

UNIVERSIDADE FEDERAL DO AMAZONAS  
INSTITUTO DE COMPUTAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

AUMENTO DE DESEMPENHO NA  
DETERMINAÇÃO DE PRECIFICAÇÕES  
ÓTIMAS LIVRES DE INVEJA

MARCOS MARREIRO SALVATIERRA

AUMENTO DE DESEMPENHO NA  
DETERMINAÇÃO DE PRECIFICAÇÕES  
ÓTIMAS LIVRES DE INVEJA

Tese apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas, Campus Universitário Senador Arthur Virgílio Filho, como requisito parcial para a obtenção do grau de Doutor em Informática.

ORIENTADOR: JUAN GABRIEL COLONNA

Manaus - AM

Maio de 2022

## Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

S182a Salvatierra, Marcos Marreiro  
Aumento de desempenho na determinação de precificações  
ótimas livres de inveja / Marcos Marreiro Salvatierra . 2022  
63 f.: il. color; 31 cm.

Orientador: Juan Gabriel Colonna  
Tese (Doutorado em Informática) - Universidade Federal do  
Amazonas.

1. Otimização combinatória. 2. Complexidade de algoritmos. 3.  
Ausência de inveja. 4. Programação dinâmica. I. Colonna, Juan  
Gabriel. II. Universidade Federal do Amazonas III. Título



PODER EXECUTIVO  
MINISTÉRIO DA EDUCAÇÃO  
INSTITUTO DE COMPUTAÇÃO



UFAM

PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# FOLHA DE APROVAÇÃO

"AUMENTO DE DESEMPENHO NA DETERMINAÇÃO  
DE PRECIFICAÇÕES ÓTIMAS LIVRES DE INVEJA"

MARCOS MARREIRO SALVATIERRA

Tese de Doutorado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Juan Gabriel Colonna - PRESIDENTE

Prof. Eduardo Luzeiro Feitosa - MEMBRO INTERNO

Profa. Fabíola Guerra Nakamura - MEMBRO INTERNO

Prof. Omar Latorre Vilca - MEMBRO EXTERNO

Profa. Renata da Encarnação Onety- MEMBRO EXTERNO

Manaus, 26 de Maio de 2022

# Resumo

---

Maximizar a receita de vendedores preservando a satisfação dos consumidores apresenta alguns desafios computacionais. O *problema da precificação livre de inveja* surgiu como uma alternativa de modelagem que é APX-difícil em geral, mas alguns casos e/ou variações já foram provados ser resolvíveis de maneira exata em tempo polinomial. Este trabalho aborda os casos de *emparelhamento perfeito livre de inveja*, no qual o número de consumidores coincide com o número de itens à venda e cada consumidor pode comprar apenas um único item, e o de *demanda unitária com substituíbilidade métrica*, no qual várias unidades de um mesmo item são vendidas em localidades diferentes e os consumidores destas localidades podem comprar apenas uma unidade do item, sendo que os custos de deslocamento de uma localidade à outra para a realização da compra formam um espaço métrico. Para o primeiro caso, foi projetado um algoritmo exato baseado em uma estratégia de programação dinâmica levando em consideração as utilidades dos consumidores para calcular as precificações ótimas. Para o segundo caso, foi proposta uma estratégia algorítmica que realiza a busca das soluções ótimas através de uma redução para o primeiro caso e da simplificação dos cálculos dos caminhos mínimos para a determinação dos preços. Comparando-se os métodos propostos com os existentes na literatura, no primeiro caso houve um aumento de desempenho de, em média, 49% na busca de preços ótimos, e no segundo caso houve uma redução da complexidade computacional de tempo na busca das soluções ótimas de  $O(n^4)$  para  $O(n^3)$ .

*Palavras-chave:* otimização combinatória, complexidade de algoritmos, ausência de inveja, programação dinâmica.

# Abstract

---

Maximizing sellers' revenue while preserving consumers' satisfaction presents some computational challenges. The envy-free pricing problem is itself a modeling alternative that is APX-hard in general, but some cases and/or variations have already been proven to be solvable exactly in polynomial time, and for others, there exists approximation algorithms with constant ratio in polynomial time. This work addresses the cases of *envy-free perfect matching*, in which the number of consumers coincides with the number of items on sale and each consumer can buy only a single item, and *unit demand with substitutability metric*, in which several units of the same item are sold in different locations and consumers in these locations can buy only one unit of the item, and the costs of moving from one location to another to make the purchase form a metric space. For the first case, an exact algorithm was designed based on a dynamic programming strategy taking into account the consumers' utilities to calculate the optimal pricing. For the second case, an algorithmic strategy was proposed that performs the search for the optimal solutions through a reduction to the first case and the simplification of the calculations of the shortest paths for the determination of prices. Comparing the proposed methods with those in the literature, in the first case there was a performance increase of, on average, 49 % in the search for optimal prices, and in the second case there was a reduction in computational time complexity in the search for optimal solutions from  $O(n^4)$  to  $O(n^3)$ .

*Keywords:* combinatorial optimization, algorithm complexity, envy-freeness, dynamic programming.

# Lista de Figuras

---

2.1	Exemplo de leilão combinatório com demanda unitária. No EFPP com demanda unitária, o vendedor precisa definir uma alocação e precificação de forma que tanto a sua receita quanto a utilidade dos compradores sejam maximizadas. . . . .	15
4.1	A árvore associada à matriz $\mathcal{U}$ , construída pelo procedimento. . . . .	32
4.2	Um <i>boxplot</i> e seus principais elementos . . . . .	40
4.3	Grafo usado por Arbib, Karaşan e Pınar [Arbib et al., 2019] para encontrar os preços ótimos livres de inveja. . . . .	42
4.4	<i>Boxplots</i> dos tempos de execução para $n = 1000$ e $n = 2000$ . . . . .	43
4.5	<i>Boxplots</i> dos tempos de execução para $n = 5000$ e $n = 10000$ . . . . .	43
4.6	<i>Boxplot</i> dos tempos de execução para $n = 15000$ . . . . .	43
5.1	Uma instância do UDEFP com substituíbilidade métrica. . . . .	46
5.2	Uma instância do EFPM. . . . .	47
5.3	A rede construída para encontrar os preços livres de inveja. . . . .	48
5.4	Uma instância do UDEFP com substituíbilidade métrica e custos de deslocamento não simétricos. . . . .	52

# Lista de Tabelas

---

3.1	Formulações MIP para o EFPP e problemas correlatos . . . . .	19
3.2	Casos especiais ótimos do EFPP . . . . .	22
3.3	Heurísticas e meta-heurísticas para casos e variações do EFPP . . . . .	23
3.4	Algoritmos aproximativos e inaproximabilidade para casos do EFPP . . . . .	26
4.1	Comparação dos tempos de execução dos algoritmos de Bellman-Ford e o Algoritmo 2 . . . . .	42
4.2	Testes dos postos sinalizados de Wilcoxon com nível de significância $\alpha = 0.05$ . . . . .	44
6.1	Comparação dos tempos de execução dos algoritmos para o EFPM . . . . .	63



# Sumário

---

<b>Resumo</b>	<b>vii</b>
<b>Abstract</b>	<b>ix</b>
<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Justificativa . . . . .	2
1.2 Questão da pesquisa . . . . .	4
1.3 Objetivos . . . . .	5
1.4 Organização do trabalho . . . . .	5
<b>2 Fundamentação teórica</b>	<b>7</b>
2.1 Programação inteira mista . . . . .	7
2.2 Teoria dos algoritmos . . . . .	8
2.2.1 Técnicas de projeto de algoritmos . . . . .	8
2.2.2 Técnicas de análise de algoritmos . . . . .	9
2.3 Teoria dos grafos . . . . .	10
2.3.1 Problema do caminho mínimo . . . . .	10
2.3.2 Problema do emparelhamento de peso máximo . . . . .	11
2.3.3 Problema do conjunto independente de peso máximo . . . . .	11
2.4 Teoria dos jogos algorítmica . . . . .	12
2.4.1 Problemas de precificação . . . . .	12
2.4.2 Problema da precificação livre de inveja . . . . .	13
2.5 Considerações finais . . . . .	16

<b>3</b>	<b>Revisão da Literatura</b>	<b>17</b>
3.1	Formulações MIP . . . . .	17
3.2	Casos polinomiais . . . . .	18
3.3	Heurísticas e meta-heurísticas . . . . .	21
3.4	Algoritmos aproximativos/inaproximabilidade . . . . .	23
3.5	Considerações finais . . . . .	25
<b>4</b>	<b>O caso de emparelhamento perfeito livre de inveja</b>	<b>27</b>
4.1	Preliminares . . . . .	27
4.2	Sequências de utilidades . . . . .	31
4.3	Algoritmo, corretude e complexidade de tempo . . . . .	35
4.4	Comparação com a estratégia de caminhos mínimos . . . . .	39
4.4.1	Materiais e métodos . . . . .	39
4.4.2	Resultados dos experimentos . . . . .	41
4.5	Considerações finais . . . . .	44
<b>5</b>	<b>O caso de demanda unitária com substituíbilidade métrica</b>	<b>45</b>
5.1	Equilíbrio competitivo . . . . .	46
5.1.1	Redução do UDEFPM com substituíbilidade métrica ao EFPM	46
5.1.2	Alocação . . . . .	47
5.1.3	Precificação . . . . .	48
5.1.4	Algoritmo, corretude e complexidade de tempo . . . . .	49
5.2	Cenário geral . . . . .	50
5.3	Comparação com a estratégia de conjuntos independentes de peso máximo . . . . .	51
5.4	Considerações finais . . . . .	52
<b>6</b>	<b>Conclusões</b>	<b>53</b>
	<b>Referências Bibliográficas</b>	<b>57</b>

# Introdução

---

Propor algoritmos cada vez mais eficientes para problemas computacionais está na pauta não só da pesquisa em otimização combinatória, mas da ciência da computação em geral. Pode-se citar o caso clássico da multiplicação de matrizes, para o qual Coppersmith e Winograd [Coppersmith & Winograd, 1987] apresentaram um algoritmo de tempo polinomial  $O(n^{2.375})$ , aumentando o desempenho na busca de soluções exatas em relação ao algoritmo de Strassen [Strassen, 1969], que é de tempo  $O(n^{2.807})$ . Outro problema computacional clássico que continua atraindo a atenção de pesquisadores na busca de soluções cada vez mais eficientes é o problema da ordenação, para o qual Alotaibi, Almutairi e Kurdi [Alotaibi et al., 2020] propuseram um algoritmo de tempo quadrático que apresenta melhor desempenho que os algoritmos *bubble sort*, *selection sort* e *insertion sort*.

No que diz respeito à otimização combinatória, especificamente, um problema clássico que continua desafiando os pesquisadores na busca de soluções ótimas ou aproximadas de maneira cada vez mais eficiente é o problema do caixeiro viajante, para o qual se desenvolveram desde um algoritmo exato de tempo  $O(n^{2^{2^n}})$  – que é o algoritmo de Held-Karp (Held & Karp [1962]) – até um algoritmo de tempo polinomial que aproxima a solução ótima a um fator  $3/2$ , como o proposto por Karlin, Klein e Ghazan (Karlin et al. [2021]).

Outro problema que tem sido explorado é o *problema da precificação livre de inveja* (EFPP – *envy-free pricing problem*, em inglês). De forma coloquial, neste problema – que é o foco do presente estudo – um vendedor possui uma certa quantidade  $n$  de itens à venda, cada um com uma ou várias cópias disponíveis, e

$m$  consumidores atribuem valores a cada item ou a pacotes específicos de itens. O vendedor precisa decidir quais itens ou pacotes de itens (juntamente com as cópias de cada item definidas pelos consumidores) devem ser vendidos a determinados consumidores, de forma que os preços estabelecidos os deixem satisfeitos no sentido de que um consumidor não tenha "inveja" de um item ou pacote de itens que não tenha sido alocado a ele – ou seja, a diferença entre o que o consumidor contemplado está disposto a pagar pelos itens ou pacotes de itens alocados a ele e os preços estabelecidos a estes é maior ou igual aos dos outros itens ou pacotes de itens, sendo esta diferença não negativa – ao mesmo tempo em que a receita com a venda seja maximizada. Além do mais, os preços estabelecidos são os mesmos para todos os consumidores, isto é, consumidores que comprem um determinado item pagam o mesmo valor por ele.

Várias propostas de aplicações práticas do EFPP podem ser encontradas na literatura: venda de energia para abastecimento de veículos elétricos e venda de espaços de publicidade em páginas da web (Anshelevich et al. [2017]); aquisição de itens e materiais por parte de órgãos públicos, e venda de *slots* de memória em servidores remotos (Bilò et al. [2017]); venda de combustível em diferentes localidades, e venda online de produtos diversos para diferentes países (Chen et al. [2011]), apenas para citarmos algumas. A definição formal do EFPP encontra-se na Seção 2.4.2.

Dado que soluções ótimas para casos particulares do EFPP podem ser encontradas em tempo polinomial, como os estudados por Günlük [Günlük, 2008], Chen e Deng [Chen & Deng, 2014], e Arbib, Karaşan e Pınar [Arbib et al., 2019], por exemplo, um desafio computacional que se apresenta é propor métodos cada vez mais eficientes na busca de soluções ótimas para casos e variações do EFPP.

## 1.1 Justificativa

O conceito de *ausência de inveja*<sup>1</sup> surgiu em problemas de divisão justa de fatias de bolo [Gamow, 1958, Steinhaus, 1949], e mais tarde foi formalizado por Foley [Foley, 1967] no contexto da divisão justa de recursos na administração pública.

---

<sup>1</sup>Tradução livre para *envy-freeness*.

A partir daí, várias técnicas e métodos foram desenvolvidos para a resolução e aplicação de problemas envolvendo este conceito em diversos contextos [Maskin, 1987, Saari, 1988, Varian, 1974].

Mais recentemente, Guruswami et. al [Guruswami et al., 2005] formularam o EFPP no contexto de leilões combinatórios – nos quais o leiloeiro disponibiliza vários itens indivisíveis à venda e os participantes podem dar lances tanto a itens singularmente quanto a pacotes de itens – provando que, em geral, o EFPP é APX-difícil, ou seja, que não é possível projetar algoritmos que produzem soluções aproximadas com razão constante em relação às soluções ótimas em tempo polinomial. Eles apresentaram um algoritmo que encontra soluções aproximadas com razão  $O(\log m)$  para o caso de demanda unitária (no qual pode-se comprar apenas um único item), e outro algoritmo que encontra soluções aproximadas com razão  $O(\log m + \log n)$  para o caso de demanda por pacotes específicos<sup>2</sup> (no qual pode-se comprar apenas conjuntos específicos de itens previamente determinados pelos consumidores), em que  $m$  é o número de consumidores participantes e  $n$  é o número de itens disponibilizados à venda.

A partir daí, vários estudos foram realizados no sentido de se buscar soluções tanto ótimas quanto aproximadas cada vez mais eficientes para casos e variações do EFPP. Pode-se citar, por exemplo, o trabalho de Demaine et al. [Demaine et al., 2008], que propuseram um algoritmo aproximativo com razão  $O(\log n)$  para um caso de demanda por pacotes específicos. Já Anshelevich, Kar e Sekar [Anshelevich et al., 2017], ao considerarem um cenário de demanda unitária em grandes mercados, conseguiram projetar um algoritmo que produz soluções aproximadas com razão constante, a saber, 1.88.

No que diz respeito ao desenvolvimento de algoritmos que encontram soluções ótimas em tempo polinomial para casos do EFPP, pode-se citar os trabalhos de Chen, Ghosh e Vassilvitskii [Chen et al., 2011], e Arbib, Karaşan e Pinar [Arbib et al., 2019], que provaram que o EFPP com demanda unitária pode ser resolvido de maneira exata em tempo  $O(n^4)$  e  $O(n^3)$ , respectivamente, em casos específicos: o de substituíbilidade métrica e o de emparelhamento perfeito livre de inveja, respectivamente.

Mesmo em casos NP-difíceis do EFPP (ou seja, para os quais não é possível

---

<sup>2</sup>Tradução livre para *single-minded demand*.

encontrar sempre soluções ótimas em tempo polinomial), há o interesse de se desenvolver estratégias algorítmicas que produzam soluções, ótimas ou não, num tempo cada vez mais rápido. Isto se reflete, por exemplo, nos trabalhos de Shioda, Tunçel e Myklebust [Shioda et al., 2011] e Myklebust, Sharpe e Tunçel [Myklebust et al., 2016], que desenvolveram heurísticas baseadas em formulações em termos de programação inteira mista (MIP – *mixed-integer programming*, em inglês), na busca de soluções para casos de demanda unitária do EFPP. Formulações MIP são tipos de modelagem matemática consistindo de funções objetivo, a serem maximizadas ou minimizadas, seguidas de restrições envolvendo igualdades e/ou desigualdades que descrevem as características e propriedades do problema, com variáveis inteiras e contínuas. Elas são úteis no desenvolvimento de estratégias algorítmicas, pois definem o espaço de busca através das restrições, bem como possuem indicadores da qualidade das soluções a serem produzidas, orientados pelas funções objetivo. Uma descrição formal de modelos MIP encontra-se na Seção 2.1.

