

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Diretoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Modelagem Matemática e Computacional

Métodos Baseados em Programação Matemática para Solução do Problema da Múltipla Partição de Números Multidimensional

Alexandre Frias Faria

Belo Horizonte
Dezembro de 2022

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
Diretoria de Pesquisa e Pós-Graduação
Programa de Pós-Graduação em Modelagem Matemática e Computacional

Métodos Baseados em Programação Matemática para Solução do Problema da Múltipla Partição de Números Multidimensional

Tese de Doutorado submetida ao Programa de Pós-Graduação em Modelagem Matemática e Computacional como requisito final para a obtenção do título de Doutor em Modelagem Matemática e Computacional.

Aluno: Alexandre Frias Faria

Orientador: Prof. Dr. Sérgio Ricardo de Souza

Co-Orientadora: Profa. Dra. Elisangela Martins de Sá

Belo Horizonte
Dezembro de 2022

Faria, Alexandre Frias
F224m Métodos baseados em programação matemática para solução do problema da múltipla partição de números multidimensional / Alexandre Frias Faria. – 2022.
70 f.

Tese de doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientador: Sérgio Ricardo de Souza.
Coorientadora: Elisângela Martins de Sá.
Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Otimização combinatória – Teses. 2. Análise numérica – Teses. 3. Programação (Matemática) – Teses. 4. Processos de ramificação – Teses. I. Souza, Sérgio Ricardo de. II. Sá, Elisângela Martins de. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.

CDD 519.77

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6º 1592 / Cefet/MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"MÉTODOS BASEADOS EM PROGRAMAÇÃO MATEMÁTICA PARA SOLUÇÃO DO PROBLEMA DA MÚLTIPLA PARTIÇÃO DE NÚMEROS MULTIDIMENSIONAL".

Tese de Doutorado apresentada por **Alexandre Frias Faria**, em 16 de dezembro de 2022, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Sérgio Ricardo de Souza (Orientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Profª. Drª. Elisângela Martins de Sá (Coorientadora)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Carlos Alexandre Silva
Instituto Federal de Minas Gerais

Profª. Drª. Carolina Gil Marcelino
Universidade Federal do Rio de Janeiro

Prof. Dr. Eduardo Uchoa Barboza
Universidade Federal Fluminense

Prof. Dr. Gustavo Campos Menezes
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Marcone Jamilson Freitas Souza
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Profª. Drª. Elizabeth Fialho Wanner
Presidenta do Colegiado do Programa de Pós-Graduação em Modelagem Matemática e Computacional

Agradecimentos

Eu agradeço:

- ao meu orientador, Prof. Dr. Sérgio Ricardo de Souza, pelo apoio, paciência, compreensão e atencioso acompanhamento durante todo o trabalho realizado no doutorado;
- à minha co-orientadora, Profa. Dra. Elisangela Martins de Sá, pelo acompanhamento da pesquisa e ajuda no seu desenvolvimento;
- aos docentes do Programa de Pós-Graduação em Modelagem Matemática e Computacional, pelo conhecimento compartilhado, especialmente ao Prof. Dr. Marcone Jamilson Freitas Souza pela colaboração nos artigos;
- aos discentes e amigos do Programa de Pós-Graduação em Modelagem Matemática e Computacional, pelas várias horas de convivência durante essa longa jornada;
- aos funcionários do CEFET-MG, especialmente aos trabalhadores do restaurante universitário, da limpeza e da cantina, que fizeram meus longos dias no Campus bem mais fáceis e confortáveis;
- ao Centro Federal de Educação Tecnológica de Minas Gerais (CEFET-MG) e ao Programa de Pós-Graduação em Modelagem Matemática e Computacional, pela oportunidade de doutoramento;
- por fim, à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo financiamento da pesquisa, o que viabilizou todo o trabalho desenvolvido.

