\right)^{\delta-1}$. Portanto, concluímos que $\mathbb{E}(P_b | b \text{ está emparelhado}) \geq 1 - \left(\frac{1}{2}\right)^{\delta-1}$. □

A seguir apresentamos outro lema que será usado na prova do nosso próximo teorema.

Lema 4.8.2 (Azar et al. [ABKN09])**.** O tamanho esperado de um emparelhamento produzido por RankingSimulate(G) é pelo menos $(1 - 1/\sqrt{e} + o(1))\mathrm{OPT}$, onde OPT é o tamanho de um emparelhamento de segundo preço máximo em G.

Combinando os Lemas 4.8.1 e 4.8.2, podemos provar o teorema a seguir.

Teorema 4.8.3. RankingSimulate é $\left(\frac{2^{\delta-1}}{2^{\delta-1}-1}\,\frac{\sqrt{e}}{\sqrt{e}-1}\right)$-competitivo para 2PM($\delta$).

Demonstração. Combinamos os Lemas 4.8.1 e 4.8.2 para derivar um limitante inferior no lucro

esperado sobre todos os compradores:

$$\begin{aligned}
\sum_{b \in B} \mathbb{E}(P_b) &= \sum_{b \in B} \mathbb{E}(P_b | b \text{ está emparelhado})\, \mathbb{P}(b \text{ está emparelhado}) \\
&\geq \left(1 - \left(\frac{1}{2}\right)^{\delta - 1}\right) \sum_{b \in B} \mathbb{P}(b \text{ está emparelhado}) \\
&\geq \left(1 - \left(\frac{1}{2}\right)^{\delta - 1}\right) \left(1 - \frac{1}{\sqrt{e}} + o(1)\right) \mathrm{OPT}.
\end{aligned}$$

□

Acreditamos que a razão de competitividade do RankingSimulate não é justa para o 2PM(δ), já que utilizamos o grau de um item apenas para calcular a probabilidade de obter lucro de um comprador emparelhado.

Capítulo 5

Conclusão

Nessa tese consideramos três problemas diferentes, o Problema da Compra Máxima com Oferta Limitada, o Problema da Precificação Livre de Inveja (com demanda unitária) e, por fim, o Leilão de Anúncios de Segundo Preço com Lances Binários.

Acreditamos que problemas de precificação com oferta limitada são bem interessantes porque tratam de uma restrição realista e são uma generalização dos problema de precificação com oferta ilimitada. Assim, para o Problema da Compra Máxima com Oferta Limitada, consideramos o caso onde exigimos uma escada de preços e o caso onde não temos tal restrição. Nossos resultados melhoram as razões de aproximação previamente conhecidas para ambos os problemas (com e sem a restrição da escada de preços). Além disso, é possível que a técnica utilizada no Teorema 2.3.3 de enumerar todas as possíveis alocações de itens a compradores possa ajudar também em outros problemas de precificação.

Alguns problemas continuam em aberto. É interessante notar que o nosso algoritmo de aproximação para o Problema da Compra Máxima com Oferta Limitada tem a mesma razão de aproximação apresentada por Aggarwal et al. [AFMZ04] para o Problema da Compra Máxima (com oferta ilimitada) embora sejam algoritmos bem diferentes. Seria interessante projetar um algoritmo com razão de aproximação melhor do que $e/(e-1)$ para o Problema da Compra Máxima com Oferta Limitada (ou, até menos, com oferta ilimitada) ou encontrar um limitante inferior melhor que $16/15$ na razão de aproximação de qualquer algoritmo para esse problema.

No caso do Problema da Compra Máxima com Oferta Limita e com Escada de Preços, seria interessante desenvolver um PTAS (já que existe um PTAS para o caso de oferta ilimitada [AFMZ04]) ou provar que esse problema é APX-difícil. Além disso, note que a escada de preços não nos ajudou a obter melhores razões de aproximação como acontece para o caso de oferta ilimitada (para o qual existe um PTAS). Isto é, de certa forma, contra a nossa intuição de que saber a ordem dos preços tornaria mais fácil encontrar boas precificações. Não sabemos se isso é algo intrínseco ao problema ou se existem outras formas de explorar a restrição de escada de preços para obter melhores aproximações para o problema.

Para o Problema da Precificação Livre de Inveja, apresentamos quatro novas formulações MIP juntamente com resultados empíricos que comparam tais formulações. Nossos resultados mostram que as novas formulações são melhores do que a formulação previamente conhecida na literatura.

Adicionalmente, apresentamos um novo resultado mostrando a dificuldade em aproximar uma variante do problema. O fato que restringir os preços para serem escolhidos a partir de uma série geométrica não deixa o problema mais fácil de aproximar é interessante já que apresenta mais evidências de quão difícil é atacar esse problema do ponto de vista teórico.

Também apresentamos três modelos para gerar instâncias para leilões de demanda unitária. Até onde sabemos, esses são os primeiros geradores para esses leilões na literatura. Além disso, disponibilizamos tais geradores como software livre para poderem ser usados em outros estudos.

Por fim, consideramos o Leilão de Anúncios de Segundo Preço com Lances Binários, uma generalização do Emparelhamento de Segundo Preço. Mostramos como obter aproximações para o

primeiro a partir de algoritmos específicos para o segundo.

Além disso, focamos em mercados competitivos, analisando as razões de aproximação dos algoritmos previamente propostos para as versões online e offline do Emparelhamento de Segundo Preço bem como a sua complexidade. Ademais, propusemos uma melhor aproximação para o Emparelhamento de Segundo Preço quando todos os itens têm grau pelo menos 4 e essa aproximação também pode ser usada para obter uma aproximação para o Leilão de Anúncios de Segundo Preço com Lances Binários.

Ainda existe uma grande distância entre o melhor limitante inferior e o melhor limitante superior para a razão de aproximação do Emparelhamento de Segundo Preço, inclusive em mercados competitivos. Em particular, é concebível que a segunda variante do Marcar-e-Matar apresentada na Seção 4.7 tenha uma razão de aproximação melhor do que a razão do Marcar-e-Matar para o 2PM(δ) com $\delta > 2$.

Além disso, seria interessante projetar um algoritmo de aproximação para o Emparelhamento de Segundo Preço que não utilize um emparelhamento máximo, como, por exemplo, fizemos na nossa terceira variante do Marcar-e-Matar (também apresentada na Seção 4.7). Idealmente, deveríamos ser capazes de encontrar emparelhamentos grandes que levassem a grandes emparelhamentos de segundo preço. Seria interessante também considerar a versão online do Leilão de Anúncios de Segundo Preço com Lances Binários.

Por fim, acreditamos que essa tese seja uma contribuição interessante para a área, já que obtivemos diversos resultados teóricos interessantes e também abordamos alguns aspectos práticos.