Percebe-se, então, que há um esforço na pesquisa em otimização combinatória e, mais especificamente, sobre o EFPP, em se projetar algoritmos cada vez mais eficientes seja no sentido da diminuição da razão de aproximação, seja no aumento de desempenho (em relação ao tempo computacional) na busca de soluções ótimas, sendo este último aspecto o abordado no presente trabalho.

Portanto, esta tese dá a sua contribuição propondo algoritmos eficientes para dois casos particulares do EFPP: o caso de emparelhamento perfeito livre de inveja e o de demanda unitária com substituíbilidade métrica. No primeiro caso, resultados teórico-empíricos mostram que a estratégia utilizada aqui possui a mesma complexidade de tempo  $O(n^3)$  que a encontrada na literatura, porém com aumento de eficiência de aproximadamente 49% no tempo da busca de soluções ótimas quando os algoritmos são avaliados empiricamente. No segundo caso, resultados teóricos mostram que os algoritmos propostos neste trabalho reduzem a complexidade de tempo na busca de soluções ótimas de  $O(n^4)$  para  $O(n^3)$ .

## 1.2 Questão da pesquisa

Diante do exposto, estabelece-se a questão da pesquisa: **para quais casos do problema da precificação livre de inveja com demanda unitária pode-**

### **se desenvolver algoritmos exatos mais eficientes que os constantes na literatura?**

A investigação teve como base as seguintes hipóteses:

- (i) casos de demanda unitária possuem características combinatórias menos complexas que outros casos de demanda;
- (ii) algoritmos combinatórios de propósitos específicos podem ser mais eficientes que algoritmos de propósitos gerais para os problemas em questão, ou seja, projetar algoritmos que levem as particularidades de cada caso pode resultar em melhores desempenhos, ao invés de se adaptar algoritmos desenvolvidos para casos mais gerais e/ou problemas correlatos; e
- (iii) dividir o EFPP em dois subproblemas – alocação e precificação – pode produzir soluções ótimas de maneira mais eficiente do que atacar o problema como um todo.

## **1.3 Objetivos**

O objetivo geral da pesquisa é desenvolver algoritmos exatos eficientes para casos especiais do EFPP com demanda unitária utilizando propriedades intrínsecas dos problemas abordados que auxiliem na simplificação dos cálculos, proporcionando, dessa forma, mais eficiência na busca de soluções ótimas para problemas de otimização combinatória no contexto de leilões combinatórios.

Partindo desse objetivo geral, define-se os objetivos específicos:

- identificar casos polinomiais do EFPP com demanda unitária;
- demonstrar matematicamente as propriedades dos casos polinomiais selecionados que possam ser úteis na formulação de algoritmos eficientes.

## **1.4 Organização do trabalho**

A partir deste capítulo introdutório, o presente trabalho organiza-se da seguinte forma: no Capítulo 2 são apresentados os fundamentos teóricos que sustentam

as formulações e os métodos empregados na pesquisa. No Capítulo 3 faz-se um resumo dos resultados que atualizam o estado da arte, englobando formulações MIP e estratégias algorítmicas tanto para casos polinomiais quanto para casos NP-difíceis, assim como resultados sobre inaproximabilidade. A apresentação dos resultados obtidos durante a pesquisa será feita nos Capítulos 4 e 5, juntamente com a discussão destes e a comparação com as estratégias da literatura. Por fim, o Capítulo 6 tece as considerações finais com um resumo dos resultados obtidos, assim como aponta perspectivas para trabalhos futuros.



# Fundamentação teórica

---

Este capítulo apresenta o ferramental teórico necessário ao entendimento do problema computacional estudado na pesquisa: programação inteira mista, teoria dos algoritmos, teoria dos grafos e teoria dos jogos algorítmica.

## 2.1 Programação inteira mista

Modelos MIP possuem a seguinte estrutura geral:

$$\begin{aligned} \min_{x,y} \quad & f(x, y) \\ \text{s.a.} \quad & h(x, y) = 0 \\ & g(x, y) \leq 0 \\ & x \in \mathbb{R}^n \\ & y \in \mathbb{Z}^m \end{aligned}$$

Este tipo de formulação abrange uma gama de problemas de otimização, caracterizados de acordo com considerações apropriadas ou eliminação dos seus elementos. Se as variáveis inteiras são eliminadas, e tanto a função objetivo como as restrições são lineares, a formulação é de um problema de *programação linear*. Se as variáveis inteiras são eliminadas e existem termos não lineares na função objetivo e/ou nas restrições, a formulação é de um problema de *programação não linear*. Se as variáveis inteiras possuem relação linear e separável com as variáveis contínuas, e tanto a função objetivo como as restrições são lineares, a formulação

é de um problema de *programação inteira mista linear*. Se existem termos não lineares na função objetivo e/ou nas restrições, a formulação é de um problema de *programação inteira mista não linear*.

Em muitas aplicações de modelos MIP as variáveis inteiras são binárias, dando surgimento a dificuldades devido à natureza combinatória dessas variáveis.

Análises teóricas e práticas de problemas lineares e não lineares podem ser encontradas nos livros de Floudas [Floudas, 1995], Parker e Radin [Parker & Rardin, 2014], Schrijver [Schrijver, 1998], e Wolsey e Nemhauser [Wolsey & Nemhauser, 1999].

## 2.2 Teoria dos algoritmos

A teoria dos algoritmos ocupa-se do estudo de técnicas de projeto e análise de algoritmos para se resolver problemas computacionais. O presente trabalho, bem como os encontrados na literatura sobre o EFPP, utilizaram algumas destas técnicas, que serão descritas a seguir.

### 2.2.1 Técnicas de projeto de algoritmos

Dentre as principais estratégias de projeto de algoritmos desenvolvidos para a busca de soluções do EFPP, pode-se destacar:

- **estratégia gulosa:** esta técnica realiza a busca de soluções ótimas de um determinado problema de otimização tomando a melhor solução a cada passo, ou seja, se o problema é de maximização, o algoritmo constrói a solução a partir de um valor inicial digamos, 0, e percorre o espaço de soluções a cada passo tomando como solução corrente aquela que mais aumenta o valor em relação à solução anterior, sem nunca voltar atrás. Um exemplo de algoritmo guloso é o algoritmo de Prim [Prim, 1957], que resolve o problema da árvore geradora mínima.
- **programação dinâmica:** esta estratégia, assim como a de divisão e conquista, divide a instância original de um problema em instâncias menores, com a diferença de que as soluções do problema para as várias subinstâncias

são armazenadas. Ao final, a solução do problema original é obtida a partir dos valores armazenados. Convém notar que um problema só pode ser resolvido através de programação dinâmica se toda solução para uma instância qualquer do problema contiver as soluções das subinstâncias. Esta propriedade é conhecida como *subestrutura ótima*. Um exemplo de algoritmo que utiliza a estratégia de programação dinâmica é o algoritmo de Floyd-Warshall [Floyd, 1962], que resolve o problema dos caminhos mínimos entre todos os pares de vértices de um grafo direcionado.

### 2.2.2 Técnicas de análise de algoritmos

No que diz respeito à análise de algoritmos, são observados os tempos de execução (complexidade de tempo) e as quantidades de memória necessárias à execução dos algoritmos (complexidade de espaço), sendo utilizadas unidades abstratas de medida para as quantificações, isto é, as medidas não dependem das especificidades de uma máquina. Na análise de ambas as complexidades, os cálculos são realizados através de funções dependentes, em geral, dos tamanhos das instâncias. Para se estipular e representar os limites superiores assintóticos das complexidades, ou seja, os piores casos, utiliza-se a notação *Big-O*.

Dadas duas funções  $f, g : \mathbb{N} \rightarrow \mathbb{R}$ , diz-se que  $f(n)$  pertence à classe  $O(g(n))$  e escreve-se, por abuso de notação,  $f(n) = O(g(n))$ , se existem um número real  $c > 0$  e um número natural  $n_0$  de forma que  $f(n) \leq cg(n)$  para todo  $n \geq n_0$ , ou seja, se a função  $g$  limita superior e assintoticamente a função  $f$ . Dessa forma, se  $f$  é uma função que descreve o número de operações realizadas por um algoritmo projetado para resolver um determinado problema de acordo com o tamanho das instâncias, diz-se que o algoritmo é executado, ou resolve o problema, em tempo  $O(g(n))$ .

Para uma consulta mais detalhada sobre a teoria dos algoritmos, recomenda-se os livros de Kleinberg e Tardos [Kleinberg & Tardos, 2006], Dasgupta, Papadimitriou e Vazirani [Dasgupta et al., 2008] e Cormen et al. [Cormen et al., 2009], entre outros.

## 2.3 Teoria dos grafos

Um *grafo*  $G = G(V, E)$  consiste num par de conjuntos  $E$  e  $V$  de forma que  $V \neq \emptyset$  e  $E$  é formado por pares de elementos de  $V$ . Os elementos de  $V$  são chamados de *vértices* e os elementos de  $E$ , *arestas*. Se os elementos de  $E$  são pares ordenados, diz-se que  $G$  é *direcionado*<sup>1</sup>. Caso contrário,  $G$  é dito ser *não direcionado*. Se existem  $V_1, V_2 \subsetneq V$  tais que  $V = V_1 \cup V_2$ ,  $V_1 \cap V_2 = \emptyset$  e  $u_i, v_i \in V_i \implies \nexists (u_i, v_i) \in E$  para cada  $i = 1, 2$ , diz-se que  $G$  é *bipartido*. Uma sequência de vértices  $v_1, \dots, v_k$  tais que  $(v_{j-1}, v_j) \in E$  para todo  $2 \leq j \leq k$  é chamado de *caminho* em  $G$ . Se  $k \geq 3$  e  $v_1 = v_k$ , o caminho é chamado de *ciclo*.

Na modelagem de certos problemas, atribui-se valores numéricos às arestas, o que se chama de *pesos*. Dessa forma, o *comprimento* do caminho  $v_1, \dots, v_k$  é igual à soma dos pesos das arestas desse caminho. No caso de não haver pesos nas arestas, o comprimento é  $k - 1$ . A *distância* entre dois vértices  $u$  e  $v$ , denotada por  $d(u, v)$ , é o menor comprimento dos caminhos  $v_1, \dots, v_k$  tais que  $v_1 = u$  e  $v_k = v$ . Um *emparelhamento* em  $G$  é um subconjunto de arestas  $M \subseteq E$  tal que para todo par de arestas distintas  $(u_1, u_2), (v_1, v_2) \in M$  tem-se  $u_i \neq v_j, \forall i, j = 1, 2$ . Um emparelhamento  $M$  é dito ser *perfeito* se para cada vértice  $v \in V$ , existe alguma aresta  $(x, y) \in M$  tal que  $v = x$  ou  $v = y$ , e o *peso de um emparelhamento*  $M$  é a soma dos pesos das arestas de  $M$ , denotado por  $w(M)$ .

Um *conjunto independente* de um grafo  $G = G(V, E)$  é um conjunto de vértices  $V' \subseteq V$  tal que  $u, v \in V' \implies \nexists (u, v) \in E$  para quaisquer  $u, v \in V'$ , ou seja, não existem arestas que conectam dois vértices quaisquer de  $V'$ .

### 2.3.1 Problema do caminho mínimo

Um problema estudado em termos de modelagem em grafos é o *problema do caminho mínimo*: dados um grafo  $G(V, E)$  e dois vértices  $u, v \in V$ , encontrar um caminho que determine  $d(u, v)$ . Este problema pode ser resolvido em tempo polinomial, e os algoritmos clássicos para resolvê-lo são:

- **algoritmo de Dijkstra** [Dijkstra et al., 1959]: utiliza uma estratégia gulosa

<sup>1</sup>Grafos direcionados também costumam ser chamados de *redes*, seus vértices de *nós* e suas arestas de *arcos*.

para encontrar a distância de um vértice dado a todos os outros vértices do grafo. Este algoritmo foi projetado para calcular distâncias em grafos que não possuem arestas com peso negativo. Sua complexidade de tempo é  $O(|E| + |V| \log |V|)$ , em que  $|V|$  e  $|E|$  é o número de vértices e o número de arestas do grafo, respectivamente;

- **algoritmo de Bellman-Ford** [Bellman, 1958]: utiliza uma estratégia em programação dinâmica. Calcula a distância de um vértice dado a todos os outros vértices do grafo em tempo  $O(|V||E|)$ . Este algoritmo foi projetado para calcular distâncias em grafos que não possuem ciclos com comprimento negativo, ou seja, em grafos que não possuem caminhos que iniciam e terminam num mesmo vértice e cujas somas dos pesos das arestas desses caminhos não seja negativa.

### 2.3.2 Problema do emparelhamento de peso máximo

Outro problema estudado em teoria dos grafos é o *problema do emparelhamento de peso máximo*: dado um grafo  $G(V, E)$ , encontrar um emparelhamento  $M \subseteq E$  de forma que  $M = \arg \max_{M' \subseteq E} \{w(M')\}$ . Dentre os algoritmos clássicos para resolver este problema em tempo polinomial  $O(|V||E|^2)$  encontram-se o **método húngaro** [Kuhn, 1955] e o **algoritmo de Edmonds** [Edmonds, 1965], que utilizam estratégias primais-duais e são aplicados em grafos bipartidos e generalizados, respectivamente.

### 2.3.3 Problema do conjunto independente de peso máximo

No *problema do conjunto independente de peso máximo* é dado um grafo  $G = G(V, E)$  não direcionado com pesos nos vértices, e procura-se um conjunto independente  $V' \subseteq V$  que maximiza  $\sum_{v_i \in V'} w(v_i)$ . Vários algoritmos que encontram soluções ótimas em tempo polinomial para casos particulares deste problema têm sido desenvolvidos, como os apresentados por Chen, Kuo e Sheu [Chen et al., 1988], que apresentaram um algoritmo de tempo  $O(|V|)$ , Lozin e Milanič [Lozin & Milanič, 2008] que apresentaram uma estratégia algorítmica de tempo  $O(|V|\mathcal{T})$  em que  $\mathcal{T}$  é o tempo necessário para resolver o problema em grafos cujos subconjuntos

de quatro vértices possuem mais de três arestas conectando-os, e Keil et al. [Keil et al., 2017] que apresentaram um algoritmo de tempo  $O(|V|^3)$ , entre outros.

Estudos e exposições mais aprofundadas sobre a teoria dos grafos podem ser encontrados nos livros de Bondy e Murty [Bondy & Murty, 2008], Gibbons [Gibbons, 1985], Godsil e Royle [Godsil & Royle, 2001] e Szwarcfiter [Szwarcfiter, 2018], entre outros.

## 2.4 Teoria dos jogos algorítmica

A teoria dos jogos algorítmica (TJA) estabelece um elo entre a teoria dos jogos e a teoria da computação. Ela preocupa-se em analisar resultados da teoria dos jogos do ponto de vista computacional, pois as aplicações desta teoria podem enfrentar dificuldades no que tange à grandeza das instâncias envolvidas. Por outro lado, a TJA ocupa-se em abordar problemas clássicos da computação – mormente os de otimização – do ponto de vista da teoria dos jogos, considerando que os agentes visam maximizar seus ganhos e/ou minimizar suas perdas de acordo com as decisões tomadas pelo conjunto de agentes.

### 2.4.1 Problemas de precificação

Uma classe de problemas estudados em teoria dos jogos algorítmicas consiste em *problemas de precificação*, nos quais um vendedor deseja vender itens a vários consumidores, atribuindo preços a cada um dos itens de modo a maximizar o seu lucro/receita. Nesses ambientes, as preferências e características dos dos consumidores são levadas em consideração no processo de decisão pois, se os preços estiverem além do alcance orçamentário dos consumidores, então o vendedor terá dificuldades em vender seus produtos. De outro modo, se os preços forem muito baixos, o vendedor pode ter seu lucro/receita comprometido.

Dentre alguns dos problemas pertencentes à essa classe, encontram-se:

- **problema da compra máxima:** dadas valorações dos consumidores atribuídas aos itens, encontrar uma alocação e uma precificação que maximizam a receita do vendedor de forma que cada comprador recebe um item cujo

preço está dentro do seu orçamento, ou seja, o preço do item é menor ou igual à valor que o comprador atribui a esse item;

- **problema da compra por preferência:** dadas valorações e preferências dos consumidores atribuídas aos itens, encontrar uma alocação e uma precificação que maximizam a receita do vendedor de forma que cada comprador recebe seu item cujo preferido, se existir, de acordo com seu orçamento;
- **problema da compra mínima:** dadas valorações dos consumidores atribuídas aos itens, encontrar uma alocação e uma precificação que maximizam a receita do vendedor de forma que cada comprador recebe um item de menor preço que está dentro do seu orçamento.