Resumo

Esta tese apresenta o estudo de técnicas baseadas em programação matemática para a solução do Problema da Múltipla Partição de Números Multidimensional. Este problema tem como objetivo distribuir os vetores de uma dada sequência em k subconjuntos não vazios e disjuntos minimizando o diâmetro dos k vetores resultantes das somas dos elementos de cada subconjunto. O principal foco do trabalho é o desenvolvimento de modelos matemáticos distintos a partir de uma família de restrições centrais. Os modelos matemáticos desenvolvidos servem de base para a aplicação de um método iterativo de solução baseado na inserção e remoção de cortes criados a partir de subproblemas restritos a uma vizinhança da solução corrente, técnica chamada de *Local Branching*. Os resultados descritivos e comparativos testam a validade do modelo proposto comparando-o com um modelo da literatura proposto para um caso mais particular do problema. Para casos mais gerais, os resultados certificam um melhor comportamento da formulação proposta quando combinada à técnica *Local Branching*.

PALAVRAS-CHAVE: Otimização Inteira, Problema da Partição de Números, Programação Matemática, *Local Branching*.

Abstract

This thesis presents the study of techniques based on mathematical programming for solving the Multidimensional Multi-way Number Partitioning Problem. This problem aims to distribute a given sequence of vectors into k non-empty and disjoint subsets minimizing the diameter of the k vectors, resulting from the sums of each subset. The main focus of the work is developing mathematical models from a central core of constraints. The developed mathematical models are used in an iterative method based on adding and removing cuts created from subproblems restricted to a neighborhood of the current solution, a technique called Local Branching. Descriptive and comparative results make the validity test by comparing the proposed model with another model from the literature for a restricted version of the problem. For more general versions, the results show a better behavior of the proposed formulation when combined with the Local Branching technique.

KEYWORDS: *Combinatorial Optimization*, *Number Partitioning Problem*, *Mathematical Programming*, *Local Branching*.

Sumário

Lista de Tabelas viii
Lista de Figuras ix

1 Introdução **1**
1.1 Apresentação Geral 1
1.2 Justificativa 2
1.3 Objetivos 3
1.4 Metodologia 3
1.5 Organização do Texto 4

2 Caracterização do Problema **5**
2.1 Definições e Notação 5
2.2 Problema da Partição de Números 6
2.3 Problema da Múltipla Partição de Números 7
2.4 Problema da Partição de Números Multidimensional 8
2.5 Problema da Múltipla Partição de Números Multidimensional 10

3 Revisão Bibliográfica **14**
3.1 Revisão Literária – TWNPP 14
3.2 Revisão Literária – MWNPP 14
3.3 Revisão Literária – MDTWNPP 15
3.4 Revisão Literária – MDTWNPP 15
3.5 Revisão Literária – Sumário 16
3.6 Publicações 16

4 Formulações Matemáticas para o MDMWNPP **20**
4.1 Análise da Função Objetivo 20
4.2 Dedução das Restrições 1 22
4.3 Dedução das Restrições 2 24
4.4 Comparação entre a Formulação Matemática do MDTWNPP e a Proposta 25
4.5 Eliminação das Simetrias 28

5 Método *Local Branching* **31**
5.1 Restrições *Local Branching* 31
5.2 Construção do Subproblema Local 32
5.3 Algoritmo *Local Branching* 33

6 Resultados **37**
6.1 Configuração geral dos experimentos e das análises estatísticas 37
6.2 Experimentos das Formulações 39
6.2.1 Resultados para $k = 2$ 40
6.2.2 Resultados para $k = 3$ e $k = 4$ 41
6.3 Experimentos usando o algoritmo *Local Branching* 42
6.4 Análise dos resultados para comparação com o modelo (4.31) 44

7 Conclusões Gerais **47**
7.1 Publicações 47
7.2 Considerações Finais 48
7.3 Trabalhos Futuros 49

Referências **50**

A Comparação entre modelos **55**
A.1 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 50$ 55
A.2 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 100$ 56
A.3 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 200$ 57
A.4 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 300$ 58
A.5 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 400$ 59
A.6 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 500$ 60
A.7 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 50$ 61
A.8 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 100$ 62
A.9 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 200$ 63
A.10 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 300$ 64
A.11 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 400$ 65
A.12 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 500$ 66

B Modelos Extras **67**
B.1 Dedução das Restrições 3 67
B.2 Dedução das Restrições 4 68

Lista de Tabelas

2.1 Exemplo de soluções do MWNPP para diferentes valores de k referentes ao Exemplo 2.3.1. 8
2.2 Valor de função objetivo das soluções apresentadas na Tabela 2.1. . . 8
2.3 Exemplo 2.5.1: Avaliação da função objetivo 11
2.4 Exemplo 2.5.1: Cálculo da função objetivo 11
2.5 Exemplo 2.5.1: Soluções factíveis e soluções ótimas para valores de $k \in \{3, 4, 5\}$ 11