Apêndice A

Formulações para o Problema proposto por Shioda et al.

Nesse apêndice apresentamos a formulação originalmente proposta por Shioda et al. [STM11] para a generalização do Problema da Precificação Livre de Inveja mencionada na Seção 3.1 juntamente com as modificações necessárias para adaptar nossas formulações para tal problema. Para uma dada formulação (F), denotamos por (F_δ) a formulação baseada em (F) para essa generalização. Optamos por omitir as provas de que tais programas inteiros mistos são de fato formulações para o problema considerado.

$$
\begin{array}{llrcll}
(\mathrm{STM}_\delta) & \text{maximize} & \displaystyle\sum_{b\in B}\sum_{i\in I}\hat{p}_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & \displaystyle\sum_{i\in I\setminus\{k\}} (v_{ib}x_{ib} - \hat{p}_{ib}) & \geq & (v_{kb}+\delta_b)\displaystyle\sum_{i\in I\setminus\{k\}} x_{ib} \;-\; p_k & \forall k \in I, \forall b \in B \\
& & v_{ib}x_{ib} - \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \leq & p_i, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & p_i - R_i(1-x_{ib}), & \forall b \in B, \forall i \in I \\
& & x_{ib} & \in & \{0,1\}, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & p_i & \geq & 0, & \forall i \in I.
\end{array}
$$

$$
\begin{array}{llrcll}
(\mathrm{A}_\delta) & \text{maximize} & \displaystyle\sum_{b\in B}\sum_{i\in I} N_b\hat{p}_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & \displaystyle\sum_{i\in I}(v_{ib}x_{ib} - \hat{p}_{ib}) & \geq & v_{kb} - p_k + \delta_b \displaystyle\sum_{i\in I\setminus\{k\}} x_{ib}, & \forall k \in I, \forall b \in B \\
& & v_{ib}x_{ib} - \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \leq & p_i, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & p_i - R_i(1-x_{ib}), & \forall b \in B, \forall i \in I \\
& & x_{ib} & \in & \{0,1\}, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & p_i & \geq & 0, & \forall i \in I
\end{array}
$$

$$
\begin{array}{llrcll}
(\mathrm{R}_\delta) & \text{maximize} & \displaystyle\sum_{b\in B}\sum_{i\in I} N_b \hat{p}_{ib} & & & \\
& \text{sujeito a} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & \displaystyle\sum_{i\in I}(v_{ib}x_{ib} - \hat{p}_{ib}) & \geq & v_{kb} - p_k + \delta_b \displaystyle\sum_{i\in I\setminus\{k\}} x_{ib}, & \forall k \in I, \forall b \in B \\
& & v_{ib}x_{ib} - \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & p_i - R_i(1 - x_{ib}), & \forall b \in B, \forall i \in I \\
& & x_{ib} & \in & \{0,1\}, & \forall b \in B, \forall i \in I \\
& & \hat{p}_{ib} & \geq & 0, & \forall b \in B, \forall i \in I \\
& & p_i & \geq & 0, & \forall i \in I
\end{array}
$$

$$
\begin{array}{llrcll}
(\mathrm{L}_\delta) & \text{maximize} & \displaystyle\sum_{b\in B} N_b \ell_b & & & \\
& \text{sujeito a} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & \displaystyle\sum_{i\in I} v_{ib}x_{ib} - \ell_b & \geq & v_{kb} - p_k + \delta_b \displaystyle\sum_{i\in I\setminus\{k\}} x_{ib}, & \forall k \in I, \forall b \in B \\
& & \displaystyle\sum_{i\in I} v_{ib}x_{ib} - \ell_b & \geq & 0, & \forall b \in B \\
& & \ell_b & \geq & p_i - R_i(1 - x_{ib}), & \forall b \in B, \forall i \in I \\
& & x_{ib} & \in & \{0,1\}, & \forall b \in B, \forall i \in I \\
& & p_i & \geq & 0, & \forall i \in I \\
& & \ell_b & \geq & 0, & \forall b \in B.
\end{array}
$$

$$
\begin{array}{llrcll}
(\mathrm{U}_\delta) & \text{maximize} & \displaystyle\sum_{i\in I}\sum_{b\in B} N_b v_{ib} x_{ib} - \sum_{b\in B} N_b u_b & & & \\
& \text{subject to} & \displaystyle\sum_{i\in I} x_{ib} & \leq & 1, & \forall b \in B \\
& & u_b & \geq & v_{ib} - p_i + \delta_b \displaystyle\sum_{i'\in I\setminus\{i\}} x_{i'b}, & \forall i \in I, \forall b \in B \\
& & u_b & \leq & v_{ib}x_{ib} - p_i + (1 - x_{ib})(R_i + S_b), & \forall i \in I \\
& & u_b & \leq & \displaystyle\sum_{i\in I} v_{ib}x_{ib}, & \forall b \in B \\
& & u_b & \geq & 0, & \forall b \in B \\
& & p_i & \geq & 0, & \forall i \in I \\
& & x_{ib} & \in & \{0,1\}, & \forall i \in I, \forall b \in B.
\end{array}
$$

Apêndice B

Resultados para o Problema da Precificação Livre de Inveja

Nessa seção apresentamos outros resultados empíricos para o Problema da Precificação Livre de Inveja, complementando a Seção 3.4.

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
50	20	20	20	20	20
100	0	20	20	20	20
150	0	10	13	5	12
200	0	6	8	0	4
250	0	1	1	0	0
300	0	0	0	0	0

(a) *Modelo das Características*

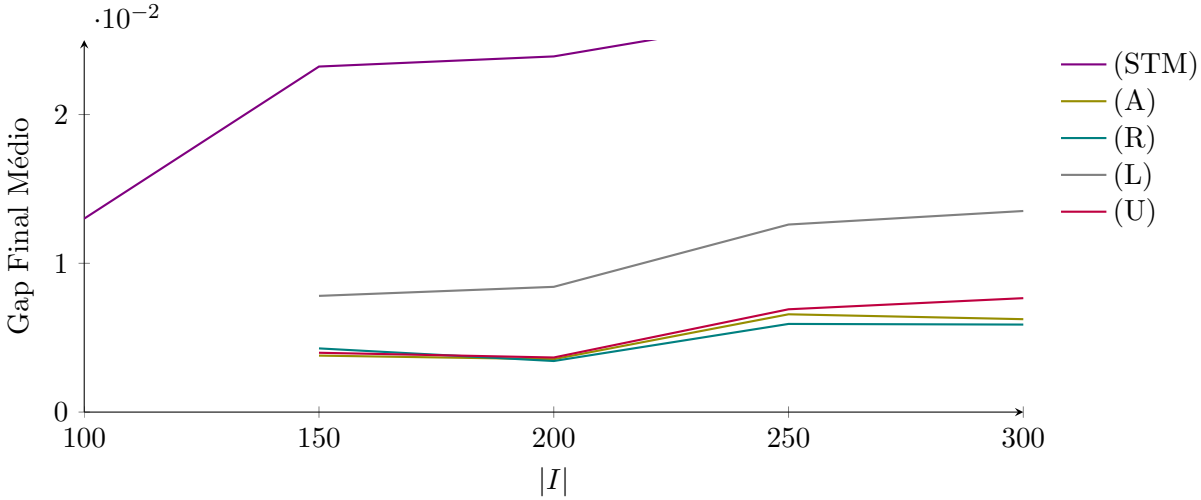
$\|I\|$	(STM)	(A)	(R)	(L)	(U)
50	20	20	20	20	20
100	11	20	20	19	20
150	0	15	17	4	15
200	0	4	7	0	4
250	0	1	2	0	0
300	0	0	0	0	0