O problema de precificação estudado neste trabalho é o EFPP, que será definido formalmente na próxima subseção. Uma visão mais aprofundada sobre a teoria dos jogos algorítmica e problemas de precificação pode ser encontrada nos textos de Nisan et al. [Nisan et al., 2007] e Schouery et al. [Schouery et al., 2015], dentre outros.

### 2.4.2 Problema da precificação livre de inveja

Para se definir o EFPP, assume-se que existe um conjunto  $I$  de  $m$  consumidores e um conjunto  $J$  de  $n$  itens diferentes. Cada item possui  $c_j$  cópias, e então um *vetor de disponibilidade* é dado por  $\mathbf{c} = (c_1, c_2, \dots, c_n) \in \mathbb{N}^n$ . Cada consumidor possui uma *valoração*  $v_i(S)$  para cada pacote  $S \subseteq J$  de itens (por conveniência, é assumido que  $v_i(\emptyset) = 0$  para todo consumidor  $i \in I$ ); a *matriz de valorações*  $m \times 2^n$  (cujas linhas representam os consumidores, as linhas os pacotes e cada entrada  $v_{iS}$  representa a valoração do consumidor  $i$  ao pacote  $S$ ) é denotada por  $V \in \mathbb{R}_{\geq 0}^{m \times 2^n}$ .

Dado um *vetor de preços*  $\mathbf{p} = (p_1, p_2, \dots, p_n) \in \mathbb{R}_{\geq 0}^n$ , a *utilidade* que o consumidor  $i$  obtém do pacote  $S$  é  $U_i(S) = v_i(S) - p_S$ , em que  $p_S = \sum_{j \in S} p_j$ . Se a utilidade do consumidor  $i$  para o pacote  $S$  é não-negativa, é dito que  $S$  é *factível* para  $i$ .

O *conjunto de demanda*  $D_i$  do consumidor  $i$  contém os pacotes que o deixam mais satisfeito. Formalmente,  $D_i = \{S \mid U_i(S) = \max_{S'} U_i(S')\}$ . Como não

comprar nenhum pacote é sempre uma opção com utilidade  $U_i(\emptyset) = 0$ , segue que  $U_i(S) \geq 0$  para todo  $S \in D_i$ .

De acordo com esta terminologia, define-se:

**Definição 2.4.1.** *Uma alocação  $(S_1, S_2, \dots, S_m)$  para os consumidores é factível se cada item  $j$  está em no máximo  $c_j$  conjuntos  $S_i$ .*

**Definição 2.4.2.** *Dada uma precificação  $\mathbf{p} = (p_1, p_2, \dots, p_n)$ , uma alocação  $(S_1, S_2, \dots, S_m)$  é livre de inveja se  $S_i \in D_i$  para  $i$ , isto é, cada consumidor recebe um pacote de seu conjunto de demanda.*

**Definição 2.4.3.** *Uma precificação  $\mathbf{p}$  é livre de inveja se ela admite uma alocação factível livre de inveja*

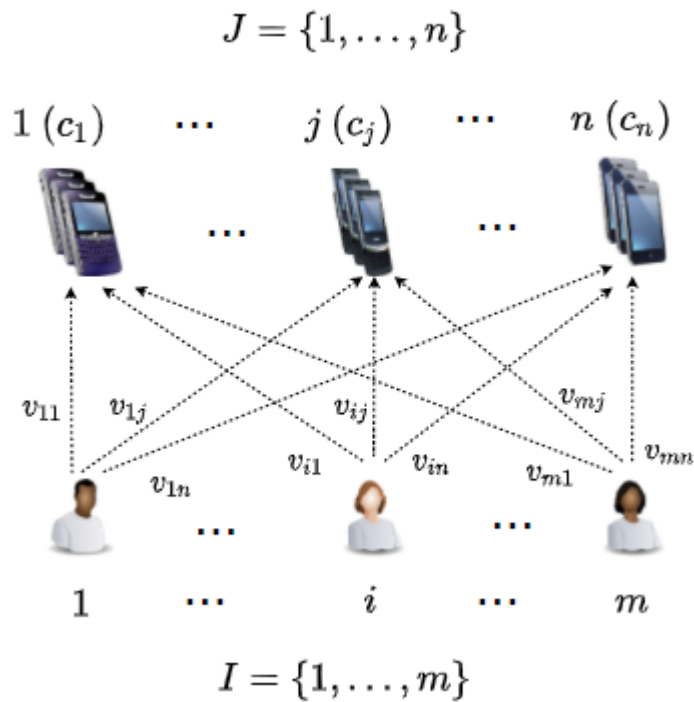
Então, o EFPP consiste em: dada a entrada  $(m, n, V, \mathbf{c})$ , calcular uma precificação livre de inveja  $\mathbf{p}$  e uma alocação livre de inveja correspondente  $(S_1, S_2, \dots, S_m)$  de forma a maximizar a receita do vendedor  $\sum_{i=1}^m p_{S_i}$ .

Dois casos de demanda do EFPP são os mais abordados:

- *Consumidores com demanda unitária:* cada consumidor  $i$  está interessado em comprar exatamente um item, e então  $v_i(S) > 0$  somente quando  $|S| = 1$ . Logo, a dimensão da matriz de valorações se reduz de  $m \times 2^n$  para  $m \times n$ , e a entrada  $v_{ij}$  representa a valoração do consumidor  $i$  para o item  $j$ . A Figura ilustra este caso
- *Consumidores com demanda por pacotes específicos:* cada consumidor  $i$  está interessado em somente um pacote específico de itens  $S_i$ . Logo, as valorações podem ser resumidas em um conjunto de pares  $(v_i, S_i)$ , significando que o consumidor  $i$  está interessado no pacote  $S_i$ , e atribui o valor  $v_i$  a ele.

Em ambos os casos o EFPP é APX-difícil, ainda que os itens não tenham limite de disponibilidade [Guruswami et al., 2005]. Um problema que está intimamente relacionado ao EFPP é o de encontrar uma precificação e uma alocação de forma a obter um *equilíbrio walrasiano* (EW), também chamado de *equilíbrio competitivo*. Um EW consiste numa precificação  $\mathbf{p}$  e numa alocação de forma que  $\mathbf{p}$  é livre de inveja e os itens não alocados possuem preço igual a zero. Esta é uma





**Figura 2.1.** Exemplo de leilão combinatório com demanda unitária. No EFPP com demanda unitária, o vendedor precisa definir uma alocação e precificação de forma que tanto a sua receita quanto a utilidade dos compradores sejam maximizadas.

definição formal para a clássica noção de equilíbrio de mercado: oferta = demanda (no contexto de ausência de inveja). Um cenário economicamente interessante é aquele no qual todos os itens são vendidos, ocasionando um balanceamento do mercado.

Uma variação possível de EW é obtida incorporando-se *preços de reserva*. Um preço de reserva para um item  $j$  é o menor preço pelo qual o vendedor está disposto a comercializar este item. Dessa forma, um *equilíbrio walrasiano com preços de reserva* consiste numa precificação livre de inveja  $\mathbf{p}$  e numa alocação de forma que, dado um vetor de preços de reserva  $\mathbf{r} = (r_1, \dots, r_n)$ , se tenha  $p_j \geq r_j$  para todo  $j$ , e se um item  $j$  não é vendido, seu preço é  $r_j$ .

## 2.5 Considerações finais

Neste capítulo foram apresentadas as definições formais dos conceitos envolvidos nos problemas estudados nesta tese. Uma formulação em programação inteira mista será utilizada na formulação de um algoritmo para o caso de *emparelhamento perfeito livre de inveja* – um caso particular do EFPP a ser definido formalmente no Capítulo 4. Em relação às técnicas de projeto de algoritmos, a programação dinâmica, será utilizada no problema acima citado, enquanto a estratégia gulosa será utilizada na formulação de um algoritmo para o caso de *demanda unitária com substituíbilidade métrica* – a ser definido formalmente no Capítulo 5. Com relação às técnicas de análise de algoritmos, a notação *Big-O* foi explorada, pois será utilizada na análise de complexidade de tempo dos algoritmos propostos nesta tese, bem como na comparação destes com os encontrados na literatura.

A teoria dos grafos, bem como alguns dos algoritmos utilizados para resolver problemas clássicos desta área, foram apresentados pois também serão utilizados nos algoritmos propostos nesta tese. Tanto o método húngaro – que resolve o problema do emparelhamento máximo – quanto o algoritmo de Bellman-Ford, que resolve o problema do caminho mínimo, serão utilizados nas formulações dos algoritmos propostos nesta tese e na comparação destes com os encontrados na literatura para ambos os casos citados no parágrafo anterior. A apresentação do problema do conjunto independente de peso máximo é útil para se compreender a estratégia utilizada na literatura para resolver o caso de demanda unitária com substituíbilidade métrica e compará-la com a estratégia utilizada nesta tese.

Por fim, o EFPP – que insere-se na teoria dos jogos algorítmica – foi apresentado formalmente, pois ele é o foco de estudo desta tese. No próximo capítulo será apresentada uma revisão da literatura sobre resultados obtidos para o EFPP, bem como para alguns de seus casos particulares e variações.

# Revisão da Literatura

---

Neste capítulo serão descritos trabalhos encontrados na literatura sobre formulações MIP, casos ótimos polinomiais, estratégias heurísticas, algoritmos aproximativos e resultados sobre inaproximabilidade para casos do EFPP e algumas de suas variações.

## 3.1 Formulações MIP

Em 1988, Dobson e Kalish [Dobson & Kalish, 1988] propuseram uma formulação MIP para um problema de posicionamento e precificação de linhas de produtos que viria a ser utilizada posteriormente no EFPP por outros autores. Eles mesmos adaptaram esta formulação quando se incorporam ao problema uma análise conjunta de dados de custos [Dobson & Kalish, 1993]. Na primeira formulação, já se utilizou a estratégia de decomposição do problema em dois subproblemas: alocação e precificação. Na segunda formulação, os autores utilizaram a vantagem de a precificação poder ser obtida através da solução dual de um problema de fluxo de custo mínimo [Magnanti et al., 1993].

Esta última formulação de Dobson e Kalish foi estendida por Day e Venkataramanan [Day & Venkataramanan, 2006] ao incorporar custos fixos comuns de manufatura e decisões de composição dos produtos. A não linearidade da formulação proposta por eles foi resolvida (ou seja, fez-se uma linearização do modelo) através da fixação da precificação e, além disso, as variáveis contínuas que modelam os preços foram reduzidas a variáveis discretas, tendo em vista a aplicação do

modelo em situações reais embora esta redução limitasse o lucro obtido através da modelagem.

Günlük [Günlük, 2008] apresentou um modelo MIP para um problema de precificação de linhas de produtos no qual os consumidores possuem demanda não uniforme para um de vários itens com disponibilidade ilimitada, o que chamaram de problema de design de linhas de produtos ótimas na ausência de competição. Este problema também captura a maximização da utilidade dos compradores, assim como o EFPP.

Shioda, Tunçel e Myklebust [Shioda et al., 2011] formularam um modelo MIP para o EFPP no caso de demanda unitária em que os consumidores são agrupados em segmentos e os produtos possuem disponibilidade ilimitada, propondo várias desigualdades válidas. Mais tarde, Myklebust, Sharpe e Tunçel [Myklebust et al., 2016] abordaram o mesmo problema com a mesma formulação propondo novas heurísticas mais eficientes.

Fernandes et al. [Fernandes et al., 2016] apresentaram duas novas formulações MIP para o EFPP com demanda unitária e disponibilidade ilimitada de itens, ao mesmo tempo em que propuseram três modelos geradores de instâncias que proporcionam interpretações econômicas no que diz respeito às características dos produtos, às localizações dos consumidores em relação aos produtos e à popularidade dos produtos.

A Tabela 3.1 reúne e sintetiza os trabalhos citados nesta seção. Pode-se notar que, na maioria das interpretações econômicas das modelagens propostas, o principal foco foi a maximização da receita e do lucro, sendo a satisfação dos consumidores deixado como algo secundário. A maximização da satisfação dos consumidores possui não apenas um apelo social, mas também pode produzir soluções computacionais mais eficientes, como será visto no Capítulo 4.

## 3.2 Casos polinomiais

Günlük [Günlük, 2008] provou que se as valorações dos consumidores satisfazem a propriedade de Monge, isto é, se na matriz que as representa qualquer submatriz  $2 \times 2$  possuir a característica de a soma dos elementos da diagonal ser maior ou igual à soma dos outros elementos, o caso do EFPP com demanda não uniforme

**Tabela 3.1.** Formulações MIP para o EFPP e problemas correlatos

<b>Problema</b>	<b>Função objetivo</b>	<b>[Autor, ano]</b>
Posicionamento e precificação de linhas de produtos	Maximização do lucro	[Dobson & Kalish, 1988]
Posicionamento e precificação de linhas de produtos usando análise conjunta de dados de custo	Maximização do lucro/bem-estar social	[Dobson & Kalish, 1993]
Precificação de linhas de produtos incorporando custos fixos comuns de manufatura e decisões de composição dos produtos	Maximização do lucro	[Day & Venkataramanan, 2006]
Design de linhas de produtos ótimas na ausência de competição	Maximização da receita	[Günlük, 2008]
EFPP com demanda unitária, consumidores agrupados em segmentos e disponibilidade ilimitada de produtos	Maximização da receita	[Shioda et al., 2011]
EFPP com demanda unitária, consumidores agrupados em segmentos e disponibilidade ilimitada de produtos	Maximização da receita	[Myklebust et al., 2016]
EFPP com demanda unitária e disponibilidade ilimitada de produtos	Maximização da receita	[Fernandes et al., 2016]

– ou seja, as cópias de itens desejadas varia de consumidor para consumidor – e disponibilidade ilimitada de itens pode ser resolvido em tempo polinomial  $O(mn^2)$ , em que  $m$  é o número de consumidores e  $n$  é o número de itens, através de pro-

gramação dinâmica baseada no MIP formulado.

Chen, Gosh e Vassilvitskii [Chen et al., 2011] mostraram que o caso do EFPP com demanda unitária e substituíbilidade métrica no qual os custos de deslocamento formam um espaço métrico se reduz ao problema do conjunto independente de peso máximo em grafos perfeitos, e portanto pode ser resolvido em tempo polinomial  $O(n^4)$ .

Chen e Deng [Chen & Deng, 2014] provaram que o caso do EFPP em que cada item possui disponibilidade de uma única cópia, a demanda é unitária e cada consumidor valora positivamente no máximo dois itens, também se reduz ao problema do conjunto independente de peso máximo em grafos perfeitos, e portanto pode ser resolvido em tempo polinomial  $O(n^4)$ .

Já Chen et al. [Chen et al., 2016] consideraram um caso do EFPP no qual cada item possui disponibilidade de uma única cópia, os consumidores desejam um ou mais itens e só aceitam comprar o número de itens desejados exatamente ou não compram nenhum item, o que chamaram de demanda de multi-itens exata. Além disso, cada consumidor  $i$  possui uma única valoração  $v_i$  e cada item possui uma métrica de qualidade  $q_j$ , e daí o valor do item  $j$  ao consumidor  $i$  é definido por  $v_i q_j = v_{ij}$ ,  $\forall i = 1, \dots, m$ ,  $\forall j = 1, \dots, n$ . Dessa forma, eles mostraram que o EFPP pode ser resolvido em tempo polinomial  $O(nm^3 \log n)$  se a demanda de cada consumidor é limitada por uma constante.

Eles construíram uma estratégia na qual primeiro se encontram conjuntos de 'candidatos' a compradores, e então um programa linear caracteriza o conjunto ótimo. Logo após, um algoritmo em programação dinâmica encontra a alocação. Ambos programa linear e algoritmo efetuam a busca em tempo polinomial. Ao final um algoritmo em tempo polinomial que utiliza outro programa linear resolvível polinomialmente toma a alocação ótima como entrada e produz a precificação ótima como saída.

Arbib, Karaşan e Pınar [Arbib et al., 2019] provaram que o EFPP pode ser resolvido em tempo polinomial  $O(n^3)$  se cada item possui disponibilidade de uma única cópia, o número de compradores coincide com o número de itens, e a demanda é unitária, ao que chamaram de emparelhamento perfeito livre de inveja (EFPM – *envy-free perfect matching*, em inglês). Eles consideraram os subproblemas da alocação e da precificação separadamente e mostraram que a

alocação pode ser obtida por um emparelhamento perfeito de peso máximo e a precificação por caminhos mínimos, após a alocação ser determinada.

A Tabela 3.2 faz um resumo dos resultados apresentados nesta seção. Nota-se que as menores complexidades de tempo são alcançadas quando algumas particularidades são consideradas: quando se tem a propriedade de Monge na matriz de valorações [Günlük, 2008], e quando o número de consumidores coincide com o número de itens à venda, ou seja,  $m = n$  [Arbib et al., 2019].