3.1 Sumário da Bibliografia do MDMWNPP. 17

6.1 Dados descritivos sumarizados. 38
6.2 Tabela de contingência. 38
6.3 Múltiplas comparações usando Teste de Wilcoxon ou Fisher. 39
6.4 Resultados obtidos através do modelo (4.34). 40
6.5 Resultados obtidos através do modelo (4.23) usando $k = 2$. 41
6.6 Teste de Wilcoxon pareado ($\Delta = \mu_{k=2} - \mu_{kojic}$). 41
6.7 Contagem de resultados obtidos pelo modelo (4.23) em relação ao modelo (4.34). 41
6.8 Resultados do modelo (4.23) para $k = 3$. 41
6.9 Resultados do modelo (4.23) para $k = 4$. 42
6.10 Sumário de resultados para o MILP (4.23) com $k = 3$. 42
6.11 Sumário de resultados para o MILP (4.23) com $k = 4$. 43
6.12 Sumário de resultados para o MILP (4.23) com $k = 5$. 43
6.13 Sumário de resultados para o algoritmo *Local Branching* com $k = 3$. . 43
6.14 Sumário de resultados para o algoritmo *Local Branching* com $k = 4$. . 43
6.15 Sumário de resultados para o algoritmo *Local Branching* com $k = 5$. . 44
6.16 Resultados da aplicação do Teste de Wilcoxon pareado sobre os valores de UB, para a hipótese $H_1 : \Delta_{UB} = \mu_{Local} - \mu_{MILP} < 0$, e sobre os valores de LB, para a hipótese $H_1 : \Delta_{LB} = \mu_{Local} - \mu_{MILP} > 0$. . 44
6.17 Número de vezes em que o modelo atinge o melhor UB da literatura. 45
6.18 Proporção de melhor solução alcançada. 45
6.19 Resultados dos p-valores ajustados para a proporção de melhor solução alcançada por pares. 46
6.20 Número de vezes em que o modelo atinge o resultado ótimo. 46

A.1 Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 50$. 55
A.2 Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 100$ 56
A.3 Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 200$ 57

A.4 Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 300$ 58
A.5 Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 400$ 59
A.6 Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 500$ 60
A.7 Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 50$ 61
A.8 Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 100$ 62
A.9 Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 200$ 63
A.10 Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 3000$ 64
A.11 Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 50$ 65
A.12 Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 500$ 66

Lista de Figuras

2.1 Exemplo geométrico de uma solução para o MDMWNPP. 12

5.1 Representação da exploração do Algoritmo 1 assumindo que todos os subproblemas encontram somente soluções factíveis de melhora sem otimalidade comprovada 35

5.2 Efeito das restrições adicionadas na linha 12 do algoritmo 1 no espaço de busca do Algoritmo 1 assumindo que todos os subproblemas encontram soluções ótimas de melhora. 36

Capítulo 1

Introdução

Este Capítulo está organizado da seguinte maneira. A Seção 1.1 introduz a família dos problemas de partição de números e mostra sua relação com problemas similares da literatura. A Seção 1.2 justifica a importância do trabalho para o campo de pesquisa em otimização. A Seção 1.3 apresenta as contribuições teórica e prática desta pesquisa. A Seção 1.4 apresenta um resumo de todo o processo de estudo e experimentos a serem percorridos para estabelecer os objetivos apresentados. A Seção 1.5 apresenta uma descrição dos capítulos posteriores desta tese.

1.1 Apresentação Geral

Problemas de otimização aparecem em diversas disciplinas que envolvem análises quantitativas. O desenvolvimento de métodos para a resolução de problemas de otimização tanto matemática quanto computacionalmente tem sido de grande interesse. Os problemas de otimização podem ser classificados em várias categorias. Uma dessas categorias é a dos problemas de otimização combinatória, que estão fortemente relacionados por suas características básicas e propriedades discretas de suas soluções (Bazaraa et al., 2009).