(b) *Modelo da Vizinhança*

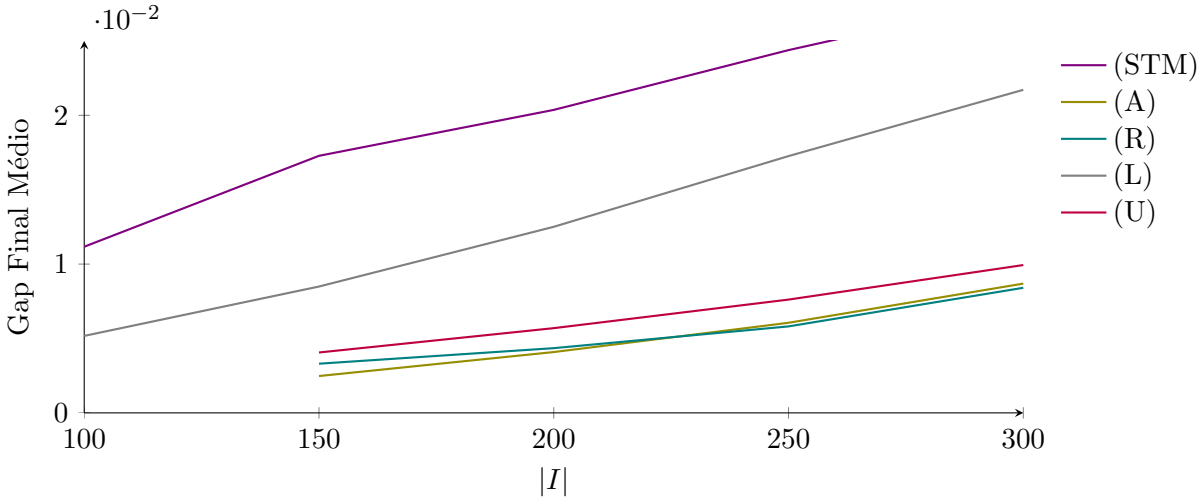
$\|I\|$	(STM)	(A)	(R)	(L)	(U)
50	20	20	20	20	20
100	3	20	20	19	20
150	0	14	14	7	14
200	0	5	7	0	4
250	0	0	0	0	0
300	0	0	0	0	0

(c) *Modelo da Popularidade*

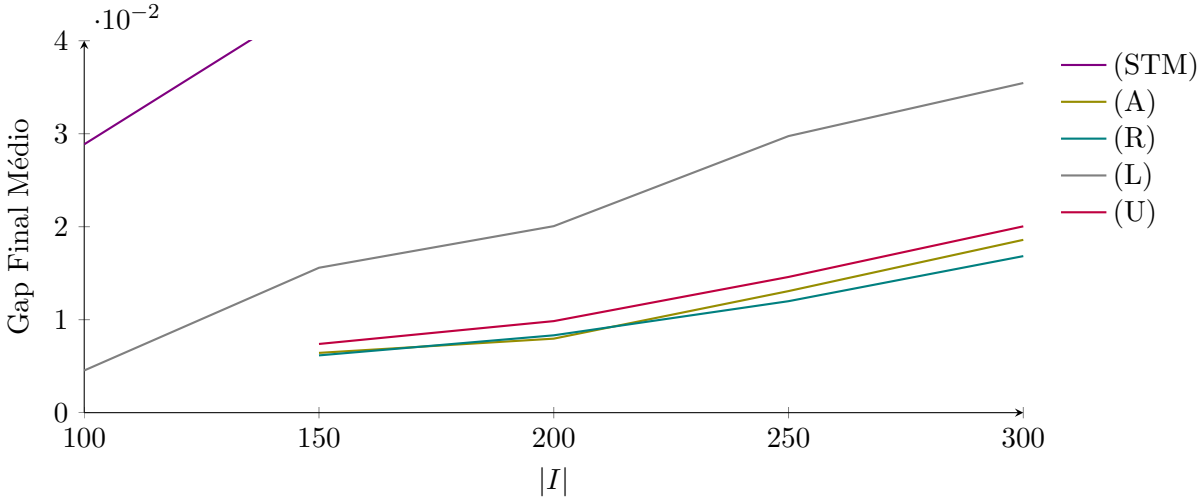
Tabela B.1: *Número de soluções ótimas encontradas para instâncias pequenas em cada um dos modelos.*



(a) *Modelo das Características*



(b) *Modelo da Vizinhança*



(c) *Modelo da Popularidade*

Figura B.1: *Gap final médio dos três modelos para instâncias pequenas que não foram resolvidas dentro do limite de tempo.*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
50	-	-	-	-	-
100	0,013 (0,00648)	-	-	-	-
150	0,0232 (0,00678)	0,0038 (0,00257)	0,00428 (0,00202)	0,00781 (0,00348)	0,00399 (0,00248)
200	0,0239 (0,00417)	0,00356 (0,00145)	0,00343 (0,000715)	0,00842 (0,00287)	0,00367 (0,00152)
250	0,0266 (0,00433)	0,00657 (0,00257)	0,00593 (0,00259)	0,0126 (0,00352)	0,00691 (0,00271)
300	0,026 (0,00439)	0,00624 (0,00243)	0,00588 (0,00245)	0,0135 (0,00344)	0,00766 (0,00257)

(a) *Modelo das Características*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
50	-	-	-	-	-
100	0,0112 (0,0062)	-	-	0,00517 (0)	-
150	0,0173 (0,00646)	0,00248 (0,00247)	0,00331 (0,00021)	0,00849 (0,00416)	0,00406 (0,00168)
200	0,0204 (0,00498)	0,00409 (0,0026)	0,00435 (0,0024)	0,0125 (0,00471)	0,00569 (0,00191)
250	0,0244 (0,00636)	0,00606 (0,00351)	0,00581 (0,00357)	0,0173 (0,00422)	0,00761 (0,00375)
300	0,0279 (0,00442)	0,00869 (0,00245)	0,00841 (0,00257)	0,0217 (0,00415)	0,00994 (0,00289)