### 3.3 Heurísticas e meta-heurísticas

Dobson e Kalish [Dobson & Kalish, 1988, 1993], Shioda, Tunçel e Myklebust [Shioda et al., 2011] e Myklebust, Sharpe e Tunçel [Myklebust et al., 2016] utilizaram estratégias semelhantes baseadas nos MIPs formulados. Algoritmos gulosos foram utilizados para determinar as alocações factíveis e então estas soluções alimentam o próximo passo da precificação a ser obtida através de caminhos mínimos. As alocações são rearranjadas e o processo repetido até que não se aumente o valor da receita total.

A principal diferença entre as estratégias utilizadas reside na determinação das alocações. Enquanto Dobson e Kalish [Dobson & Kalish, 1988, 1993] iniciam o cálculo da precificação com uma alocação que leva em consideração as maiores valorações para cada item, Shioda, Tunçel e Myklebust [Shioda et al., 2011] o fazem através de um rearranjo da alocação de modo a obter uma precificação sempre crescente através da modelagem do problema por árvores geradoras, obtendo assim um melhor desempenho de tempo na busca de soluções. Já Myklebust, Sharpe e Tunçel [Myklebust et al., 2016] propuseram realizar iterações utilizando a estratégia anterior até que se atinja um ponto fixo, obtendo um desempenho de tempo ainda melhor na busca de soluções.

Day e Venkataramanan [Day & Venkataramanan, 2006] utilizaram dois algoritmos genéticos em sequência para estimar a solução ótima. O primeiro algoritmo retorna a receita total após determinar alocações que maximizam as utilidades dos compradores. O segundo determina as precificações ótimas de acordo com as alocações produzidas pelo primeiro através de chamadas do MIP formulado.

Tabela 3.2. Casos especiais ótimos do EFPP

	Características			Estratégia algorítmica	Complexidade de tempo	[Autor, ano]
Disponibilidade de itens	Demanda	Outros				
ilimitada	não uniforme	valorações satisfazendo a propriedade de Monge		programação dinâmica	$O(mn^2)$	[Güntik, 2008]
ilimitada	unitária	os custos de deslocamento de localidade formam um espaço métrico		redução ao problema do conjunto independente de peso máximo em grafos perfeitos	$O(n^4)$	[Chen et al., 2011]
limitada	unitária	cada consumidor valora positivamente no máximo dois itens		redução ao problema do conjunto independente de peso máximo em grafos perfeitos	$O(n^4)$	[Chen & Deng, 2014]
limitada	exata de múltiplos itens	cada item possui uma métrica de qualidade, e a demanda de cada consumidor é limitada por uma constante		programação dinâmica combinada com programas lineares	$O(mn^3 \log n)$	[Chen et al., 2016]
limitada	unitária	$m = n$		emparelhamento perfeito de peso máximo e caminhos mínimos	$O(n^3)$	[Arbib et al., 2019]



A Tabela 3.3 apresenta um resumo das estratégias heurísticas utilizadas para casos do EFPP e variações. Nota-se que, apesar das várias estratégias utilizadas para a determinação das alocações, as precificações quase sempre foram obtidas através da busca de caminhos mínimos nos grafos que modelam os problemas.

**Tabela 3.3.** Heurísticas e meta-heurísticas para casos e variações do EFPP

Problema	Estratégia algorítmica		[Autor, ano]
	Alocação	Precificação	
posicionamento e precificação de linhas de produtos	algoritmo guloso	caminhos mínimos	[Dobson & Kalish, 1988]
posicionamento e precificação de linhas de produtos	algoritmo guloso	caminhos mínimos	[Dobson & Kalish, 1993]
precificação de linhas de produtos	algoritmo genético	algoritmo genético	[Day & Venkataramanan, 2006]
EFPP	algoritmo guloso	caminhos mínimos	[Shioda et al., 2011]
EFPP	algoritmo guloso	caminhos mínimos	[Myklebust et al., 2016]

### 3.4 Algoritmos aproximativos/inaproximabilidade

Guruswami et al. [Guruswami et al., 2005] utilizaram emparelhamentos de peso máximo das matrizes de valoração na construção de um algoritmo  $\log m$ -aproximativo para o EFPP com demanda unitária e disponibilidade ilimitada de itens. Para o caso de demanda por pacotes específicos, uma simples estratégia gulosa fornece uma  $\log m + \log n$ -aproximação.

Briest [Briest, 2008] estabeleceu uma conexão do EFPP com demanda unitária e disponibilidade ilimitada de itens com o problema do conjunto independente bipartido balanceado [Khot, 2006] para mostrar que não é possível uma aproximação em tempo polinomial melhor que  $\log^\epsilon m$ , para algum  $\epsilon > 0$ . Tal conexão

foi feita através do problema de seqüências de expansão máxima [Briest & Krysta, 2011], que relaciona diferentes níveis de preços do EFPP com demanda unitária a conjuntos independentes em grafos. Para o caso de demanda por pacotes específicos, ele mostrou que é  $R3SAT^*(2^{O(n^\delta)})$ -difícil aproximar o problema em  $O(n^\epsilon)$ , para todo  $\delta > 0$  e algum  $\epsilon > 0$ .

Demaine et al. [Demaine et al., 2008] reduziram o problema da cobertura única [Demaine et al., 2006] ao EFPP com demanda por pacotes específicos e disponibilidade ilimitada de itens para obter uma aproximação  $O(\log n)$ . Tal redução toma uma coleção da instância do problema da cobertura única e a transforma num item do EFPP, e cada elemento do conjunto universo é transformado num consumidor. Em relação às valorações, conjuntos que contém elementos específicos são transformados em pacotes que contém itens correspondentes ao consumidor que os desejam e, dessa forma, atribui-se a valoração 1 a esses pacotes. Consequentemente, os preços terão valores 0 ou 1 (logo livres de inveja), e maximizar o número de pacotes com exatamente um item de preço 1 equivale a encontrar uma cobertura única no outro problema.

Chakermsook et al. [Chalermsook et al., 2012] mostraram que o EFPP com demanda tanto unitária quanto por pacotes específicos é  $\log^{1-\epsilon}(m+n)$ -difícil de aproximar através de reduções a partir do problema do conjunto independente máximo. As estratégias de redução tomam um instância deste último problema obtida a partir de uma fórmula  $3SAT$  [Karp, 1972] de onde é possível construir um grafo de grau máximo até  $\Delta = 2^{\lambda \text{ poly}(\frac{1}{\epsilon})}$ , em que  $\epsilon > 0$  é uma constante qualquer suficientemente pequena e  $\lambda > 0$  é um inteiro qualquer, e a transformam numa instância intermediária de consumidores "virtuais", até se chegar numa instância do EFPP que reúne os conjuntos e as preferências destes consumidores fictícios, dessa forma capturando as características das instâncias do EFPP em si.

No caso de demanda múltipla exata e disponibilidade limitada de itens, Bilò, Flammini e Monaco [Bilò et al., 2017] apresentaram uma estratégia gulosa obtendo uma  $m$ -aproximação para o EFPP, ao mesmo tempo em que provaram ser impossível obter uma  $O(m^{1-\epsilon})$ -aproximação a menos que  $P = NP$ . Esta impossibilidade foi provada através de uma redução a partir do problema do particionamento restrito [Burkard & Yao, 1990].

A Tabela 3.4 sintetiza os resultados descritos nesta seção. Nota-se que para

os casos de demanda unitária as ordens de aproximação e inaproximabilidade dependem exclusivamente do número de consumidores, enquanto que nos casos de demanda por pacotes específicos estas ordens dependem exclusivamente do número de itens. Já quando as demandas são consideradas em conjunto, as ordens dependem de ambas as quantidades. Estes fatos sugerem que considerar o número de consumidores igual ao número de itens pode reduzir a complexidade de tempo na busca de soluções para o EFPP.

### 3.5 Considerações finais

A revisão apresentada neste capítulo teve como objetivo apontar direções para as quais é possível projetar algoritmos exatos para casos e/ou variantes do EFPP, e por outro lado os resultados sobre inaproximabilidade, algoritmos aproximativos heurísticos e meta-heurísticos para casos NP-difíceis mostram que, para estes casos, soluções ótimas não podem sempre ser obtidas em tempo polinomial. Modelagens em termos de formulações MIP são úteis no desenvolvimento de estratégias algorítmicas tanto exatas quanto aproximadas, e as técnicas de projeto de algoritmos utilizadas em todos os cenários abordados na seção fornecem uma intuição para o desenvolvimento de algoritmos eficientes que encontram soluções em casos especiais ótimos.

Em posse disso, o restante do trabalho apresentará as estratégias e os resultados obtidos para os casos em que foi possível projetar algoritmos para casos ótimos mais eficientes que os encontrados na literatura, a saber: o caso de emparelhamento perfeito livre de inveja e o caso de demanda unitária com substituíbilidade métrica.

Tabela 3.4. Algoritmos aproximativos e inaproximabilidade para casos do EPPP

Disponibilidade de itens	Características	Demanda	Aproximação	Inaproximabilidade	(Autor, ano)
limitada	unitária		$O(\log m)$	–	[Guruswami et al., 2005]
ilimitada	pacotes específicos		$O(\log m + \log n)$	–	[Guruswami et al., 2005]
ilimitada	unitária		–	$O(\log^\epsilon m)$ , $\epsilon > 0$	[Briest, 2008]
ilimitada	pacotes específicos		–	$O(n^\epsilon)$ , $\epsilon > 0$	[Briest, 2008]
ilimitada	pacotes específicos		$O(\log n)$	–	[Demaine et al., 2008]
ilimitada	unitária e pacotes específicos		–	$O(\log^{1-\epsilon}(m+n))$ , $\epsilon > 0$	[Chalermsook et al., 2012]
limitada	múltipla exata		$O(m)$	$O(n^{1-\epsilon})$ , $\epsilon > 0$	[Bilò et al., 2017]

# O caso de emparelhamento perfeito livre de inveja

---

No EFPM, o número de itens coincide com o número de consumidores, ou seja,  $m = n$ , cada item possui uma única cópia disponível e cada consumidor pode comprar apenas uma cópia de um item. Neste capítulo serão apresentados resultados teóricos e experimentais de uma estratégia que aumenta a performance na busca de preços livres de inveja ótimos em relação àquela proposta por Arbib, Karaşan e Pinar [Arbib et al., 2019].

## 4.1 Preliminares

Considere as variáveis de decisão  $x_{ij}$  tais que  $x_{ij} = 1$  se o consumidor  $i$  recebe o item  $j$ , e  $x_{ij} = 0$  caso contrário. Arbib, Karaşan e Pinar [Arbib et al., 2019] propuseram a seguinte formulação MIP que modela o EFPM:

$$\max \sum_{j=1}^n p_j \tag{4.1}$$

$$\text{s. a. } \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in I \tag{4.2}$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in I \tag{4.3}$$

$$p_j - \sum_{i=1}^n x_{ij} v_{ij} \leq 0, \quad \forall j \in I \quad (4.4)$$

$$p_j - p_k - \sum_{i=1}^n (v_{ij} - v_{ik}) x_{ij} \leq 0, \quad \forall j, k \in I \quad (4.5)$$

$$p_j \in \mathbb{R}_{\geq 0}, \quad \forall j \in I \quad (4.6)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in I \quad (4.7)$$

A função objetivo (4.1) representa a receita total do vendedor obtida com a venda dos itens. Os conjuntos de equações (4.2) e (4.3) junto com o domínio (4.7) garantem o emparelhamento perfeito. O conjunto de restrições (4.4) assegura que os itens estão nos conjuntos de demanda dos compradores e, somado a isto, (4.5) fornece preços que maximizam as utilidades deles, sendo, dessa forma, livres de inveja: os preços são obtidos por

$$v_{ij} x_{ij} - p_j x_{ij} \geq v_{ik} x_{ij} - p_k, \quad \forall i, j, k \in I,$$

daí  $(v_{ij} - v_{ik}) x_{ij} \geq p_j x_{ij} - p_k$ , e efetuando a soma sobre todo  $i \in I$ , tem-se

$$\sum_{i=1}^n (v_{ij} - v_{ik}) x_{ij} \geq \sum_{i=1}^n p_j x_{ij} - p_k = p_j \sum_{i=1}^n x_{ij} - p_k = p_j - p_k.$$

O domínio dos preços é descrito em (4.6).

Considere a seguinte formulação MIP (maximização do bem estar social):

$$\max \sum_{i=1}^n \sum_{j=1}^n x_{ij} v_{ij} \quad (4.8)$$

$$\text{s. t.} \quad \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in I \quad (4.9)$$

$$\sum_{i=1}^n x_{ij} = 1, \quad \forall j \in I \quad (4.10)$$

$$x_{ij} \in \{0, 1\}, \quad \forall i, j \in I. \quad (4.11)$$

**Lema 4.1.1.** *(caso particular do Primeiro Teorema do Bem Estar Social) Se  $X = (x_{ij}) \in \{0, 1\}^{n \times n}$  é uma solução para o modelo (4.8) – (4.11) e  $\mathbf{p} \in \mathbb{R}_{\geq 0}^n$  é uma*

solução (parcial) para o modelo (4.1) – (4.7), então  $(X, \mathbf{p})$  é uma solução para o modelo (4.1) – (4.7).

*Demonstração.* Seja  $X' = (x'_{ij}) \in \{0, 1\}^{n \times n}$  uma solução (parcial) para o modelo (4.1) – (4.7). Como  $X$  maximiza (4.8),

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} v_{ij} \geq \sum_{i=1}^n \sum_{j=1}^n x'_{ij} v_{ij}$$

e então

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} v_{ij} - \sum_{j=1}^n p_j \geq \sum_{i=1}^n \sum_{j=1}^n x'_{ij} v_{ij} - \sum_{j=1}^n p_j.$$

As restrições (4.9), (4.10), (4.2) e (4.3) permitem que esta última desigualdade seja reescrita como

$$\sum_{i=1}^n \sum_{j=1}^n x_{ij} v_{ij} - \sum_{i=1}^n \sum_{j=1}^n x_{ij} p_j \geq \sum_{i=1}^n \sum_{j=1}^n x'_{ij} v_{ij} - \sum_{i=1}^n \sum_{j=1}^n x'_{ij} p_j$$

o que implica

$$\sum_{i=1}^n \sum_{j=1}^n (v_{ij} - p_j) x_{ij} \geq \sum_{i=1}^n \sum_{j=1}^n (v_{ij} - p_j) x'_{ij}. \quad (4.12)$$

No entanto, como  $X'$  maximiza as utilidades dos compradores, segue que

$$\sum_{i=1}^n \sum_{j=1}^n (v_{ij} - p_j) x'_{ij} \geq \sum_{i=1}^n \sum_{j=1}^n (v_{ij} - p_j) x_{ij}. \quad (4.13)$$

Juntando (4.12) e (4.13), segue que  $X$  também maximiza as utilidades dos compradores, e portanto  $(X, \mathbf{p})$  é uma solução para o modelo (4.1) – (4.7).  $\square$

Conforme visto acima, o Primeiro Teorema do Bem Estar Social [Nisan et al., 2007] assegura que uma alocação que maximiza o bem estar social também maximiza a receita com uma precificação livre de inveja. Então, encontrar uma alocação ótima para o EFPM equivale a encontrar um emparelhamento perfeito de peso máximo no grafo associado à matriz de valorações  $V$ . Seja  $X = (x_{ij}) \in \{0, 1\}^{n \times n}$  a matriz binária associada ao emparelhamento perfeito de peso máximo. Tem-se o seguinte Lema:

**Lema 4.1.2.** *Seja  $\{i_1, i_2, \dots, i_r\}$  um subconjunto de  $I$  e  $V' = X^T V = (v'_{ij})$ . Então*

$$v'_{i_2 i_1} + v'_{i_3 i_2} + \dots + v'_{i_r i_{r-1}} + v'_{i_1 i_r} \leq v'_{i_1 i_1} + v'_{i_2 i_2} + \dots + v'_{i_{r-1} i_{r-1}} + v'_{i_r i_r}.$$

*Demonstração.* Se a desigualdade não fosse válida, existiria uma matriz de permutação  $\Pi$  tal que

$$\pi_{i_1 i_2} = \pi_{i_2 i_3} = \dots = \pi_{i_{r-1} i_r} = \pi_{i_r i_1} = 1,$$

e então  $tr(\Pi^T V) > tr(X^T V)$ , o que contradiz a otimalidade de  $X$ .  $\square$

Uma vez que a alocação ótima é encontrada, considere a matriz  $\mathcal{U} = (u_{ij}) \in \mathbb{R}^{n \times n}$  tal que  $u_{ij} = v'_{ij} - v'_{jj}$ . Esta matriz herda a propriedade fundamental da matriz  $V'$  mostrada no Lema 4.1.2, que é afirmada no seguinte Corolário:

**Corolário 4.1.3.** *Seja  $\{i_1, i_2, \dots, i_r\}$  um subconjunto de  $I$ . Então*

$$u_{i_1 i_r} + \sum_{k=1}^{r-1} u_{i_{k+1} i_k} \leq 0$$

*Demonstração.* A matriz  $\mathcal{U}$  é obtida adicionando-se constantes (negativas) a cada coluna de  $V'$ , o que introduz as constantes relativas às colunas  $i_1, i_2, \dots, i_r$  em ambos os lados da desigualdade do Lema 4.1.2. Como os elementos da diagonal de  $\mathcal{U}$  são nulos, o resultado segue.  $\square$

O seguinte Lema relaciona a matriz  $\mathcal{U}$  com as utilidades dos compradores:

**Lema 4.1.4.** *Se  $y_1, y_2, \dots, y_n$  são números não negativos mínimos tais que*

$$y_j \geq y_k + u_{jk}, \quad \forall j, k \in I, j \neq k, \quad (4.14)$$

*então estes números são as utilidades máximas dos compradores.*

*Demonstração.* De fato, pondo  $p_j = v'_{jj} - y_j$ ,  $\forall j \in I$ , tem-se

$$v'_{jj} - p_j = y_j \geq y_k + u_{jk} = v'_{kk} - p_k + v'_{jk} - v'_{kk} = v'_{jk} - p_k, \quad \forall j, k \in I, j \neq k,$$

e o resultado segue.  $\square$



**Proposição 4.1.5.** *Os números  $y_1, y_2, \dots, y_n$  podem ser encontrados resolvendo-se o programa linear*

$$\min \sum_{j=1}^n y_j \quad (4.15)$$

$$s. t. \quad y_j - y_k \geq u_{jk} \quad \forall j, k \in I, j \neq k \quad (4.16)$$

$$y_j \in \mathbb{R}_{\geq 0} \quad \forall j \in I. \quad (4.17)$$

*Demonstração.* A função objetivo, as restrições e o domínio satisfazem as hipóteses do Lema 4.1.4.  $\square$

## 4.2 Sequências de utilidades

Nesta seção, serão apresentadas e analisadas sequências numéricas não decrescentes que representam as utilidades dos consumidores. Será mostrado que estas sequências convergem para as utilidades máximas dos consumidores em no máximo  $n - 1$  passos, e então elas serão utilizadas para se calcular os preços ótimos livres de inveja.