O problema principal abordado nesta tese é um problema de otimização combinatória clássico da literatura. Trata-se de uma generalização de um dos problemas NP-completos mais simples conhecidos, chamado Problema da Partição de Números (*Two-Way Number Partitioning Problem* – TWNPP). Este problema consiste em, dada uma sequência numérica V, encontrar um particionamento dos seus índices em duas partes, de modo que as somas dos elementos de cada parte caibam no menor intervalo possível. As generalizações deste problema são:

(i) Problema da Múltipla Partição de Números (*Multi-Way Number Partitioning Problem* – MWNPP), que é uma extensão direta do TWNPP que expande o número de partes nas quais os índices da sequência V devem ser distribuídos para um valor inteiro $k \geq 2$;

(ii) Problema da Partição de Números Multidimensional (*Multidimensional Two-Way Number Partitioning Problem* – MDTWNPP), que considera uma sequência V de vetores de dimensão m. Seu objetivo é encontrar um particionamento

dos índices dos termos de V em duas partes, de modo que a maior distância entre vetores resultantes da soma dos vetores associados a cada parte seja minimizada;

(iii) Problema da Múltipla Partição de Números Multidimensional (*Multidimensional Multi-Way Number Partitioning Problem* – MDMWNPP), que generaliza o MDTWNPP para k partes, em que k é um valor inteiro maior ou igual a 2.

Estes quatro problemas constituem o que pode ser denominado Família de Problemas de Partição de Números, que é o objeto central deste estudo. As generalizações ocorrem por duas vias de complexificação: aumento da dimensão ou do particionamento.

Dentre os problemas de Otimização Combinatória mais conhecidos da literatura, os mais próximos ao MDMWNPP são:

(i) Problema de Escalonamento em Máquinas Idênticas (*Identical Machine Scheduling Problem* – IMSP);

(ii) Problema da Mochila (*Knapsack Problem* – KP);

(iii) Problema do Empacotamento (*Bin Packing Problem* – BPP).

Um modelo matemático para o MDMWNPP herda características dos modelos de otimização inteira para os problemas IMSP, BPP e KP, descritos em, dentre outros trabalhos, Crescenzi et al. (1997) e Ausiello et al. (2003b).

A abordagem de solução usada nesta tese para resolver o MDMWNPP é a programação matemática e suas técnicas de solução. É interessante notar que poucos problemas são mais difíceis do que o problema de programação matemática, no sentido da teoria computacional. No entanto, é comum a redução de quase todo tipo de problema de otimização a um problema de programação matemática, Problema de Programação Linear Inteira Mista (*Mixed Integer Linear Programming* – MILP) ou Problema de Programação Linear Inteira (*Integer Linear Programming* – ILP), como abordagem principal para proposição de novos algoritmos ou métodos (Karp, 1972).

O que torna o uso da programação matemática uma abordagem bem fundamentada e funcional é o fato de todos os problemas NP-completos conhecidos admitirem uma redução polinomial a um ILP ou a um MILP (Cook, 1971; Karp, 1972). O que se faz em tais abordagens é transformar uma instância de um problema $\mathcal{A}$ qualquer em uma instância de MILP tal que uma solução para o MILP implique em uma solução para $\mathcal{A}$. Logo, tendo-se um algoritmo para resolver MILPs, resolve-se o problema $\mathcal{A}$.

1.2 Justificativa

Este trabalho é importante do ponto de vista da pesquisa em otimização teórica e prática. O estudo de um problema teórico ajuda a esclarecer diversos problemas aplicados. São as propriedades específicas, destacadas em cada problema teórico, que permitem uma compreensão modularizada dos problemas práticos vistos na indústria. O MDMWNPP e sua família de problemas tem propriedades que os tornam

muito desafiadores para métodos envolvendo programação matemática, como uma função objetivo que pode se anular, tanto na solução relaxada quanto na inteira. Descobrir como romper esta barreira pode trazer grandes benefícios ao campo da programação matemática, aumentando o leque de aplicações efetivas do método exato mais flexível conhecido na Otimização.

Por outro lado, pode-se listar algumas aplicações diretas do MDMWNPP como: balanceamento de cargas, criptografia (Pop e Matei, 2013b) e a escolha de jogadores para criar times justos (Hacibeyoglu et al., 2018). Outro tipo de aplicação é a distribuição de n bens indivisíveis e de valor não generalizável com uma única medida entre k herdeiros. A relação entre os Problemas de Clusterização e o MDMWNPP é intuitiva e se expressa no compartilhamento de duas famílias de restrições comuns em suas respectivas formulações matemáticas (Piccialli et al., 2022). Assim, as contribuições para o MDMWNPP atingem diversos campos da pesquisa teórica e aplicada.