(b) *Modelo da Vizinhança*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
50	-	-	-	-	-
100	0,0289 (0,0138)	-	-	0,00454 (0)	-
150	0,0447 (0,0151)	0,00643 (0,0024)	0,00616 (0,00355)	0,0156 (0,00801)	0,00739 (0,00362)
200	0,0501 (0,013)	0,00796 (0,00424)	0,00832 (0,00454)	0,0201 (0,00858)	0,00985 (0,00475)
250	0,0611 (0,0093)	0,0131 (0,00547)	0,012 (0,0054)	0,0297 (0,00687)	0,0146 (0,00485)
300	0,0646 (0,0132)	0,0186 (0,00749)	0,0168 (0,00708)	0,0355 (0,00864)	0,02 (0,00683)

(c) *Modelo da Popularidade*

Tabela B.2: *Gap final médio para instâncias pequenas (para os três modelos) que não foram resolvidas dentro do tempo limite. O valor em parênteses denota o desvio padrão.*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
500	0,036 (0,0044)	0,0114 (0,00194)	0,0107 (0,00185)	0,022 (0,00292)	0,0135 (0,00227)
1000	0,0394 (0,00478)	0,0124 (0,00184)	0,0114 (0,00127)	0,0238 (0,00162)	0,0148 (0,00121)
1500	0,0891 (0,0434)	0,0133 (0,00172)	0,0119 (0,00138)	0,0262 (0,00207)	0,0163 (0,00303)
2000	0,145 (0,0208)	0,0256 (0,0105)	0,0152 (0,00267)	0,0301 (0,00196)	0,0198 (0,00238)
2500	0,15 (0,0212)	0,0278 (0,00734)	0,0159 (0,00458)	0,0294 (0,00199)	0,0202 (0,00221)
3000	0,2 (0,0543)	0,0393 (0,0102)	0,0311 (0,00626)	0,0331 (0,00408)	0,0205 (0,00269)

(a) *Modelo das Características*

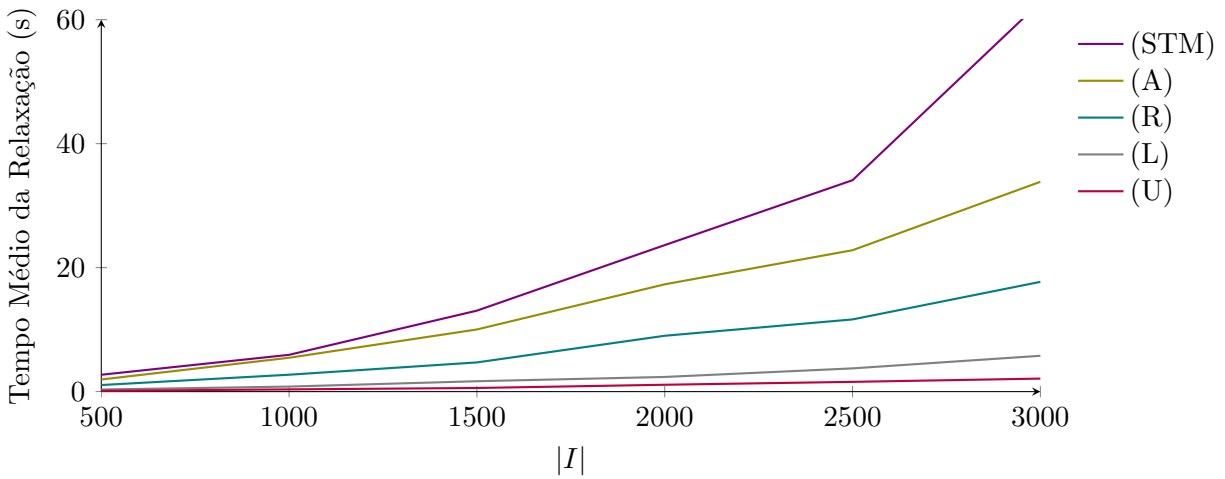
$\|I\|$	(STM)	(A)	(R)	(L)	(U)
500	0,031 (0,00614)	0,0124 (0,00351)	0,012 (0,00331)	0,0261 (0,00567)	0,0141 (0,00362)
1000	0,0363 (0,00388)	0,0156 (0,00195)	0,0152 (0,00189)	0,0325 (0,00305)	0,0179 (0,00212)
1500	0,0377 (0,00259)	0,017 (0,00253)	0,0163 (0,00177)	0,0344 (0,00253)	0,0191 (0,00185)
2000	∞	0,0375 (0,0842)	0,0177 (0,00267)	0,0519 (0,0383)	0,0203 (0,00185)
2500	∞	31,8 (138)	0,0183 (0,00229)	0,125 (0,195)	0,0289 (0,0306)
3000	∞	443 (671)	26,3 (114)	0,275 (0,218)	0,0306 (0,0114)

(b) *Modelo da Vizinhança*

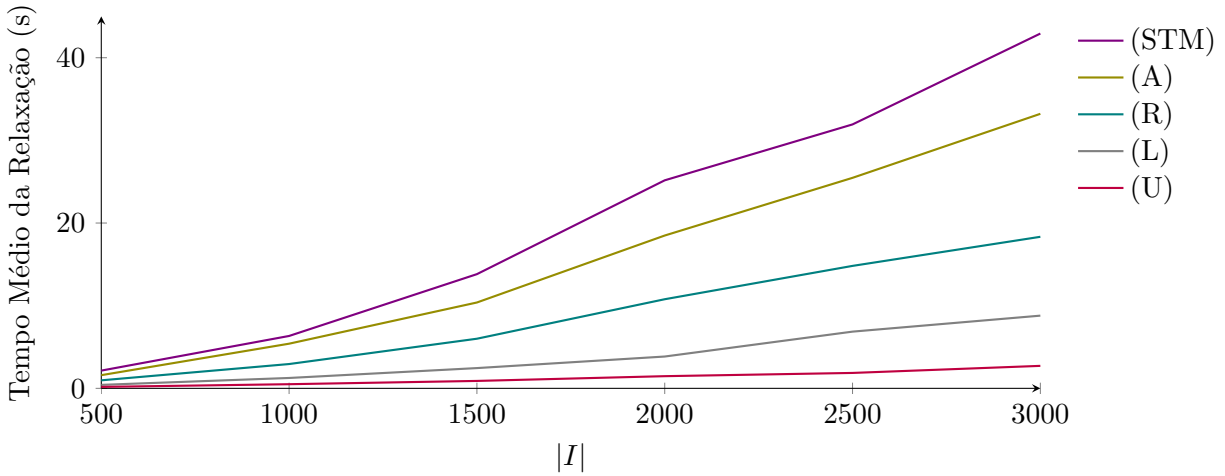
$\|I\|$	(STM)	(A)	(R)	(L)	(U)
500	0,106 (0,0177)	0,0426 (0,0101)	0,032 (0,00923)	0,0554 (0,0108)	0,033 (0,0091)
1000	0,198 (0,0648)	0,091 (0,0315)	0,0782 (0,0289)	0,092 (0,059)	0,0559 (0,0156)
1500	0,267 (0,0953)	0,977 (2,46)	0,802 (2,49)	0,217 (0,128)	0,0825 (0,0461)
2000	0,455 (0,16)	2,18 (3,97)	1,32 (3,05)	0,538 (0,783)	0,0815 (0,018)
2500	0,47 (0,202)	2,86 (4,48)	3,28 (4,87)	∞	0,151 (0,0706)
3000	0,662 (0,374)	3,59 (4,77)	2,98 (4,38)	∞	0,177 (0,0637)