Para cada  $j \in I$ , considere a sequência  $(y_j^{(t)})_{t \in \mathbb{N}}$  definida a seguir:

$$y_j^{(0)} = 0, \quad y_j^{(t+1)} = \max_{k \in I} \{y_k^{(t)} + u_{jk}\}. \quad (4.18)$$

**Teorema 4.2.1.** *As sequências definidas em (4.18) são convergentes, e convergem em no máximo  $n - 1$  passos.*

*Demonstração.* Para se visualizar este fato, constrói-se uma árvore enraizada no nó 0, e os outros nós são criados pelo seguinte procedimento: após o primeiro passo, se  $y_j^{(1)} > 0$ , então o nó  $j$  é um filho da raiz. Nos próximos passos, se

$$y_k^{(t)} + u_{jk} = \max_{l \in I} \{y_l^{(t)} + u_{jl}\},$$

então o nó  $j$  é um filho do nó  $k$  que está no nível  $t$  (se existir outro  $k'$  que satisfaz esta condição, escolha o menor, sem prejuízo para o procedimento). Como um exemplo, considere a matriz de valorações

$$V = \begin{bmatrix} 74 & 35 & 125 & 108 & 35 \\ 109 & 109 & 120 & 10 & 48 \\ 116 & 41 & 109 & 78 & 13 \\ 24 & 58 & 110 & 3 & 49 \\ 122 & 106 & 47 & 44 & 15 \end{bmatrix}$$

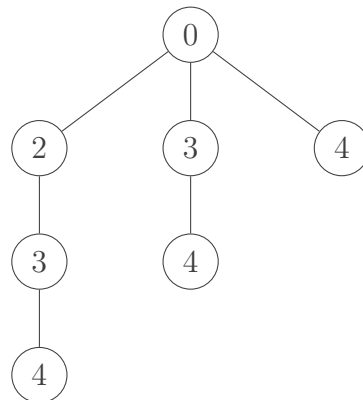
A matriz de permutação que maximiza  $tr(\Pi^T V)$  é

$$X = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Então, a matriz  $\mathcal{U}$  é

$$\mathcal{U} = \begin{bmatrix} 0 & -65 & -11 & -30 & -36 \\ 6 & 0 & -73 & -64 & -34 \\ -7 & 3 & 0 & -98 & -1 \\ -42 & -71 & 5 & 0 & -14 \\ -92 & -48 & -10 & -105 & 0 \end{bmatrix}$$

A árvore para esta matriz é mostrada na Figura 4.1.



**Figura 4.1.** A árvore associada à matriz  $\mathcal{U}$ , construída pelo procedimento.

Note que um caminho ligando qualquer nó  $j$  à raiz contém somente nós

distintos, porque se não fosse assim, suponha que exista um caminho contendo a repetição  $\{j_1, j_2, \dots, j_r, j_1\}$ . Ter-se-ia

$$\begin{aligned} y_{j_2}^{(s)} &< y_{j_1}^{(s)} + u_{j_2 j_1} \\ y_{j_3}^{(s+1)} &< y_{j_2}^{(s+1)} + u_{j_3 j_2} = y_{j_1}^{(s)} + u_{j_2 j_1} + u_{j_3 j_2} \\ &\vdots \\ y_{j_r}^{(s+r-3)} &< y_{j_{r-1}}^{(s+r-3)} + u_{j_r j_{r-1}} = y_{j_1}^{(s)} + u_{j_2 j_1} + u_{j_3 j_2} + \dots + u_{j_r j_{r-1}} \\ y_{j_1}^{(s+r-2)} &< y_{j_r}^{(s+r-2)} + u_{j_1 j_r} = y_{j_1}^{(s)} + u_{j_2 j_1} + u_{j_3 j_2} + \dots + u_{j_r j_{r-1}} + u_{j_1 j_r} \end{aligned}$$

para algum  $s \in \mathbb{N}$ , e então

$$u_{j_2 j_1} + u_{j_3 j_2} + \dots + u_{j_r j_{r-1}} + u_{j_1 j_r} > y_{j_1}^{(s+r-2)} - y_{j_1}^{(s)} \geq 0,$$

o que, pelo Corolário 4.1.3, é impossível. Assim, a altura da árvore é no máximo  $n - 1$ , e portanto

$$y_j^{(t+1)} = y_j^{(t)}, \quad \forall j \in I, \quad \forall t \geq n - 1, \quad (4.19)$$

daí o resultado segue.  $\square$

**Corolário 4.2.2.** *Existe pelo menos um  $j_0 \in I$  tal que*

$$y_{j_0}^{(t)} = 0, \quad \forall t \in \mathbb{N}.$$

*Demonstração.* De fato, como cada ramo da árvore construída na prova do Teorema 4.2.1 não contém repetições de nós, existe pelo menos um  $j_0 \in I$  tal que o nó  $j_0$  não está nela, daí o resultado segue.  $\square$

**Corolário 4.2.3.** *As sequências definidas em (4.18) convergem para números não negativos  $y_1, y_2, \dots, y_n$  satisfazendo (4.14).*

*Demonstração.* Suponha que existam números não negativos  $z_1, z_2, \dots, z_n$  satisfazendo

$$z_j \geq z_k + u_{jk}, \quad \forall j, k \in I, j \neq k,$$

com  $z_j \leq y_j$ ,  $\forall j \in I$ . Então, para cada  $j \in I$ , usando (4.19), tem-se

$$\begin{aligned} y_j &= y_j^{(n)} = y_{j_1}^{(n-1)} + u_{jj_1}, & \text{e } z_j &\geq z_{j_1} + u_{jj_1}, \\ y_{j_1}^{(n-1)} &= y_{j_2}^{(n-2)} + u_{j_1j_2}, & \text{e } z_{j_1} &\geq z_{j_2} + u_{j_1j_2}, \\ & & & \vdots \\ y_{j_{n-1}}^{(1)} &= y_{j_n}^{(0)} + u_{j_{n-1}j_n}, & \text{e } z_{j_{n-1}} &\geq z_{j_n} + u_{j_{n-1}j_n}. \end{aligned}$$

Subtraindo as equações à esquerda das desigualdades à direita, obtém-se

$$\begin{aligned} z_j - z_{j_1} &\geq y_j - y_{j_1}^{(n-1)}, \\ z_{j_1} - z_{j_2} &\geq y_{j_1}^{(n-1)} - y_{j_2}^{(n-2)}, \\ & \vdots \\ z_{j_{n-1}} - z_{j_n} &\geq y_{j_{n-1}}^{(1)} - y_{j_n}^{(0)}. \end{aligned}$$

Somando-se todas estas desigualdades, segue que

$$z_j - z_{j_n} \geq y_j - y_{j_n}^{(0)}.$$

Como  $z_{j_n} \geq 0$  e  $y_{j_n}^{(0)} = 0$ , tem-se

$$z_j \geq z_j - z_{j_n} \geq y_j - y_{j_n}^{(0)} \geq y_j.$$

Portanto,  $z_j = y_j \forall j \in I$ , daí o resultado segue.  $\square$

**Corolário 4.2.4.** *Para cada  $t = 0, \dots, n-1$ , existe um  $j_t$  tal que  $y_{j_t}^{(t)}$  é máximo.*

*Demonstração.* Com efeito, se não fosse assim, haveria repetição de nós em algum ramo da árvore construída na prova do Teorema 4.2.1 e, pelo Corolário 4.2.2, existe  $j_0 \in I$  tal que  $\max_{t=0, \dots, n-1} \{y_{j_0}^{(t)}\} = 0$ , daí o resultado segue.  $\square$

**Teorema 4.2.5.** *Seja  $\mathbf{p} = (p_1, p_2, \dots, p_n)$  tal que*

$$p_j = v'_{jj} - y_j, \quad \forall j \in I. \quad (4.20)$$

Então  $\mathbf{p}$  é um vetor de preços livres de inveja que maximiza a receita do vendedor.

*Demonstração.* Pelo Corolário 4.2.3, os números não negativos  $y_1, y_2, \dots, y_n$  são os mínimos satisfazendo (4.14). Então, pondo os preços como em (4.20), o Lema 4.1.4 assegura que eles maximizam as utilidades dos compradores. Além disso,

$$y_i \geq y_j + u_{ij} = y_j + v'_{ij} - v'_{jj} \Rightarrow y_j - y_i + v'_{ij} \leq v'_{jj}, \quad \forall i, j \in I, i \neq j.$$

Fixando  $i \in I$  tal que  $y_i = 0$ , cuja existência é garantida pelo Corolário 4.2.2, tem-se

$$y_j + v'_{ij} \leq v'_{jj} \Rightarrow y_j \leq v'_{jj}, \quad \forall j \in I.$$

Portanto, os itens alocados estão nos conjuntos de demanda dos compradores, daí o resultado segue.  $\square$

### 4.3 Algoritmo, corretude e complexidade de tempo

Agora estão dadas as condições para a apresentação de um algoritmo que calcula uma precificação livre de inveja que maximiza a receita do vendedor. O Algoritmo 1 descreve o procedimento.

---

**Algorithm 1** encontra uma precificação livre de inveja ótima

---

- 1:  $u_{ij} \leftarrow v'_{ij} - v'_{jj} \quad \forall i, j = 1, \dots, n$
- 2:  $\mathbf{y}^{(0)} \leftarrow (0, 0, \dots, 0) \in \mathbb{R}^n$
- 3:  $y_j^{(1)} \leftarrow \max_{k \in I} \{u_{jk}\} \quad \forall j = 1, \dots, n$
- 4:  $t \leftarrow 1$
- 5: **enquanto**  $y_j^{(t)} > y_j^{(t-1)}$  para algum  $j \in I$  **faça**
- 6:      $y_j^{(t+1)} \leftarrow \max_{k \in I} \{y_k^{(t)} + u_{jk}\} \quad \forall j = 1, \dots, n$
- 7:      $t \leftarrow t + 1$
- 8: **fim enquanto**
- 9:  $p_j \leftarrow v'_{jj} - y_j^{(t)} \quad \forall j = 1, \dots, n$
- 10: **return**  $\mathbf{p}$

Intuitivamente, pode-se entender o funcionamento do Algoritmo 1 da seguinte forma: a linha 1 constrói a "matriz de utilidades"  $\mathcal{U}$ , enquanto a sequência de utilidades é iniciada com zeros na linha 2, seguida das máximas utilidades na linha 3. A partir daí, nas linhas 4 – 8 a sequência de utilidades continua sendo construída até que se atinjam as utilidades máximas de acordo com o Lema 4.1.4, sendo que o loop termina em algum momento devido ao Teorema 4.2.1. Por fim, os preços ótimos livres de inveja que maximizam a receita são calculados na linha 9, de acordo com o Teorema 4.2.5. A seguir encontram-se as provas teóricas da corretude do algoritmo.

**Teorema 4.3.1.** (*invariante*) Para todo  $t \in \mathbb{N}$ , vale

$$y_j^{(t+1)} \geq y_k^{(t)} + u_{jk}, \quad \forall j, k \in I, j \neq k. \quad (4.21)$$

*Demonstração.* De fato, as linhas 2 – 3 definem

$$\begin{aligned} y_j^{(1)} &= \max_{l \in I} \{u_{jl}\} \\ &= \max_{l \in I} \{y_l^{(0)} + u_{jl}\} \\ &\geq y_k^{(0)} + u_{jk}, \quad \forall j, k \in I, j \neq k. \end{aligned}$$

Então, o invariante (4.21) vale antes da primeira iteração do laço descrito nas linhas 5 – 8. Além do mais, se o critério de parada na linha 5 não for satisfeito, então a linha 6 produz

$$\begin{aligned} y_j^{(t+1)} &= \max_{l \in I} \{y_l^{(t)} + u_{jl}\} \\ &\geq y_k^{(t)} + u_{jk}, \quad \forall j, k \in I, j \neq k, \end{aligned}$$

e portanto o invariante também vale no final de uma iteração arbitrária do laço, assim como no início, daí o resultado segue.  $\square$

**Teorema 4.3.2.** (*término*) O laço termina em no máximo  $n - 1$  iterações.

*Demonstração.* Pelo Teorema 4.2.1, vale

$$y_j^{((n-1)+1)} = y_j^{(n)} = y_j^{(n-1)}, \quad \forall j \in I.$$

Como  $t = 1$  no início do laço, o resultado segue.  $\square$

**Teorema 4.3.3.** (*corretude das utilidades*) Ao final do laço, vale

$$y_j^{(t)} \geq y_k^{(t)} + u_{jk}, \quad \forall j, k \in I, j \neq k, \quad (4.22)$$

para algum  $t \in \{1, \dots, n\}$ .

*Demonstração.* De fato, quando o critério de parada na linha 5 é satisfeito, então (4.22) vale porque

$$\begin{aligned} y_j^{(t)} = y_j^{(t-1)} &\Rightarrow y_j^{(t)} = \max_{l \in I} \{y_l^{(t-1)} + u_{jl}\} \\ &= \max_{l \in I} \{y_l^{(t)} + u_{jl}\} \\ &\geq y_k^{(t)} + u_{jk}, \quad \forall j, k \in I, j \neq k. \end{aligned}$$

$\square$

**Teorema 4.3.4.** (*corretude dos preços*) O Algoritmo 1 retorna um vetor de preços livres de inveja  $\mathbf{p}$  que maximiza a receita do vendedor.

*Demonstração.* A linha 9 produz um vetor  $\mathbf{p}$  que satisfaz (4.20). Portanto, pelo Teorema 4.2.5, o resultado segue.  $\square$

**Teorema 4.3.5.** O Algoritmo 1 encontra uma precificação livre de inveja ótima em tempo  $O(n^3)$ .

*Demonstração.* A maior custo computacional está no laço, pois nas linha anteriores realizam-se operações simples e atribuições de custo  $O(n)$ . A linha 6 executa  $n$  vezes  $O(n)$  operações, logo em tempo  $O(n^2)$ , e o laço efetua estas  $O(n^2)$  operações no máximo  $n - 1$  vezes. Portanto, o resultado segue.  $\square$

Observe o comportamento do vetor  $\mathbf{y}$  calculado pelo Algoritmo 1, utilizando como exemplo a mesma matriz  $\mathcal{U}$  da prova do Teorema 4.2.1:

$$\mathbf{y}^{(0)} = (0, 0, 0, 0, 0)$$

$$\mathbf{y}^{(1)} = (0, 6, 3, 5, 0)$$

$$\mathbf{y}^{(2)} = (0, 6, 9, 8, 0)$$

$$\mathbf{y}^{(3)} = (0, 6, 9, 14, 0)$$

Note que as componentes 1 e 5 atingem seu máximo antes da primeira iteração, a componente 2 atinge seu máximo na primeira iteração, a componente 3 atinge seu máximo na segunda iteração e a componente 4 atinge seu máximo na terceira iteração. Este exemplo mostra, juntamente com o Corolário 4.2.4, que pelo menos um consumidor atinge sua utilidade máxima a cada iteração, e daí o Algoritmo 1 explora uma subestrutura ótima do problema, e portanto possui uma abordagem em programação dinâmica.