1.3 Objetivos

O objetivo geral desta pesquisa é desenvolver métodos de resolução baseados em programação matemática para a solução da família de problemas de partição de números.

Os objetivos específicos são:

(i) Desenvolver formulações com variáveis binárias e reais e métodos de solução para o MDMWNPP baseados em programação matemática;

(ii) Descobrir como um número maior ou menor de variáveis reais ou binárias influenciam na eficiência dos modelos propostos.

(iii) Deduzir restrições para eliminar possíveis simetrias nas soluções dos modelos propostos e investigar o impacto dessas novas restrições na qualidade da solução;

(iv) Aplicar métodos de geração iterativa de cortes, adaptados aos modelos propostos, levando em conta especificidades de algumas de suas famílias de restrições.

1.4 Metodologia

Para realização da parte teórica desta pesquisa, necessita-se do desenvolvimento de um enunciado formal bem detalhado para o MDMWNPP. Com uma definição bem elaborada, é possível desenvolver restrições, desigualdades válidas, redução de variáveis reais, eliminação de simetrias e métodos de solução usando técnicas de programação matemática. Esta fase da pesquisa conta com o estudo dos livros Williams (2013), Bertsimas e Weismantel (2005) e Hooker (2011) e alguns artigos clássicos como Schrijver (1980).

Os passos a seguir definem a metodologia de pesquisa em ordem de prioridade.

(i) **Revisão da Literatura:** procurar pelos trabalhos existentes que abordem problemas relacionados ao MDMWNPP. Nessa lista estão todas as variantes

listadas da Família de Problemas de Partição de Números, Problema da Mochila, Problema do Empacotamento e Escalonamento de Máquinas;

(ii) **Fundamentação Teórica:** estudar os conceitos, teorias e métodos presentes no campo de pesquisa em otimização partindo dos livros clássicos e de suas referências;

(iii) **Caracterização do Problema:** caracterizar e definir formalmente o MDMWNPP para evitar ambiguidades e facilitar a dedução de resultados auxiliares;

(iv) **Solução Proposta:** desenvolver modelos matemáticos variados e propor técnicas de refinamento e eliminação de simetrias. Esses modelos permitirão o emprego de algoritmos clássicos do campo de otimização como *Planos de Corte, Geração de Colunas, Enumeração e Poda* em futuras pesquisas;

(v) **Realização de Experimentos:** realizar experimentos computacionais para um conjunto de instâncias do MDMWNPP em um ambiente computacional controlado;

(vi) **Avaliação dos Resultados Experimentais:** analisar os resultados obtidos com uso de inferência estatística sobre o status e valor de função objetivo da solução encontrada, inteira e relaxada.

1.5 Organização do Texto

A organização desta Tese segue de acordo com as descrições de cada capítulo. O Capítulo 2 apresenta o enunciado formal do MDMWNPP e exemplos numéricos de possíveis soluções para um maior esclarecimento de suas propriedades. O Capítulo 3 mostra uma revisão dos principais trabalhos relacionados ao MDMWNPP e um sumário das publicações derivadas desta Tese. O Capítulo 4 usa a função objetivo de Pop e Matei (2013b) e resultados do Capítulo 2 para apresentar uma dedução das restrições do modelo matemático proposto, além de uma formulação apresentada em Kojić (2010) para o MDTWNPP, citando algumas imprecisões deste. O Capítulo 5 propõe uma metodologia de solução para os modelos matemáticos propostos baseados no Método *Local Branching*. O Capítulo 6 apresenta uma análise estatística dos dados das tabelas de resultados obtidos, contendo as instâncias testadas, limites superiores, limites inferiores e tempo para todos os métodos propostos e avaliados, bem como uma reprodução dos resultados de Kojić (2010) e a análise dos resultados de Nikolic et al. (2022). O Capítulo 7 exibe as publicações desta Tese e artigos em andamento, o que é possível afirmar sobre os resultados apresentados na Tese, levando em conta todas as análises do Capítulo 6 e cita possíveis trabalhos futuros.