(c) *Modelo da Popularidade*

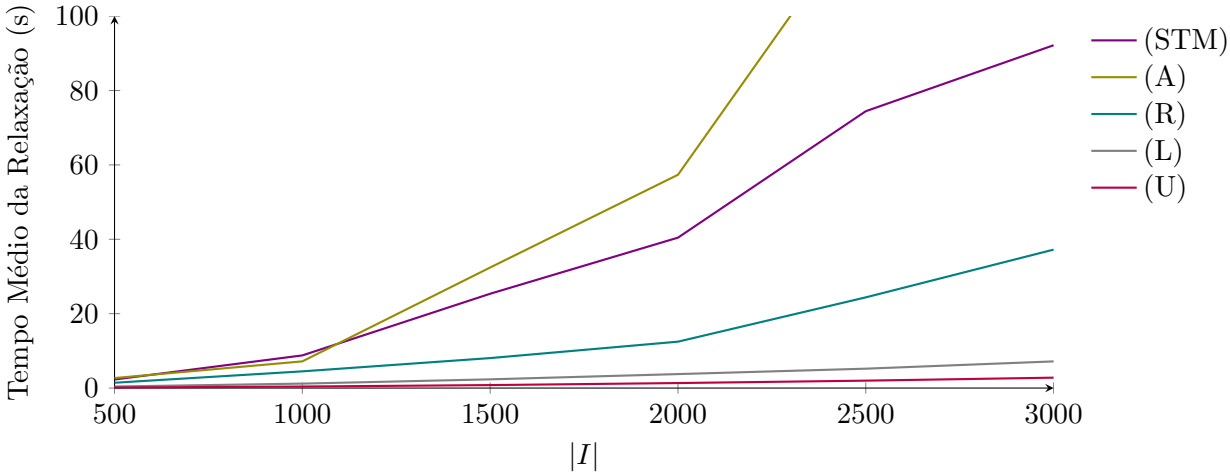
Tabela B.3: *Gap final médio dos três modelos para instâncias grandes. O valor em parênteses denota o desvio padrão.*



(a) *Modelo das Características*



(b) *Modelo da Vizinhança*



(c) *Modelo da Popularidade*

Figura B.2: *Tempo médio (em segundos) para resolver a relaxação linear do primeiro nó para instâncias grandes para os três modelos.*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
500	2,72 (0,749)	1,96 (0,485)	1,06 (0,264)	0,339 (0,071)	0,128 (0,024)
1000	5,94 (1,2)	5,46 (1,23)	2,73 (0,708)	0,814 (0,227)	0,361 (0,072)
1500	13,1 (2,7)	10 (2,1)	4,71 (1,11)	1,68 (0,422)	0,588 (0,148)
2000	23,6 (4,03)	17,3 (3,65)	9,01 (1,9)	2,37 (0,709)	1,11 (0,22)
2500	34,1 (10,7)	22,8 (3,04)	11,6 (2,29)	3,75 (0,865)	1,58 (0,351)
3000	62,8 (22,5)	33,8 (8,34)	17,7 (3,09)	5,78 (1,54)	2,1 (0,512)

(a) *Modelo das Características*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
500	2,15 (0,511)	1,59 (0,345)	0,969 (0,245)	0,42 (0,0757)	0,175 (0,0389)
1000	6,34 (1,72)	5,4 (1,38)	2,94 (0,737)	1,25 (0,241)	0,501 (0,0883)
1500	13,8 (3,16)	10,4 (2,32)	6 (1,04)	2,45 (0,697)	0,897 (0,18)
2000	25,2 (5)	18,5 (3,93)	10,8 (1,88)	3,85 (0,907)	1,47 (0,304)
2500	31,9 (6,79)	25,5 (4,42)	14,8 (2,73)	6,86 (1,03)	1,86 (0,531)
3000	42,9 (8,13)	33,2 (6,77)	18,3 (3,38)	8,8 (1,9)	2,72 (0,549)

(b) *Modelo da Vizinhança*

$\|I\|$	(STM)	(A)	(R)	(L)	(U)
500	2,3 (0,589)	2,69 (0,662)	1,43 (0,231)	0,412 (0,0788)	0,143 (0,0341)
1000	8,78 (3,1)	7,18 (2,18)	4,49 (0,667)	1,2 (0,242)	0,409 (0,0731)
1500	25,4 (15,3)	32,4 (43,2)	8,06 (2,43)	2,34 (0,615)	0,799 (0,17)
2000	40,4 (24,6)	57,3 (126)	12,5 (3,25)	3,76 (1,13)	1,35 (0,332)
2500	74,4 (51,2)	129 (262)	24,4 (10,8)	5,21 (1,22)	1,99 (0,48)
3000	92,2 (59)	250 (389)	37,2 (19,4)	7,17 (2,22)	2,8 (0,756)

(c) *Modelo da Popularidade*

Tabela B.4: *Tempo médio (em segundos) para resolver a relaxação linear do primeiro nó para instâncias grandes (para os três modelos). O valor em parênteses denota o desvio padrão.*

Referências Bibliográficas

[ABK+08] Yossi Azar, Benjamin Birnbaum, Anna Karlin, Claire Mathieu e C. Thach Nguyen. Improved approximation algorithms for budgeted allocations. Em *Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part I*, páginas 186–197, 2008. Citado na pág. 53, 56, 69