Para fins de implementação, fez-se uma leve modificação no Algoritmo 1 para se aumentar a eficiência na busca das utilidades ótimas, sobrescrevendo-se o vetor  $\mathbf{y}$  nele mesmo a cada iteração do laço quando  $j = 1, \dots, n$ . Esta versão é descrita no seguinte algoritmo:

---

**Algorithm 2** versão melhorada do Algoritmo 1

---

- 1:  $u_{ij} \leftarrow v'_{ij} - v'_{jj} \quad \forall i, j = 1, \dots, n$
- 2:  $\mathbf{y} \leftarrow (0, 0, \dots, 0) \in \mathbb{R}^n$
- 3: **enquanto**  $y_j < \max_{k \in I} \{y_k + u_{jk}\}$  para algum  $j = 1, \dots, n$  **faça**
- 4:      $y_j \leftarrow \max_{k \in I} \{y_k + u_{jk}\} \quad \forall j = 1, \dots, n$
- 5: **fim enquanto**
- 6:  $p_j \leftarrow v'_{jj} - y_j \quad \forall j = 1, \dots, n$
- 7: **return**  $\mathbf{p}$

A principal diferença do Algoritmo 2 em relação ao Algoritmo 1 reside na atualização do "vetor de utilidades"  $\mathbf{y}$ , que é sobrescrito nele mesmo a cada iteração, e a cada vez que as linhas da matriz  $\mathcal{U}$  são utilizadas nos cálculos em cada iteração resultando, assim, num ganho de tempo e memória. Note que pode-se adaptar



facilmente as análises de corretude e complexidade de tempo do Algoritmo 1 para o Algoritmo 2.

## 4.4 Comparação com a estratégia de caminhos mínimos

Nesta seção serão apresentados resultados de experimentos para fins de comparação dos algoritmos propostos com a estratégia utilizada na literatura. Primeiramente, a seguir, serão expostos os materiais e métodos utilizados nas simulações.

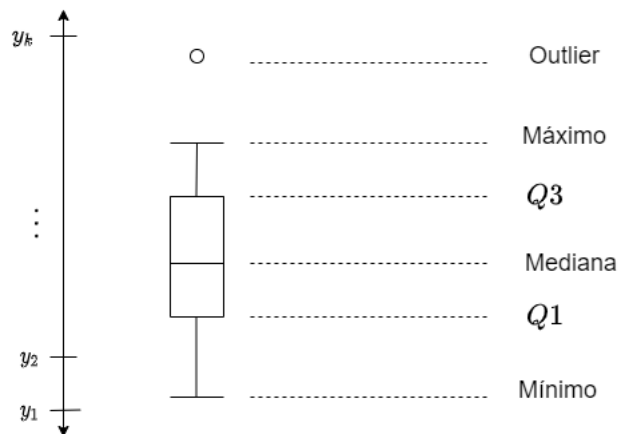
### 4.4.1 Materiais e métodos

Os algoritmos foram testados em conjuntos de dados gerados aleatoriamente através de um computador usando processador Intel Core i3-6006U rodando a 2.0 GHz e 4GB de RAM, usando Windows 10 Pro 64bits. Os códigos dos algoritmos foram implementados em C++ e compilados no g++ 8.1.0.

Foram gerados 15 casos de testes aleatórios para cada  $n = 1000, 2000, 5000, 10000$  e  $15000$ . Em todos os casos de teste, cada  $v_{ij}$  é um inteiro aleatório escolhido por distribuição uniforme sobre os inteiros entre 0 e 1000000, inclusive. A escolha de  $n$  a partir de 1000 se deu pelo fato de que para valores menores que este, os tempos (em segundos) são muito pequenos sendo, portanto, difíceis de comparar. Os códigos estão disponíveis no repositório do *GitHub* em <https://github.com/marcosmsgithub/efpm.git>.

Para efeito de comparação, os dados obtidos pelos experimentos foram analisados estatisticamente através do cálculo das médias aritméticas e dos intervalos de confiança de 95% (isto significa que, em tese, se forem realizados 100 experimentos para cada  $n$ , os tempos de execução de 95 deles estarão dentro do intervalo calculado para este  $n$ ) dos tempos de execução do algoritmo de Bellman-Ford e o Algoritmo 2.

Para uma visualização gráfica dos resultados dos experimentos, foram gerados *boxplots* [Tukey, 1972] para cada  $n$ . Um *boxplot* possibilita a observação visual de características como posição, dispersão, simetria, e valores discrepantes. Os elementos básicos de um *boxplot* são os valores mínimo e máximo, a mediana, os



**Figura 4.2.** Um *boxplot* e seus principais elementos

primeiro ( $Q1$ ) e terceiro ( $Q3$ ) quartis – valores que dividem a mostra em quatro partes iguais – e os *outliers* (valores discrepantes). Esses valores são determinados da seguinte forma:

- o **mínimo** é maior ou igual a  $Q1 - 1.5 * (Q3 - Q1)$ ;
- $Q1$  é maior ou igual a 25% e menor ou igual a 75% dos valores do conjunto de dados;
- a **mediana** corresponde ao valor central do conjunto de dados ordenado;
- $Q3$  é maior ou igual a 75% e menor ou igual a 25% dos valores do conjunto de dados;
- o **máximo** é menor ou igual a  $Q3 + 1.5 * (Q3 - Q1)$ ;
- os *outliers* são valores atípicos, ou discrepantes.

A Figura 4.2 ilustra um exemplo de *boxplot* e seus elementos básicos.

Outro recurso utilizado para se comparar as performances dos algoritmos propostos para o EFPM com os da literatura foi o *teste dos postos sinalizados de Wilcoxon* [Wilcoxon, 1992]. O uso deste teste tem como objetivo verificar se existem diferenças significativas entre os resultados. Se existirem apenas diferenças aleatórias a hipótese de nulidade ( $H0$ ) é aceita. Caso seja verificada uma

diferença de performance em favor de um dos algoritmos, a hipótese é rejeitada. Para determinar se a diferença entre a mediana dos resultados dos experimentos é estatisticamente significativa, o valor- $p$  – a probabilidade de se encontrar um resultado no qual a hipótese de nulidade seja aceita – é comparado com o nível de significância ( $\alpha$ ). Um nível de significância de 0.05 indica que o risco de se concluir que existe uma diferença, quando, na verdade, não existe nenhuma diferença real, é de 5%.

#### 4.4.2 Resultados dos experimentos

A Tabela 4.1 mostra as médias dos tempos de execução e os intervalos de confiança (IC), em segundos com precisão de dois dígitos, do algoritmo de Bellman-Ford e do Algoritmo 2. Conforme o esperado, o Algoritmo 2 obteve melhor performance que o algoritmo de Bellman-Ford em todas as simulações, com um aumento de performance na determinação dos preços livre de inveja ótimos de, em média, 49%. Além do mais, nota-se que as amplitudes dos intervalos de confiança de 95% do Algoritmo 2 são menores que do algoritmo de Bellman-Ford, mostrando que o Algoritmo 2 apresentou-se mais "estável" nos experimentos realizados, fornecendo uma melhor precisão estatística nas análises.

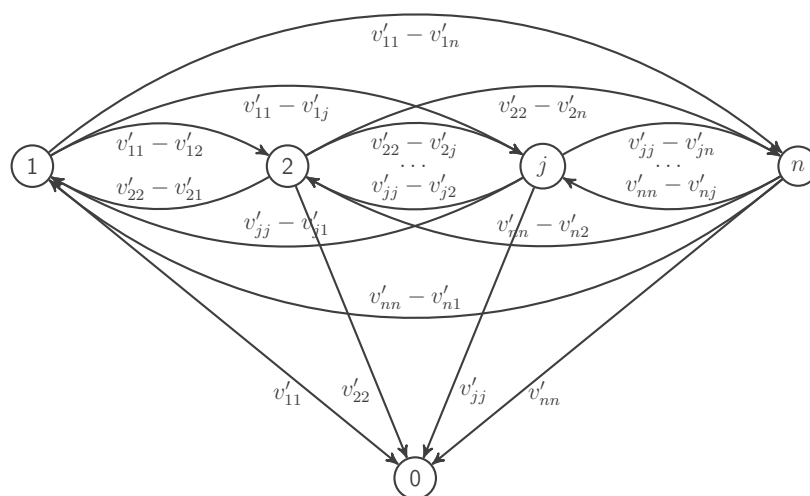
Isto se deve ao seguinte fato: a estratégia usada por Arbib, Karaşan e Pınar [Arbib et al., 2019] para encontrar os preços livres de inveja ótimos consiste em construir um grafo direcionado com  $n + 1$  nós no qual o nó 0 é um nó auxiliar, cada nó  $j$  representa um item  $j \in I$ , cada arco conectando os nós  $j$  e  $k$  para todos  $j, k \in I$ ,  $j \neq k$  possui comprimento  $v'_{jj} - v'_{jk}$ , e cada arco conectando cada nó  $j \in I$  ao nó 0 possui comprimento  $v'_{jj}$ . Então, o preço livre de inveja ótimo do item  $j$  é o comprimento do menor caminho do nó  $j$  ao nó 0. Eles enfatizaram o uso do algoritmo de Bellman-Ford para isto (pois o grafo pode conter arcos com pesos negativos). Este esquema é mostrado na Fig. 4.3.

Sendo assim, o algoritmo de Bellman-Ford realiza  $n \times O(n^2)$  operações para encontrar os caminhos mínimos ao nó 0 no caso médio, enquanto o Algoritmo 2 realiza  $(n - 1) \times O(n^2)$  operações para encontrar as utilidades máximas no pior caso, e então calcula os preços livres de inveja ótimos em tempo  $O(n)$ .

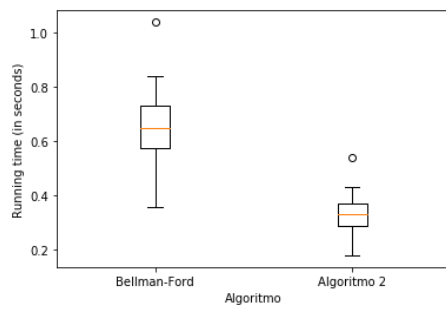
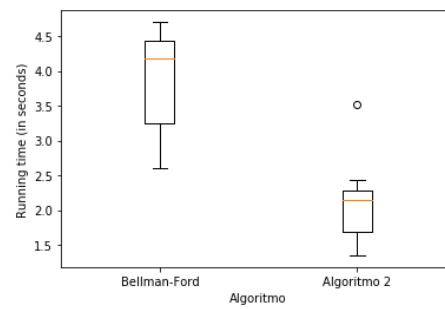
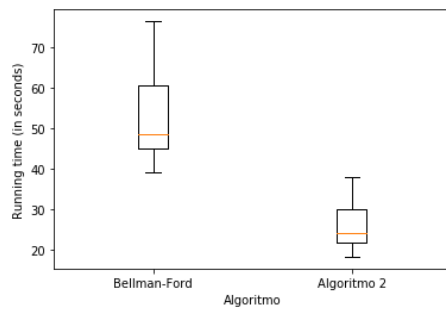
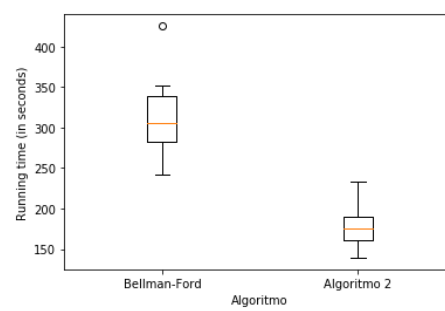
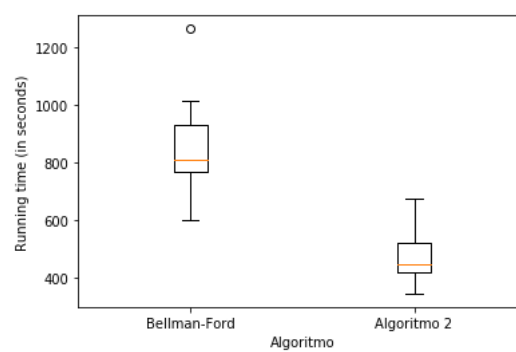
As Figuras 4.4, 4.5 e 4.6 mostram comparações gráficas dos tempos de exe-

**Tabela 4.1.** Comparação dos tempos de execução dos algoritmos de Bellman-Ford e o Algoritmo 2

n	Bellman-Ford		Algoritmo 2	
	Tempo médio (s)	IC de 95%	Tempo médio	IC de 95%
1000	0.69	(0.56,0.74)	0.33	(0.28,0.38)
2000	3.89	(3.49,4.30)	2.09	(1.80,2.39)
5000	53.82	(47.16,60.48)	26.07	(22.75,29.39)
10000	312.16	(286.39,337.94)	175.39	(161.80,188.81)
15000	864.60	(777.08,952.11)	477.21	(431.19,523.22)

**Figura 4.3.** Grafo usado por Arbib, Karaşan e Pınar [Arbib et al., 2019] para encontrar os preços ótimos livres de inveja.

cução dos algoritmos para as instâncias usadas nas simulações através de *boxplots* e, para se assegurar que os resultados dos experimentos indicam de fato uma diferença na performance dos algoritmos, o teste dos postos sinalizados de Wilcoxon foi feito para cada  $n$  utilizado nos experimentos. Em todos os testes, a hipótese de nulidade ( $H_0$ ) foi rejeitada, ou seja, a hipótese de que a abordagem em programação dinâmica proposta neste trabalho não representa uma diminuição significativa nos tempos de execução na busca de soluções ótimas em relação à abordagem de caminhos mínimos mostrou-se falsa. Os resultados dos testes são mostrados na Tabela 4.2.

(a) Experimentos para  $n = 1000$ (b) Experimentos para  $n = 2000$ **Figura 4.4.** *Boxplots* dos tempos de execução para  $n = 1000$  e  $n = 2000$ (a) Experimentos para  $n = 5000$ (b) Experimentos para  $n = 10000$ **Figura 4.5.** *Boxplots* dos tempos de execução para  $n = 5000$  e  $n = 10000$ **Figura 4.6.** *Boxplot* dos tempos de execução para  $n = 15000$

**Tabela 4.2.** Testes dos postos sinalizados de Wilcoxon com nível de significância  $\alpha = 0.05$ 

n	valor- $p$	Rejeição de $H_0$
1000	0.00064675	SIM
2000	0.00065331	SIM
5000	0.00065496	SIM
10000	0.00065496	SIM
15000	0.00065496	SIM

## 4.5 Considerações finais

Neste capítulo foram apresentadas estratégias algorítmicas para o EFPM demonstradas matematicamente e que, empiricamente, obtiveram melhor desempenho de tempo na busca de soluções ótimas se comparadas àquelas encontradas na literatura.

Para este caso, o foco foi a obtenção dos preços livres de inveja, pois a alocação pode ser obtida através de um emparelhamento perfeito de peso máximo utilizando-se algoritmos clássicos de tempo cúbico que usam uma heap de Fibonacci [Fredman & Tarjan, 1987].

Encontrada uma alocação ótima, foram construídas sequências que representam as utilidades dos compradores. Foi feita a demonstração matemática de que estas sequências convergem para as utilidades ótimas que maximizam a receita do vendedor, daí os algoritmos propostos utilizaram este fato para calcular os preços ótimos livres de inveja.

Feitas as análises estatísticas dos resultados dos experimentos, observou-se que a estratégia proposta nesta tese obteve uma melhoria no desempenho de tempo de, em média, 49% na busca de soluções ótimas para o problema se comparada à estratégia de caminhos mínimos.

# O caso de demanda unitária com substituibilidade métrica

---

Neste capítulo serão apresentados resultados teóricos sobre o aumento de performance na busca de soluções ótimas para o caso de demanda unitária com substituibilidade métrica [Salvatierra et al., 2021].

No *problema da precificação livre de inveja com demanda unitária (UDEFP – unit demand envy-free pricing, em inglês) e substituibilidade métrica*, um vendedor possui um item com disponibilidade ilimitada de cópias, e  $n$  consumidores estão em localidades diferentes. Cada consumidor  $i$  atribui uma valoração  $v_i$  ao item, e um *custo de deslocamento*  $\delta_{ij}$  que é subtraído de  $v_i$  caso a compra seja realizada na localidade  $j$ . O conjunto de todos os custos de deslocamento  $\{\delta_{ij}\}_{1 \leq i, j \leq n}$  possui a estrutura de *espaço métrico*, isto é, a função  $\delta : \{1, \dots, n\}^2 \rightarrow \mathbb{R}$  satisfaz, para todos  $1 \leq i, j, k \leq n$ :

- (i)  $\delta_{ij} = 0 \iff i = j$  (identidade de indiscerníveis);
- (ii)  $\delta_{ij} = \delta_{ji}$  (simetria);
- (iii)  $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$  (desigualdade triangular).