Capítulo 2

Caracterização do Problema

Algumas notações e definições iniciais são necessárias para o bom entendimento do texto. Os conceitos de partição, norma e diâmetro são apresentados na Seção 2.1. As referências principais dos conceitos desta seção são Stanley (1997) e Lima (2004). Após estabelecer estas definições, é possível enunciar todas as variantes da Família de Problemas de Partição de Números nas Seções 2.2, 2.3, 2.4 e 2.5, mantendo a mesma natureza da função objetivo em todos os enunciados.

2.1 Definições e Notação

Usa-se a notação $I_n = \{y \in \mathbb{Z} \ : \ 1 \leq y \leq n\}$ para representar o conjunto fechado de todos os inteiros entre 1 e n. Quando for mencionado um vetor indexado em i, $v_i = (v_{i1}, v_{i2}, \ldots, v_{il}, \ldots, v_{im}) \in \Re^m$, considera-se v_{il} como sua l-ésima coordenada.

Uma partição de um conjunto X é uma coleção de subconjuntos, mutuamente disjuntos, cuja união forma X. Uma descrição formal é apresentada na Definição 2.1.1.

Definição 2.1.1. *Seja X um conjunto e t um inteiro variando entre 1 e $|X|$. Uma partição de X é uma família de subconjuntos $\{B_j\}_{j \in I_t}$ com as seguintes propriedades:*

(i) $B_j \neq \emptyset, \quad \forall j \in I_t,$

(ii) $B_i \cap B_j = \emptyset, \quad \forall 1 \leq i < j \leq t,$

(iii) $\bigcup_{j \in I_t} B_j = X.$

A Definição 2.1.1 é utilizada para qualquer valor de t possível. Por outro lado, uma k-partição de um conjunto X é uma partição de X, com exatamente k subconjuntos B_j não vazios.

Definição 2.1.2. *Seja X um conjunto e k um inteiro positivo fixo. Uma k-partição de X é uma família de subconjuntos $\{B_j\}_{j \in I_k}$ com as seguintes propriedades:*

(i) $B_j \neq \emptyset, \quad \forall j \in I_k,$

(ii) $B_i \cap B_j = \emptyset, \quad \forall 1 \leq i < j \leq k,$

(iii) $\bigcup_{j \in I_k} B_j = X$.

Nesse texto, os subconjuntos B_j pertencentes à partição são chamados de partes para evitar possíveis conflitos de linguagem. Por exemplo, se durante alguma passagem for necessário selecionar um "subconjunto de subconjuntos" da partição ou alguma outra situação similar.

As Definições 2.1.3, 2.1.4 e 2.1.5 apresentam uma formalização dos conceitos de norma, distância entre dois elementos de um conjunto e diâmetro de um conjunto, respectivamente. Usa-se $p = \infty$ para representar a norma infinita.

Definição 2.1.3. *Uma função* $\|.\|_p : \Re^n \to \Re^+$ *é uma norma de* $\Re^n$ *se, para quaisquer* $x, y \in \Re^n$ *e* $a \in \Re$*:*

(i) $\|x\|_p \geq 0$,

(ii) $\|x\|_p = 0 \iff x = \vec{0}$,

(iii) $\|a.x\|_p = |a|.\|x\|_p$,

(iv) $\|x + y\|_p \leq \|x\|_p + \|y\|_p$.

Definição 2.1.4. *Sejam dois vetores* $x, y \in \Re^n$*. A distância entre* x *e* y *é definida por*

$$d(x, y) = \|x - y\|_p$$

Definição 2.1.5. *Seja um subconjunto* $X \in \Re^n$*. O diâmetro de* X *é definido por:*

$$diam(X) = \max_{x_1, x_2 \in X} \{d(x_1, x_2)\}$$

2.2 Problema da Partição de Números

Esta é a primeira versão do Problema de Partição de Números (*Two-Way Number Partitioning Problem* – TWNPP), listado em Karp (1972) e, portanto, bem conhecido e tratado em diversos artigos, como Gent e Walsh (1998) e Pedroso e Kubo (2010). O enunciado formal do TWNPP é apresentado na Definição 2.2.1.