[ABKN09] Yossi Azar, Benjamin E. Birnbaum, Anna R. Karlin e C. Thach Nguyen. On revenue maximization in second-price ad auctions. Em *Proceedings of the 17th Annual European Symposium on Algorithms*, páginas 155–166, 2009. http://arxiv.org/abs/0908.2834. Citado na pág. 53, 54, 58, 60, 61, 62, 68, 69, 72, 73

[AFMZ04] Gagan Aggarwal, Tomás Feder, Rajeev Motwani e An Zhu. Algorithms for multi-product pricing. Em *Proceedings of the 31th International Colloquium on Automata, Languages and Programming*, páginas 72–83, 2004. Citado na pág. 18, 20, 21, 22, 31, 34, 37, 50, 75

[AGM06] Gagan Aggarwal, Ashish Goel e Rajeev Motwani. Truthful auctions for pricing search keywords. Em *Proceedings of the 7th ACM Conference on Electronic Commerce*, páginas 1–7, 2006. Citado na pág. 3

[AMT07] Zoe Abrams, Ofer Mendelevitch e John Tomlin. Optimal delivery of sponsored search advertisements subject to budget constraints. Em *Proceedings of the 8th ACM Conference on Electronic Commerce*, páginas 272–278, 2007. Citado na pág. 53

[BA99] Albert-László Barabási e Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999. Citado na pág. 46

[BEY98] Allan Borodin e Ran El-Yaniv. *Online computation and competitive analysis*. Cambridge University Press, 1998. Citado na pág. 12

[BJN07] Niv Buchbinder, Kamal Jain e Joseph Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. Em *Proceedings of the 15th Annual European Conference on Algorithms*, páginas 253–264, 2007. Citado na pág. 53

[BK07] Patrick Briest e Piotr Krysta. Buying cheap is expensive: hardness of non-parametric multi-product pricing. Em *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2007. Citado na pág. 18, 22

[Bri08] Patrick Briest. Uniform budgets and the envy-free pricing problem. Em *Proceedings of the 35th International Colloquium on Automata, Languages and Programming*, páginas 808–819, 2008. Citado na pág. 18, 37

[CC06] Miroslav Chlebík e Janka Chlebíková. Complexity of approximating bounded variants of optimization problems. *Theoretical Computer Science*, 354(3):320–338, 2006. Citado na pág. 60

[CCKK12] Parinya Chalermsook, Julia Chuzhoy, Sampath Kannan e Sanjeev Khanna. Improved hardness results for profit maximization pricing problems with unlimited supply. Em *Proccedings of the 15th International Workshop APPROX*, páginas 73–84, 2012. Citado na pág. 18, 37, 38, 50, 51

[CD10] Ning Chen e Xiaotie Deng. Envy-free pricing in multi-item markets. Em *Proceedings of the 37th International Colloquium Conference on Automata, Languages and Programming: Part II*, páginas 418–429, 2010. Citado na pág. 37

[CG08] Deeparnab Chakrabarty e Gagan Goel. On the approximability of budgeted allocations and improved lower bounds for submodular welfare maximization and gap. Em *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, páginas 687–696, 2008. Citado na pág. 53

[Chv83] Vašek Chvátal. *Linear Programming*. W. H. Freeman, 1983. Citado na pág. 14

[Cla71] Edward H. Clarke. Multipart pricing of public goods. *Public Choice*, 11:17–33, 1971. Citado na pág. 10

[CLN13] Parinya Chalermsook, Bundit Laekhanukit e Danupon Nanongkai. Independent set, induced matching, and pricing: connections and tight (subexponential time) approximation hardnesses. Em *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, 2013. Citado na pág. 18, 37

[CSS06] Peter Cramton, Yoav Shoham e Richard Steinberg, editors. *Combinatorial Auctions*. MIT Press, 2006. Citado na pág. 8, 10, 37

[DH09] Nikhil R. Devanur e Thomas P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. Em *Proceedings of the 10th ACM Conference on Electronic Commerce*, páginas 71–78, 2009. Citado na pág. 53

[EOS07] Benjamin Edelman, Michael Ostrovsky e Michael Schwarz. Internet advertising and the generalized second price auction: Selling billions of dollars worth of keywords. *American Economic Review*, 97(1):242–259, 2007. Citado na pág. 53

[ES73] Paul Erdös e JL Selfridge. On a combinatorial game. *Journal of Combinatorial Theory, Series A*, 14(3):298–301, 1973. Citado na pág. 18, 27

[ES10] Econsultancy e SEMPO. State of search engine marketing report. Relatório técnico, 2010. http://www.sempo.org/resource/resmgr/Docs/State-of-Search-Engine-Marke.pdf. Citado na pág. 53

[FFFS14] Cristina G. Fernandes, Carlos E. Ferreira, Álvaro J. P. Franco e Rafael C. S. Schouery. The unit-demand envy-free pricing problem. Em *3nd International Symposium on Combinatorial Optimization*, 2014. to appear. Citado na pág. 16

[FS12] Cristina G. Fernandes e Rafael C. S. Schouery. Second-price ad auctions with binary bids and markets with good competition. Em *Proceedings of the 2th International Conference on Combinatorial Optimization*, páginas 439–450, 2012. Citado na pág. 16

[FS14a] Cristina G. Fernandes e Rafael C. S. Schouery. Approximation algorithms for the max-buying problem with limited supply. Em *11th Latin American Symposium*, 2014. to appear. Citado na pág. 16

[FS14b] Cristina G. Fernandes e Rafael C. S. Schouery. Second-price ad auctions with binary bids and markets with good competition. *Theoretical Computer Science*, 2014. in press. Citado na pág. 16

[FT91] Drew Fudenberg e Jean Tirole. *Game Theory.* MIT Press, 1991. Citado na pág. 3

[GHK+05] Venkatesan Guruswami, Jason D. Hartline, Anna R. Karlin, David Kempe, Claire Kenyon e Frank McSherry. On profit-maximizing envy-free pricing. Em *Proceedings of the 16th Annual ACM-SIAM Symposium on Discrete Algorithms*, páginas 1164–1173, 2005. Citado na pág. 37

[Gil61] Edgar N. Gilbert. Random plane networks. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):533–543, 1961. Citado na pág. 45

[GJ79] Michael R. Garey e David S. Johnson. *Computers and intractibility.* W. H. Freeman, 1979. Citado na pág. 14, 23

[GK05] Venkatesan Guruswami e Subhash Khot. Hardness of Max 3SAT with no mixed clauses. Em *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, páginas 154–162, 2005. Citado na pág. 18, 21

[GKT51] David Gale, Harold W. Kuhn e Albert W. Tucker. Linear programming and the theory of games. Em *Activity Analysis of Production and Allocation*, páginas 317–329, 1951. Citado na pág. 13