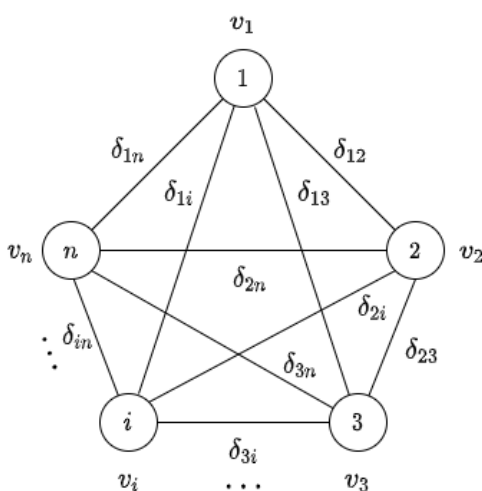
O vendedor precisa determinar o preço  $p_i$  da cópia do item vendido na localidade do consumidor  $i$  de modo a maximizar a utilidade  $v_i - (p_i + \delta_i)$  de cada comprador  $i$  e, ao mesmo tempo, maximizar a receita com a venda das cópias do item.

## 5.1 Equilíbrio competitivo

Num primeiro momento, será considerado que todos os consumidores compram uma cópia do item. Dessa forma, a alocação e a precificação são determinadas de tal maneira que existe um equilíbrio competitivo de mercado, e as soluções são Pareto Eficientes [Nisan et al., 2007].

### 5.1.1 Redução do UDEFM com substituíbilidade métrica ao EFPM

Uma instância do UDEFM com substituíbilidade métrica consiste em:  $n$ , o número de consumidores;  $v_i$ , a valoração que o consumidor  $i$  atribui ao item se a compra é realizada na sua própria localidade;  $\delta_{ij}$ , o custo de deslocamento que é subtraído de  $v_i$  se a compra do consumidor  $i$  é realizada na localidade  $j$ . As instâncias para este problema podem ser modeladas por grafos não direcionados com pesos nos vértices e nas arestas, como mostra a Figura 5.1.



**Figura 5.1.** Uma instância do UDEFM com substituíbilidade métrica.

Já no EFPM, uma instância consiste de  $n$ , o número de consumidores e itens, e de  $V = (v_{ij})$ , a matriz de valorações. As instâncias para este problema podem ser modeladas por grafos bipartidos  $G = G(I \cup J, E)$  com pesos nas arestas, em que  $I$  representa o conjunto de consumidores,  $J$  representa o conjunto de itens, e



o peso de uma aresta conectando o vértice  $i$  ao vértice  $j$  representa a valoração que o consumidor  $i$  atribui ao item  $j$ , como mostrado na Figura 5.2.

Então, pondo  $v_{ij} = v_i - \delta_{ij}$  para todos  $1 \leq i, j \leq n$ , é fácil ver que, do ponto de vista abstrato, o UDEFPP com substituíbilidade métrica é um caso particular do EFPM. Sob esta abordagem, as linhas de  $V$  representam os consumidores, e as colunas representam as localidades.

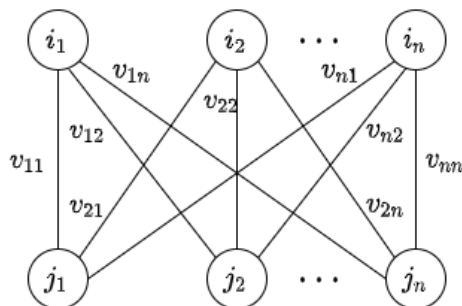


Figura 5.2. Uma instância do EFPM.

## 5.1.2 Alocação

Pelo Primeiro Teorema do Bem-Estar Social [Nisan et al., 2007], sabe-se que uma alocação ótima para o EFPM maximiza o *bem-estar social*, ou seja, a soma das valorações dos compradores para os itens alocados a eles. Em termos de programação matemática linear, o problema de encontrar tal alocação pode ser modelado pelo MIP (4.1) – (4.7).

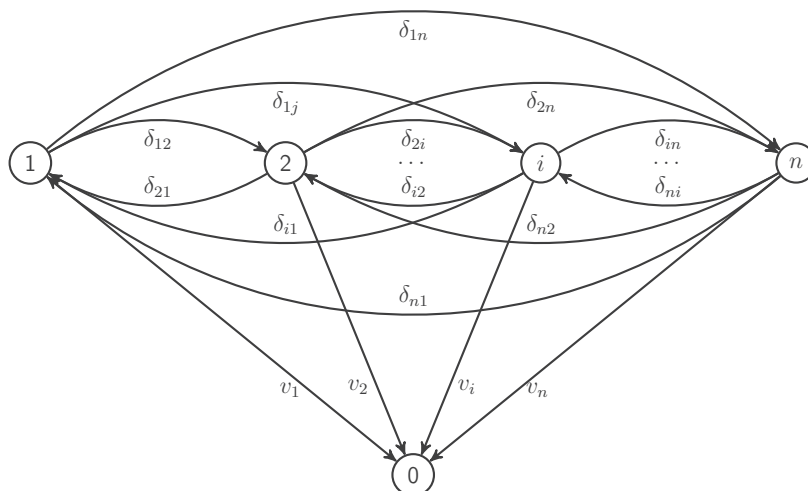
Este programa linear também modela o emparelhamento perfeito de peso máximo no grafo mostrado na Figura 5.2. Como  $v_{ii} = v_i > v_i - \delta_{ij} = v_{ij}$  para todos  $1 \leq i, j \leq n$  com  $i \neq j$ , tem-se que a matriz identidade  $n \times n$  é aquela que maximiza  $tr(\Pi^T V)$ , ou seja, a soma dos elementos da diagonal principal de  $\Pi^T V$ , em que  $\Pi$  é uma matriz de permutação  $n \times n$ . Portanto, a solução para este MIP é  $x_{ij} = 1$  se  $i = j$ , e  $x_{ij} = 0$ , caso contrário, em que  $x_{ij}$  são as variáveis binárias.

Estas considerações acerca da alocação ótima permitem que se estabeleça a seguinte proposição:

**Proposição 5.1.1.** *Cada consumidor compra uma cópia do item em sua própria localidade.*

### 5.1.3 Precificação

Uma vez que uma alocação ótima é encontrada, considere a seguinte rede: o conjunto de nós consiste dos nós  $0, 1, \dots, n$ ; existe um arco de cada nó  $1 \leq i \leq n$  ao nó  $0$  com peso  $v_{ii} = v_i$ ; existe um arco do nó  $i$  ao nó  $j$ , para todos  $1 \leq i, j \leq n$  tais que  $i \neq j$ , com peso  $v_{ii} - v_{ij} = \delta_{ij}$ . Então, os caminhos mínimos dos nós  $1 \leq i \leq n$  ao nó  $0$  correspondem aos preços livres de inveja  $p_1, \dots, p_n$  [Arbib et al., 2019]. A rede é mostrada na Figure 5.3.



**Figura 5.3.** A rede construída para encontrar os preços livres de inveja.

Ao invés de se usar o algoritmo de Bellman-Ford para encontrar os caminhos mínimos ao nó  $0$ , a estrutura métrica do conjunto dos custos de deslocamento será explorada para simplificar os cálculos mas, antes disso, será visto que é possível encontrar um preço particular sem haver a necessidade de se calcular o caminho mínimo correspondente.

**Lema 5.1.2.** *Se  $v_1 \leq v_2 \leq \dots \leq v_n$ , então  $p_1 = v_1$ .*

*Demonstração.* De fato, se  $p_1 > v_1$ , a condição de liberdade de inveja, ou seja, de maximização da utilidade não-negativa é violada. Se  $p_1 < v_1$ , os outros preços serão forçados a baixar até as utilidades dos outros compradores alcançarem no mínimo  $v_{i1} - p_1$ , para todo  $2 \leq i \leq n$ , e então os preços não maximizarão a receita, de onde o resultado segue. □

Agora estão dadas as condições para se encontrar iterativamente os preços restantes a partir desta primeira descoberta.

**Lema 5.1.3.** *Se  $v_1 \leq v_2 \leq \dots \leq v_n$ , então  $p_i = \min_{1 \leq j < i} \{v_i, v_j + \delta_{ij}\}$  para todo  $2 \leq i \leq n$ .*

*Demonstração.* Da estrutura métrica do conjunto dos custos de deslocamento, tem-se que  $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$  para todos  $1 \leq i, j, k \leq n$ . Então, o caminho mínimo do nó  $i$  ao nó 0 consiste ou do arco conectando o nó  $i$  ao nó 0 com peso  $v_i$ , ou dos arcos conectando o nó  $i$  a algum nó  $j$  e o nó  $j$  ao nó 0 com pesos  $\delta_{ij}$  e  $v_j$ , respectivamente. Como  $v_i \leq v_k$  para todos  $i \leq k$ , tem-se que  $v_i \leq v_k + \delta_{ik}$  para todos  $i \leq k$ , e então  $\min_{1 \leq j < i} \{v_i, v_j + \delta_{ij}\} \leq v_k + \delta_{ik}$  para todo  $2 \leq i \leq n$  com  $i \leq k$ , de onde o resultado segue.  $\square$

### 5.1.4 Algoritmo, corretude e complexidade de tempo

As considerações feitas acima permitem que se apresente um algoritmo para encontrar uma solução ótima para o UDEFP com substituibilidade métrica e equilíbrio competitivo. O procedimento é descrito abaixo.

---

**Algorithm 3** encontra um equilíbrio competitivo ótimo

---

- 1: Faça uma permutação  $\pi$  tal que  $v_{\pi^{-1}(1)} \leq v_{\pi^{-1}(2)} \leq \dots \leq v_{\pi^{-1}(n)}$
- 2:  $p_{\pi^{-1}(1)} \leftarrow v_{\pi^{-1}(1)}$
- 3:  $p_{\pi^{-1}(i)} \leftarrow \min_{1 \leq j < i} \{v_{\pi^{-1}(i)}, v_{\pi^{-1}(j)} + \delta_{\pi^{-1}(i)\pi^{-1}(j)}\}$  para todo  $2 \leq i \leq n$

Este algoritmo permite que se estabeleça o seguinte teorema:

**Teorema 5.1.4.** *O UDEFP com substituibilidade métrica e equilíbrio competitivo pode ser resolvido de maneira ótima em tempo  $O(n^2)$ .*

*Demonstração.* O Lema 5.1.2 assegura que a Linha 2 retorna um preço livre de inveja ótimo para o consumidor  $\pi^{-1}(1)$ , e o Lema 5.1.3 assegura que a Linha 3 retorna preços livres de inveja ótimos para os outros. A permutação na Linha 1 pode ser feita em tempo  $O(n^2)$  por qualquer algoritmo de ordenação clássico, e a Linha 3 realiza  $n - 1$  vezes  $O(n)$  operações. Portanto, a complexidade de tempo total do algoritmo é  $O(n^2)$ .  $\square$

## 5.2 Cenário geral

Em alguns casos, pode haver um aumento da receita se o vendedor não vender cópias do item em algumas localidades. Como um exemplo, considere dois consumidores com  $v_1 = 10$ ,  $v_2 = 4$ , e  $\delta_{12} = \delta_{21} = 1$ . Então  $\mathbf{p} = (5, 4)$  é a precificação livre de inveja ótima com equilíbrio competitivo com receita 9, mas  $\mathbf{p}' = (10, 10)$  é uma precificação livre de inveja com receita 10, em que o consumidor 2 não compra uma cópia do item.

É claro que a Proposição 5.1.1 também vale para o cenário geral (pois valorações maiores produzem preços maiores), com a seguinte modificação:

**Proposição 5.2.1.** *Cada comprador realiza a compra na sua própria localidade.*

Ao invés de se usar força bruta para verificar todas as combinações de subalocações da alocação definida na Seção 5.1.2, é fácil ver que é necessário verificar apenas as alocações para consumidores com valorações mais altas. Então, se  $v_1 \geq v_2 \geq \dots \geq v_n$ , pode-se calcular os preços livres de inveja e as receitas a partir das submatrizes principais da matriz de valorações  $V$ , e então escolhe-se a alocação e a precificação que produzem a maior receita. O procedimento é descrito no seguinte algoritmo:

---

**Algorithm 4** encontra uma precificação livre de inveja ótima

---

- 1: Faça uma permutação  $\sigma$  tal que  $v_{\sigma^{-1}(1)} \geq v_{\sigma^{-1}(2)} \geq \dots \geq v_{\sigma^{-1}(n)}$
- 2: Use o Algoritmo 3 para calcular os preços na matriz  $k \times k$  dos consumidores  $\sigma^{-1}(1), \dots, \sigma^{-1}(k)$  para todo  $1 \leq k \leq n$ . Se  $k < n$ ,  $p_i \leftarrow v_{\sigma^{-1}(1)}$  Para todo  $k < i \leq n$
- 3: Escolha a alocação e a precificação que produzem a receita máxima

Este segundo algoritmo permite que se estabeleça o seguinte teorema:

**Teorema 5.2.2.** *O UDEFP com substituíbilidade métrica pode ser resolvido de maneira ótima em tempo  $O(n^3)$ .*

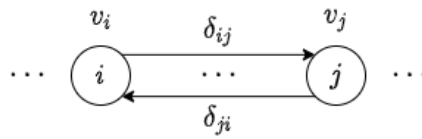
*Demonstração.* O maior custo computacional está na Linha 2, pois nela realiza-se uma chamada do Algoritmo 4. Esta linha realiza  $n$  vezes  $O(n^2)$  operações, pelo Teorema 5.1.4, de onde o resultado segue.  $\square$

## 5.3 Comparação com a estratégia de conjuntos independentes de peso máximo

Os resultados apresentados neste trabalho sobre as propriedades das alocações e precificações livres de inveja ótimas estão de acordo com aqueles apresentados por Chen, Gosh e Vassilvitskii [Chen et al., 2011], que utilizaram a estratégia de conjuntos independentes de peso máximo: cada consumidor que compra uma cópia do item realiza a compra na sua própria localidade, e o preço pago por um consumidor é igual à sua valoração ou ao preço pago por outro consumidor mais o custo de deslocamento para a localidade deste último consumidor, ou seja, para cada comprador  $1 \leq i \leq n$  tem-se  $p_i = v_i$  ou  $p_i = p_j + \delta_{ij}$  para algum  $j \neq i$  (mais especificamente,  $j < i$ ), com  $p_j = v_j$ .

A fim de se justificar esta última afirmação, suponha que exista um  $i$  tal que  $p_i = v_j + \delta_{ij}$  para algum  $j < i$  com  $v_j \neq p_j$ . Isto implica que  $p_j < v_j$ , e então  $p_j = v_k + \delta_{jk}$  para algum  $k < j$ . Como  $\delta_{ik} \leq \delta_{ij} + \delta_{jk}$ , ocorreria que  $v_k + \delta_{ik} \leq v_k + \delta_{ij} + \delta_{jk} = p_j + \delta_{ij} < v_j + \delta_{ij}$ , e então  $p_i \neq \min_{1 \leq j' < i} \{v_i, v_{j'} + \delta_{ij'}\}$ , uma contradição.

A abordagem do UDEFP com substituíbilidade métrica em termos do EFPM permitiu, além de apresentar provas teóricas alternativas de algumas propriedades, também reduzir a complexidade de tempo na busca de soluções ótimas para o problema. Além do mais, nota-se que a propriedade da simetria no conjunto dos custos de deslocamento não foi exigida pra se encontrar uma precificação livre de inveja ótima. Então, pode-se assumir que  $\delta_{ij}$  não precisa ser necessariamente igual a  $\delta_{ji}$  (mas positivo) quando  $i \neq j$ , o que incorpora mais subjetividade à modelagem do problema, tornando-a mais realística. Dessa forma, as instâncias do UDEFP com substituíbilidade métrica tornam-se redes com pesos nos nós e nos arcos (ver Figura 5.4).



**Figura 5.4.** Uma instância do UDEFM com substituíbilidade métrica e custos de deslocamento não simétricos.

## 5.4 Considerações finais

Neste capítulo foram apresentadas propostas algorítmicas teóricas matematicamente demonstradas para o UDEFM com substituíbilidade métrica.

Para o desenvolvimento dos algoritmos, primeiramente uma instância genérica do UDEFM com substituíbilidade métrica foi transformada em uma instância genérica do EFPM. As propriedades da diagonal principal da matriz de avaliações, sob esta nova perspectiva, foram usadas para a determinação da alocação. Uma rede cujos comprimentos dos arcos são baseados em relações específicas dos elementos da matriz de avaliações foi construída e, como os caminhos mínimos a um nó auxiliar correspondem aos preços livres de injeção que maximizam a receita, a desigualdade triangular foi usada para simplificar os cálculos destes caminhos mínimos através de programação dinâmica, ao invés de se recorrer ao algoritmo de Bellman-Ford.