Definição 2.2.1. *Seja* $V = \{v_i\}_{i \in I_n}$ *uma sequência numérica real, de modo que cada elemento* $v_i \in \Re$*. O objetivo do TWNPP é encontrar uma 2-partição de* I_n*, na forma* $\{A_1, A_2\}$*, que minimize o diâmetro do multiconjunto* $\left\{ \sum_{i \in A_1} v_i, \sum_{i \in A_2} v_i \right\}$*. A função objetivo é dada por:*

$$f(\{A_1, A_2\}) = \left| \sum_{i \in A_1} v_i - \sum_{i \in A_2} v_i \right| \tag{2.1}$$

O Exemplo 2.2.1 apresenta uma instância do TWNPP, uma solução factível para a instância considerada e uma solução ótima para esta mesma instância.

Exemplo 2.2.1. *A sequência*

$$V = \{87, 6, 5, 45, 34, 2, 24, 12, 7, 6, 54, 34\}$$

pode ser particionada na forma:

$$\underbrace{\{87, 34, 24, 6, 5\}}_{\sum_{i\in A_1} v_i = 156}, \quad \underbrace{\{54, 45, 34, 12, 7, 6, 2\}}_{\sum_{i\in A_2} v_i = 160}$$

O valor da função objetivo (2.1) para essa partição é, portanto, $160 - 156 = 4$. *Por outro lado, a partição ótima é dada por:*

$$\underbrace{\{87, 34, 24, 6, 5, 2\}}_{\sum_{i\in A_1} v_i = 158}, \quad \underbrace{\{54, 45, 34, 12, 7, 6\}}_{\sum_{i\in A_2} v_i = 158}$$

com valor de função objetivo $158 - 158 = 0$. □

2.3 Problema da Múltipla Partição de Números

A primeira generalização do TWNPP é o Problema da Múltipla Partição de Números (*Multi-Way Number Partitioning Problem* – MWNPP), que aumenta a quantidade de partes nos quais o conjunto de índices I_n deve ser particionado. Os textos principais que abordam esse problema são Karmarkar e Karp (1982), Korf (2009), Korf et al. (2013). Um trabalho recente de forte interesse, pela sua completude, é Schreiber (2014). O enunciado formal para o MWNPP é apresentado na Definição 2.3.1.

Definição 2.3.1. *Seja* $V = \{v_i\}_{i\in I_n}$ *uma sequência numérica real, de modo que cada elemento* $v_i \in \Re$, *e considere* k *um número inteiro positivo. O MWNPP busca encontrar uma* k*-partição dos índices de* I_n, *na forma* $\{A_j\}_{j\in I_k}$, *que minimize o diâmetro do multiconjunto* $\left\{\sum_{i\in A_j} v_i\right\}_{j\in I_k}$. *Assim, a função objetivo deste problema é dada por:*

$$f(\{A_j\}_{j\in I_k}) = \max_{j'\neq j}\left\{\sum_{i\in A_{j'}} v_i - \sum_{i\in A_j} v_i\right\} \tag{2.2}$$

$$= \max_{j}\left\{\sum_{i\in A_j} v_i\right\} - \min_{j}\left\{\sum_{i\in A_j} v_i\right\} \tag{2.3}$$

O Exemplo 2.3.1 a seguir mostra algumas possibilidades para esse problema.

Exemplo 2.3.1. *A sequência*

$$V = \{11, 25, 13, 34, 89, 65, 43, 96, 56, 87\}$$

pode ser particionada das maneiras mostradas na Tabela 2.1 para os valores $k \in \{3, 4, 5, 6\}$. *É interessante observar, nos resultados mostrados na Tabela 2.1, que as partes não precisam ter a mesma cardinalidade.*

Tabela 2.1: Exemplo de soluções do MWNPP para diferentes valores de k referentes ao Exemplo 2.3.1.