[GLS88] Martin Grötschel, Laszlo Lovász e Alexander Schrijver. *Geometric algorithms and combinatorial optimization.* Springer-Verlag, 1988. Citado na pág. 13, 24

[GM08] Gagan Goel e Aranyak Mehta. Online budgeted matching in random input models with applications to adwords. Em *Proceedings of the 19th Annual ACM-SIAM Symposium on Discrete Algorithms*, páginas 982–991, 2008. Citado na pág. 53

[GMNS08] Ashish Goel, Mohammad Mahdian, Hamid Nazerzadeh e Amin Saberi. Advertisement allocation for generalized second pricing schemes. Em *Fourth Workshop on Ad Auctions*, 2008. Citado na pág. 53

[Gro73] Theodore Groves. Incentives in teams. *Econometrica*, 41:617–631, 1973. Citado na pág. 10

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001. Citado na pág. 21

[Hoc82] Dorit S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982. Citado na pág. 23

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984. Citado na pág. 12

[Kha79] Leonid G. Khachiyan. A polynomial algorithm in linear programming. *Doklady Akademii Nauk SSSR*, 244:1093–1096, 1979. Citado na pág. 12

[KK82] Narendra Karmarkar e Richard M. Karp. An efficient approximation scheme for the one-dimensional bin-packing problem. Em *Proceedings of the 23rd Annual Symposium on Foundations of Computer Science*, páginas 312–320, 1982. Citado na pág. 14

[Kri09] Vijay Krishna. *Auction Theory.* Academic Press, segunda edição, 2009. Citado na pág. 2, 7

[Kuh55] Harold W. Kuhn. The Hungarian Method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. Citado na pág. 16, 32

[LBPS00] Kevin Leyton-Brown, Mark Pearson e Yoav Shoham. Towards a universal test suite for combinatorial auction algorithms. Em *Proceedings of the 2nd ACM Conference on Electronic Commerce*, páginas 66–76, 2000. Citado na pág. 37

[LLN06] Benny Lehmann, Daniel Lehmann e Noam Nisan. Combinatorial auctions with decreasing marginal utilities. *Games and Economic Behavior*, 55(2):270–296, 2006. Citado na pág. 53

[LRS11] Lap Chi Lau, R. Ravi e Mohit Singh. *Iterative Methods in Combinatorial Optimization*. Cambridge Texts in Applied Mathematics. Cambridge University Press, 2011. Citado na pág. 14

[MST12] T. G. J. Myklebust, M. A. Sharpe e Levent Tunçel. Efficient heuristic algorithms for maximum utility product pricing problems. http://www.math.uwaterloo.ca/~ltuncel/publications/pricedown.pdf, 2012. Citado na pág. 38

[MSVV05] Aranyak Mehta, Amin Saberi, Umesh Vazirani e Vijay Vazirani. Adwords and generalized on-line matching. Em *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, páginas 264–273, 2005. Citado na pág. 53

[Mun57] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957. Citado na pág. 16, 19, 32, 57

[Nas50] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. Citado na pág. 7

[Nob13] Nobelprize.org. All prizes in economic sciences. http://www.nobelprize.org/nobel_prizes/economics/laureates/, Novembro 2013. Citado na pág. 3, 8

[NRTV07] Noam Nisan, Tim Roughgarden, Eva Tardos e Vijay V. Vazirani, editors. *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA, 2007. Citado na pág. 1, 3, 9, 53

[NW99] George L. Nemhauser e Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley, 1999. Citado na pág. 14

[OSW84] Shmuel S. Oren, Stephen A. Smith e Robert B. Wilson. Product line pricing. *Journal of Business*, 57(1):S73–S79, 1984. Citado na pág. 17

[OSW87] Shmuel S. Oren, Stephen A. Smith e Robert B. Wilson. Multi-product pricing for electric power. *Energy Economics*, 9(2):104–114, 1987. Citado na pág. 17

[Ras81] Stephen J. Rassenti. *0-1 Decision Problems with Multiple Resource Constraints: Algorithms and Applications*. Tese de Doutorado, University of Arizona, 1981. Citado na pág. 8

[Rou08] T. Roughgarden. An algorithmic game theory primer. Em *Proceedings of the 5th IFIP International Conference on Theoretical Computer Science*, 2008. Citado na pág. 1

[RRG06] Paat Rusmevichientong, Benjamin Van Roy e Peter W. Glynn. A nonparametric approach to multiproduct pricing. *Operations Research*, 54(1):82–98, 2006. Citado na pág. 17

[RSB82] Stephen J. Rassenti, Vernon L. Smith e Robert L. Bulfin. A combinatorial auction mechanism for airport time slot allocation. *Bell Journal of Economics*, 13:402–417, 1982. Citado na pág. 8

[Sch03] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*. Algorithms and combinatorics. Springer, 2003. Citado na pág. 16, 32, 56, 57

[Sen82] Subrata Sen. Issues in optimal product design. Em *Analytic Approaches to Product and Marketing Planning: The Second Conference*, páginas 265–274, 1982. Citado na pág. 17

[Smi86] Stephen A. Smith. New product pricing in quality sensitive markets. *Marketing Science*, 5(1):70–87, 1986. Citado na pág. 17

[Spe87] Joel Spencer. Ten lectures on the probabilistic method. *Society for Industrial and Applied Mathematics*, 1987. Citado na pág. 18, 27

[Sri08] Aravind Srinivasan. Budgeted allocations in the full-information setting. Em *Proceedings of the 11th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, páginas 247–253, 2008. Citado na pág. 53

[STM11] Romy Shioda, Levent Tunçel e T. G. J. Myklebust. Maximum utility product pricing models and algorithms based on reservation price. *Computational Optimization and Applications*, 48(2):157–198, Março 2011. Citado na pág. 37, 38, 39, 44, 77

[Var07] Hal R. Varian. Position auctions. *International Journal of Industrial Organization*, 25(6):1163–1178, 2007. Citado na pág. 53

[Vaz01] Vijay V. Vazirani. *Approximation algorithms*. Springer-Verlag, 2001. Citado na pág. 11

[Vic61] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961. Citado na pág. 10

[vN45] John von Neumann. First draft of a report on the EDVAC, 1945. Citado na pág. 1

[vNM44] John von Neumann e Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University press, primeira edição, 1944. Citado na pág. 1

[WS11] David P. Williamson e David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. Citado na pág. 10, 11

Índice Remissivo

$\mathcal{N}(\mu, \sigma^2)$, 44
α-aproximação, 10
α-competitivo, 12
$\wedge$, 20
$\neg$, 20
$\vee$, 20
$\lfloor y \rfloor_{1+\varepsilon}$, 50
2PM(δ), 56
2PM, *veja* Emparelhamento de Segundo Preço