Como resultado, obteve-se uma redução teórica da complexidade de tempo de  $O(n^4)$  para  $O(n^3)$  na busca de soluções ótimas para o problema, melhorando, assim, o desempenho na busca dessas soluções, comparado-se com estratégia utilizada na literatura, que reduziu o problema ao de conjuntos independentes de peso máximo.

## Conclusões

---

### Contribuições da tese e resultados

Neste trabalho foram apresentadas duas propostas algorítmicas mais eficientes que as encontradas na literatura para casos especiais ótimos do problema da precificação livre de inveja: o caso de emparelhamento perfeito livre de inveja e o caso de demanda unitária com substituibilidade métrica. Tanto os resultados obtidos quanto os processos envolvidos na obtenção destes dão a sua contribuição para a teoria da ciência da computação e, mais especificamente, para a área de otimização combinatória nos seguintes aspectos: por um lado, aumenta-se o repertório de estratégias possíveis – matematicamente demonstradas – na busca de soluções ótimas eficientes para casos particulares de problemas computacionais mais gerais, como é o caso do estudo feito sobre o emparelhamento perfeito livre de inveja; por outro lado, reduz-se o limite superior assintótico na busca de soluções ótimas para um determinado problema computacional, e a redução de tal problema a outro conhecido contribui para a consolidação e a unificação da teoria, como é o caso do estudo feito sobre demanda unitária com substituibilidade métrica, no qual reduziu-se a complexidade de tempo de  $O(n^4)$  para  $O(n^3)$ , resultado este aceito e publicado no periódico *Algorithms* em setembro de 2021 [Salvatierra et al., 2021], enquanto aquele outro resultado – sobre o emparelhamento perfeito livre de inveja – foi submetido para publicação no periódico *Algorithmica*, e um *survey* baseado na

revisão de literatura apresentada no Capítulo 3 está em fase desenvolvimento para submissão em periódico especializado na área de otimização/pesquisa operacional.

## Limitações/Dificuldades encontradas

Vale ressaltar que a estratégia proposta para a busca de preços ótimos no caso de emparelhamento perfeito livre de inveja aplica-se somente quando é fornecido um emparelhamento perfeito de peso máximo exato como entrada. Se o emparelhamento é uma aproximação do ótimo, o algoritmo entrará num *loop* infinito, pois o procedimento violará a condição estrutural da matriz  $\mathcal{U}$  apresentada no Corolário 4.1.3.

Um dificuldade encontrada durante a pesquisa foi a implementação do algoritmo proposto na literatura para o caso de demanda unitária com substituíbilidade métrica devido à sofisticação teórica utilizada da modelagem do problema e à complexidade dos grafos envolvidos, o que tornou inviável a comparação empírica entre as performances este algoritmo e o proposto neste trabalho.

## Trabalhos futuros

As limitações e dificuldades descritas acima encorajam o desenvolvimento de algoritmos aproximativos eficientes para o cálculo dos preços livres de inveja quando as instâncias tiverem grande magnitude. Como algoritmos aproximativos de tempo linear com razão de aproximação  $1 - \epsilon$ , para qualquer  $\epsilon > 0$ , têm sido apresentados para a determinação de emparelhamentos de peso máximo – tal como o proposto por Duan e Pettie [Duan & Pettie, 2014] – uma possibilidade de trabalho futuro é o desenvolvimento de um algoritmo mantendo estas mesmas razão de aproximação (com um fator possivelmente dependendo de  $\epsilon$ ) e complexidade de tempo para a obtenção de preços livres de inveja.

Ainda para o caso de emparelhamento perfeito livre de inveja, pode-se trabalhar na identificação de classes de instâncias que indiquem o melhor e pior casos do algoritmo proposto.



Um desafio que também permanece é a implementação e análise da performance empírica tanto dos algoritmos propostos neste trabalho quanto do encontrado na literatura para o caso de demanda unitária com substituíbilidade métrica.

Outra possibilidade de investigação futura consiste em verificar se, reforçando-se a desigualdade triangular e transformando-se os custos de deslocamento em um espaço ultramétrico – ou seja, ao invés de se ter  $\delta_{ij} \leq \delta_{ik} + \delta_{kj}$  para quaisquer  $i, j, k$  tem-se  $\delta_{ij} \leq \max\{\delta_{ik}, \delta_{kj}\}$  – o caso de demanda unitária com substituíbilidade (ultra)métrica pode ser resolvido de maneira ótima num tempo menor que  $O(n^2)$  no cenário de equilíbrio competitivo, e num tempo menor que  $O(n^3)$  em geral.



# Referências Bibliográficas

---

- Alotaibi, A.; Almutairi, A. & Kurdi, H. (2020). Onebyone (obo): A fast sorting algorithm. *Procedia Computer Science*, 175:270--277.
- Anshelevich, E.; Kar, K. & Sekar, S. (2017). Envy-free pricing in large markets: Approximating revenue and welfare. *ACM Trans. Econ. Comput.*, 5(3):16:1--16:42.
- Arbib, C.; Karaşan, O. & Pinar, M. (2019). On envy-free perfect matching. *Discrete Applied Mathematics*, 261:22 – 27. GO X Meeting, Rigi Kaltbad (CH), July 10–14, 2016.
- Bellman, R. (1958). On a routing problem. *Quarterly of applied mathematics*, 16(1):87--90.
- Bilò, V.; Flammini, M. & Monaco, G. (2017). Approximating the revenue maximization problem with sharp demands. *Theor. Comput. Sci.*, 662:9–30.
- Bondy, J. A. & Murty, U. S. R. (2008). Graph theory.
- Briest, P. (2008). Uniform budgets and the envy-free pricing problem. Em Aceto, L.; Damgård, I.; Goldberg, L. A.; Halldórsson, M. M.; Ingólfssdóttir, A. & Walukiewicz, I., editores, *Automata, Languages and Programming*, pp. 808--819, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Briest, P. & Krysta, P. (2011). Buying cheap is expensive: Approximability of combinatorial pricing problems. *SIAM Journal on Computing*, 40(6):1554--1586.
- Burkard, R. E. & Yao, E.-Y. (1990). Constrained partitioning problems. *Discrete applied mathematics*, 28(1):21--34.

- Chalermsook, P.; Chuzhoy, J.; Kannan, S. & Khanna, S. (2012). Improved hardness results for profit maximization pricing problems with unlimited supply. Em Gupta, A.; Jansen, K.; Rolim, J. & Servedio, R., editores, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pp. 73--84, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Chen, G.-H.; Kuo, M. & Sheu, J. (1988). An optimal time algorithm for finding a maximum weight independent set in a tree. *BIT Numerical Mathematics*, 28(2):353--356.
- Chen, N. & Deng, X. (2014). Envy-free pricing in multi-item markets. *ACM Transactions on Algorithms (TALG)*, 10(2):1--15.
- Chen, N.; Deng, X.; Goldberg, P. W. & Zhang, J. (2016). On revenue maximization with sharp multi-unit demands. *Journal of Combinatorial Optimization*, 31(3):1174--1205.
- Chen, N.; Ghosh, A. & Vassilvitskii, S. (2011). Optimal envy-free pricing with metric substitutability. *SIAM Journal on Computing*, 40(3):623--645.
- Coppersmith, D. & Winograd, S. (1987). Matrix multiplication via arithmetic progressions. Em *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pp. 1--6.
- Cormen, T. H.; Leiserson, C. E.; Rivest, R. L. & Stein, C. (2009). *Introduction to algorithms*. MIT press.
- Dasgupta, S.; Papadimitriou, C. H. & Vazirani, U. V. (2008). *Algorithms*. McGraw-Hill Higher Education New York.
- Day, J. M. & Venkataramanan, M. (2006). Profitability in product line pricing and composition with manufacturing commonalities. *European journal of operational research*, 175(3):1782--1797.
- Demaine, E. D.; Feige, U.; Hajiaghayi, M. & Salavatipour, M. R. (2008). Combination can be hard: Approximability of the unique coverage problem. *SIAM Journal on Computing*, 38(4):1464--1483.

- Demaine, E. D.; Hajiaghayi, M. T.; Feige, U. & Salavatipour, M. R. (2006). Combination can be hard: approximability of the unique coverage problem, soda'06: Proceedings of the seventeenth annual acm-siam symposium on discrete algorithm.
- Dijkstra, E. W. et al. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269--271.
- Dobson, G. & Kalish, S. (1988). Positioning and pricing a product line. *Marketing Science*, 7(2):107--125.
- Dobson, G. & Kalish, S. (1993). Heuristics for pricing and positioning a product-line using conjoint and cost data. *Management Science*, 39(2):160--175.
- Duan, R. & Pettie, S. (2014). Linear-time approximation for maximum weight matching. *Journal of the ACM (JACM)*, 61(1):1--23.
- Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449--467.
- Fernandes, C. G.; Ferreira, C. E.; Franco, A. J. & Schouery, R. C. (2016). The envy-free pricing problem, unit-demand markets and connections with the network pricing problem. *Discrete Optimization*, 22:141--161.
- Floudas, C. A. (1995). *Nonlinear and mixed-integer optimization: fundamentals and applications*. Oxford University Press.
- Floyd, R. W. (1962). Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345.
- Foley, D. K. (1967). Resource allocation and the public sector. *Yale Economic Essays*, 7(1):45--98.
- Fredman, M. L. & Tarjan, R. E. (1987). Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM (JACM)*, 34(3):596--615.
- Gamow, G. (1958). *Puzzle-Math. By G. Gamow & M. Stern*. Viking, New York.

- Gibbons, A. (1985). *Algorithmic graph theory*. Cambridge university press, Cambridge.
- Godsil, C. & Royle, G. F. (2001). *Algebraic graph theory*, volume 207. Springer Science & Business Media, Berlin.
- Günlük, O. (2008). A pricing problem under monge property. *Discrete Optimization*, 5(2):328--336.
- Guruswami, V.; Hartline, J. D.; Karlin, A. R.; Kempe, D.; Kenyon, C. & McSherry, F. (2005). On profit-maximizing envy-free pricing. Em *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '05, pp. 1164--1173, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.
- Held, M. & Karp, R. M. (1962). A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196--210.
- Karlin, A. R.; Klein, N. & Gharan, S. O. (2021). A (slightly) improved approximation algorithm for metric tsp. Em *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pp. 32--45.
- Karp, R. M. (1972). Reducibility among combinatorial problems. Em *Complexity of computer computations*, pp. 85--103. Springer.
- Keil, J. M.; Mitchell, J. S.; Pradhan, D. & Vatschelle, M. (2017). An algorithm for the maximum weight independent set problem on outerstring graphs. *Computational Geometry*, 60:19--25.
- Khot, S. (2006). Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. *SIAM Journal on Computing*, 36(4):1025--1071.
- Kleinberg, J. & Tardos, E. (2006). *Algorithm design*. Pearson Education India.
- Kuhn, H. W. (1955). The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83--97.

- Lozin, V. V. & Milanič, M. (2008). A polynomial algorithm to find an independent set of maximum weight in a fork-free graph. *Journal of Discrete Algorithms*, 6(4):595--604.
- Magnanti, T.; Ahuja, R. & Orlin, J. (1993). Network flows: theory, algorithms, and applications. *PrenticeHall, Upper Saddle River, NJ*.
- Maskin, E. S. (1987). *Arrow and the Foundations of the Theory of Economic Policy*, chapter On the Fair Allocation of Indivisible Goods, pp. 341--349. Palgrave Macmillan UK, London.
- Myklebust, T. G. J.; Sharpe, M. A. & Tunçel, L. (2016). Efficient heuristic algorithms for maximum utility product pricing problems. *Computers & OR*, 69:25--39.
- Nisan, N.; Roughgarden, T.; Tardos, E. & Vazirani, V. V. (2007). *Algorithmic Game Theory*. Cambridge University Press, New York, NY, USA.
- Parker, R. G. & Rardin, R. L. (2014). *Discrete optimization*. Elsevier, Amsterdam.
- Prim, R. C. (1957). Shortest connection networks and some generalizations. *The Bell System Technical Journal*, 36(6):1389--1401.
- Saari, D. (1988). Fair Allocation (H. Peyton Young). *SIAM Review*, 30(2):350--352.
- Salvatierra, M. M.; Salvatierra, M. & Colonna, J. G. (2021). Short communication: Optimally solving the unit-demand envy-free pricing problem with metric substitutability in cubic time. *Algorithms*, 14(10).
- Schouery, R. C. S.; Lee, O.; Miyazawa, F. K. & Xavier, E. C. (2015). *Tópicos da teoria dos jogos em computação*. IMPA, Rio de Janeiro.
- Schrijver, A. (1998). *Theory of linear and integer programming*. John Wiley & Sons, Hoboken.
- Shioda, R.; Tunçel, L. & Myklebust, T. G. J. (2011). Maximum utility product pricing models and algorithms based on reservation price. *Comp. Opt. and Appl.*, 48(2):157--198.

- Steinhaus, H. (1949). Sur la division pragmatique. *Econometrica: Journal of the Econometric Society*, pp. 315--319.
- Strassen, V. (1969). Gaussian elimination is not optimal. *Numerische mathematik*, 13(4):354--356.
- Szwarcfiter, J. L. (2018). *Teoria computacional de grafos: Os Algoritmos*. Elsevier Brasil, Rio de Janeiro.
- Tukey, J. W. (1972). Some graphic and semigraphic displays. *Statistical papers in honor of George W. Snedecor*, 5:293--316.
- Varian, H. R. (1974). Equity, envy, and efficiency. *Journal of Economic Theory*, 9(1):63 – 91.
- Wilcoxon, F. (1992). Individual comparisons by ranking methods. Em *Breakthroughs in statistics*, pp. 196--202. Springer.
- Wolsey, L. A. & Nemhauser, G. L. (1999). *Integer and combinatorial optimization*, volume 55. John Wiley & Sons, Hoboken.



# Apêndice

A Tabela 6.1 mostra os tempos de execução, em segundos com precisão de dois dígitos seguidos dos números do testes executados, e do números de iterações do algoritmo de Bellman-Ford bem como do Algoritmo 2 para o EFPM.

**Tabela 6.1.** Comparação dos tempos de execução dos algoritmos para o EFPM

n	Bellman-Ford	Algoritmo 1	BELLMAN-FORD	Algoritmo 2	BELLMAN-FORD	Algoritmo 2
1000	0.74/#1	0.37/#1	0.64/#6	0.32/#6	0.72/#11	0.37/#11
	0.40/#2	0.20/#2	0.65/#7	0.33/#7	0.59/#12	0.30/#12
	0.65/#3	0.33/#3	0.36/#8	0.18/#8	0.75/#13	0.39/#13
	0.52/#4	0.27/#4	0.57/#9	0.29/#9	1.04/#14	0.54/#14
	0.84/#5	0.43/#5	0.58/#10	0.29/#10	0.67/#15	0.35/#15
2000	4.18/#1	2.19/#1	4.70/#6	2.41/#6	4.68/#11	2.44/#11
	3.35/#2	1.74/#2	3.16/#7	1.65/#7	4.09/#12	2.14/#12
	4.62/#3	3.52/#3	4.25/#8	2.21/#8	4.20/#13	2.14/#13
	4.28/#4	2.24/#4	4.08/#9	2.13/#9	2.60/#14	1.36/#14
	2.97/#5	1.56/#5	4.57/#10	2.33/#10	2.68/#15	1.35/#15
5000	74.06/#1	34.17/#1	48.33/#6	23.57/#6	42.02/#11	18.96/#11
	48.53/#2	24.22/#2	44.86/#7	22.02/#7	67.45/#12	32.50/#12
	40.25/#3	18.40/#3	58.09/#8	28.52/#8	76.38/#13	38.10/#13
	39.25/#4	19.32/#4	55.14/#9	26.52/#9	45.00/#14	21.76/#14
	46.75/#5	22.76/#5	62.82/#10	31.22/#10	58.37/#15	29.02/#15
10000	304.96/#1	176.05/#1	335.15/#6	184.42/#6	322.86/#11	177.23/#11
	241.80/#2	139.48/#2	290.19/#7	164.01/#7	273.38/#12	158.46/#12
	302.36/#3	162.36/#3	248.69/#8	144.37/#8	301.33/#13	172.00/#13
	351.91/#4	196.47/#4	327.81/#9	180.96/#9	425.29/#14	233.60/#14
	345.52/#5	195.11/#5	269.28/#10	148.88/#10	341.91/#15	196.22/#15
15000	930.12/#1	524.26/#1	772.08/#6	419.99/#6	776.11/#11	427.11/#11
	748.38/#2	414.48/#2	794.75/#7	444.66/#7	925.81/#12	518.57/#12
	810.88/#3	446.22/#3	933.89/#8	518.36/#8	736.72/#13	406.53/#13
	1013.15/#4	552.09/#4	892.60/#9	479.97/#9	1007.04/#14	573.30/#14
	598.82/#5	345.61/#5	1263.47/#10	672.64/#10	765.12/#15	414.32/#15