À Prova de Estratégia, *veja* Compatível com Incentivo
(A), 40
(A$_\delta$), 77
Algoritmo
  de Aproximação, 10
  α-Especial, 57
  Especial, 57
  Online, 12
Alocação, 9, 15, 19
  Livre de Inveja, 38
  Viável, 34
APX, 11
APX-difícil, 11
Aresta
  Para Baixo, 61
  Para Cima, 61

B, 15
B2PAA, *veja* Leilão de Anúncio de Segundo Preço com Lances Binários
B2PAA(δ), 56
Bem-Estar Social, 7, 9

$c(.)$, 25
Casando Centavos, *veja* Jogo Casando Centavos
CATS, 37
Cláusula, 20
  Mista, 20
  Negativa, 20
  Positiva, 20
Compatível com Incentivo, 8, 10
Competidor
  do Topo, 63
  Inferior, 61
  Superior, 61
Comprador, 15
  de Segundo Preço, 55
  Livre de Inveja, 38
  Novo, 59
  Original, 59

DeterministicStarRounding, 28
Dilema do Prisioneiro, 3, 4
d_k, 31, 50
$\mathcal{D}_{1+\varepsilon}$, 50

Emparelhamento de Segundo Preço, 54, 56
Envy-Free Pricing Problem, *veja* Problema da Precificação Livre de Inveja
Equilibrio de Nash, 6
  Misto, 7
Escada de Preços, 17
Esperança, 28
Esperança-Comprador, 29
Esquema de Aproximação Polinomial, *veja* PTAS
Esquema de Aproximação Totalmente Polinomial, *veja* FPTAS
Estratégia, 4
  À Prova de, *veja* Compatível com Incentivo
  Dominante, 5
Estrela, 22

First-Price Ad Auctions, *veja* Leilão de Anúncio de Primeiro Preço
Forma Normal Conjuntiva, 20
Fortemente NP-difícil, 11
FPTAS, 11
(F2PM), 68
Fully Polynomial-Time Approximation Scheme, *veja* FPTAS
Função
  de Alocação, 9
  de Precificação, 9
  de Utilidade, *veja* Utilidade

G_O^M, 57
GSP, 10

I, 15
Interessado, 55
Item, 15
  Cópia de, 9, 59
  Original, 59
  Viável, 17, 19

Jogador, 4
  Racional, 4
Jogo, 4
  Casando Centavos, 6
  de Congestionamento, 4
  Extensão Mista de um, 6
Juntar, 57

$L(.)$, 34
(L), 42
(L_δ), 78
Lance, 8
Leilão
  Combinatório, 9
  de Anúncios, 9
    de Primeiro Preço, 53
    de Segundo Preço, 53, 55
    de Segundo Preço com Lances Binários, 55
  de Bens Digitais, 9
  de Demanda Unitária, 9
  de Oferta Ilimitada, 9
  de Oferta Limitada, 9
  de Segundo Preço, 8
  de Segundo Preço Generalizado, *veja* GSP
  de Um Único Item, 7
  de Vickrey, *veja* Leilão de Segundo Preço
  Inglês, 7
Literal, 20
  Negativo, 20
  Positivo, 20
Livre Disposição, 8
L, 29
Lucro
  do Comprador, 7
  do Leiloeiro, 9
$L(b)$, 25, 29

$M^{-1}(\cdot)$, 56
Marcar-e-Matar, 63
Marcar-e-Matar$_w$, 68
Max 3SAT, 20
Max E3SAT, 20
Max NM3SAT, 21
Max NM-E3SAT, 21
Max SAT, 20
Max-Buying Problem, *veja* Problema da Compra Máxima
Mecanismo VCG, *veja* VCG
Mercado δ-Competitivo, 54
Min-Buying Problem, *veja* Problema da Compra Mínima
MIP, 14
  Formulação, 14
Mixed Integer Program, *veja* MIP
MK$(\cdot)$, 68
MK$_w(\cdot)$, 68
Modelo
  da Popularidade, 45
  da Vizinhança, 45
  das Características, 44

O-Alg, 58
O-Emparelhamento de Segundo Preço, 55
O-EmparelhamentoMáximo, 57
O-Marcar-e-Matar, 65
O_b, 54
Oferta Limitada, 17
OPT$_{2P}(\cdot)$, 59
OPT$_{VC}(\cdot)$, 59
Orçamentos Uniformes, 17

p_i, 15
$P_{(i,S)}$, 23
Polynomial-Time Approximation Scheme, *veja* PTAS
Precificação, 15, 19
Preço da Anarquia, 10
Problema da Compra Máxima, 17
  com Orçamentos Uniformes, 20
  Limitada, 19
    EP, 19
    EP-(α, t), 31
    Online, 34
Problema da Compra Mínima, 17
Problema da Compra por Preferência, 17
Problema da Precificação Livre de Inveja, 38
Problema da Satisfatibilidade Máxima, 20
Problema da Separação, 13
Problema do Emparelhamento de Segundo Preço, 56
Problema Online, 11
Programa
  Dual, 13
  Inteiro Misto, *veja* MIP
  Linear, 12
    Ilimitado, 13
    Inviável, 13

Programação
Inteira Mista, 14
Linear, 12
Projeto de Mecanismos, 6
PTAS, 11

(R), 41
(R_δ), 78
Rank-Buying Problem, *veja* Problema da Compra por Preferência
RankingSimulate, 73
Razão
de Aproximação, 11
de Competitividade, 12
Relaxação Linear, 14
Resposta Ótima, 5
Resultado do Jogo, 4
ReverseMatching, 61
R_i, 39
$\mathrm{RL}_{(.)}(v)$, 41
$\mathrm{RM}(\cdot)$, 68

$\mathbb{S}$, 28
S_{-i}, 5
s_{-i}, 5
S_b, 39
Second-Price Ad Auction, *veja* Leilão de Anúncios de Segundo Preço
with Binary Bids, *veja* Leilão de Anúncios de Segundo Preço com Lances Binários
Second-Price Ad Auctions, *veja* Leilão de Anúncio de Segundo Preço
Second-Price Matching, *veja* Emparelhamento de Segundo Preço
Separar, 57
(SF), 23
$\mathrm{SOL}(p)$, 50
Solução
Ótima, 13
Viável, 12
StarRounding, 24
(STM), 40
(STM_δ), 77

Teorema Forte da Dualidade, 13
Teoria dos Jogos, 3
Algorítmica, 1

(U), 42
(U_δ), 78
Utilidade, 4, 38

V, 31
Valoração, 8, 15, 19
de um comprador, 15
de um item, 15
VCG, 10
v_{ib}, 15

$\mathbb{X}$, 28
$\overline{\mathbb{X}}$, 28
x_p, 50

Y_S, 29