	Factível	Ótima
$k = 3$	$\underbrace{\{96, 43, 34\}}_{173}$, $\underbrace{\{89, 56, 25\}}_{170}$, $\underbrace{\{87, 65, 13, 11\}}_{176}$	$\underbrace{\{89, 87\}}_{176}$, $\underbrace{\{25, 13, 34, 43, 56\}}_{171}$, $\underbrace{\{11, 65, 96\}}_{172}$
$k = 4$	$\underbrace{\{96, 25, 13\}}_{134}$, $\underbrace{\{89, 34\}}_{123}$, $\underbrace{\{87, 43\}}_{130}$, $\underbrace{\{65, 56, 11\}}_{132}$	$\underbrace{\{25, 13, 89\}}_{127}$, $\underbrace{\{43, 87\}}_{130}$, $\underbrace{\{34, 96\}}_{130}$, $\underbrace{\{11, 65, 56\}}_{132}$
$k = 5$	$\underbrace{\{11, 89, 13\}}_{113}$, $\underbrace{\{25, 87\}}_{112}$, $\underbrace{\{96\}}_{96}$, $\underbrace{\{34, 65\}}_{99}$, $\underbrace{\{43, 56\}}_{99}$	$\underbrace{\{11, 89\}}_{100}$, $\underbrace{\{25, 87\}}_{112}$, $\underbrace{\{13, 96\}}_{109}$, $\underbrace{\{34, 65\}}_{99}$, $\underbrace{\{43, 56\}}_{99}$
$k = 6$	$\underbrace{\{96\}}_{96}$, $\underbrace{\{89\}}_{89}$, $\underbrace{\{87\}}_{87}$, $\underbrace{\{11, 13, 65\}}_{89}$, $\underbrace{\{56, 25\}}_{81}$, $\underbrace{\{43, 34\}}_{77}$	$\underbrace{\{96\}}_{96}$, $\underbrace{\{89\}}_{89}$, $\underbrace{\{87\}}_{87}$, $\underbrace{\{11, 25, 43\}}_{79}$, $\underbrace{\{13, 65\}}_{78}$, $\underbrace{\{34, 56\}}_{90}$

Tabela 2.2: Valor de função objetivo das soluções apresentadas na Tabela 2.1.

	Factível	Ótima
$k = 3$	176 - 170 = 6	176 - 171 = 5
$k = 4$	134 - 123 = 11	132 - 127 = 5
$k = 5$	113 - 96 = 17	112 - 99 = 13
$k = 6$	96 - 77 = 19	96 - 78 = 18

Os valores de função objetivo para cada valor de k considerado, para as soluções apresentadas na Tabela 2.1, estão apresentados na Tabela 2.2. □

2.4 Problema da Partição de Números Multidimensional

A segunda generalização do TWNPP é o Problema da Partição de Números Multidimensional (*Multidimensional Two-Way Number Partitioning Problem* – MDTWNPP). Neste problema, considera-se que a dimensão m dos elementos v_i seja tal que $m > 1$, transformando os elementos da sequência V em vetores, na forma:

$$V = \{v_i \in \Re^m \ : \ \forall i \in I_n\}.$$

Para definir este problema, faz-se necessário o uso dos conceitos de norma, distância e diâmetro apresentados nas Definições 2.1.3, 2.1.4 e 2.1.5. Seu enunciado matemático, proposto por Kojić (2010), é muito parecido com o TWNPP. A busca é por uma 2-partição, na forma $\{A_1, A_2\}$, que minimize a função objetivo do problema. O enunciado formal deste problema é apresentado na Definição 2.4.1.

Definição 2.4.1. *Seja $V = \{v_i\}_{i \in I_n}$ uma sequência de vetores em $\Re^m$. O objetivo do MDTWNPP é encontrar uma 2-partição do conjunto de índices I_n, na forma $\{A_1, A_2\}$, que minimize a distância entre os vetores $\sum_{i \in A_1} v_i$ e $\sum_{i \in A_2} v_i$ resultantes da soma dos vetores referentes a cada parte. A função objetivo, considerando a distância definida pela norma infinita, é dada por:*

$$f(\{A_1, A_2\}) = \max_{l \in I_m} \left\{ \left| \sum_{i \in A_1} v_{il} - \sum_{i \in A_2} v_{il} \right| \right\} \tag{2.4}$$

sendo v_{il} o l-ésimo elemento do vetor v_i.

O Exemplo 2.4.1 mostra uma instância do MDTWNPP, identificando uma solução factível e uma solução ótima para esta instância.

Exemplo 2.4.1. *O conjunto:*

$$V = \left\{ \begin{bmatrix} 8 \\ 3 \\ 6 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 12 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 17 \\ 5 \\ 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \\ 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \\ 10 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 6 \\ 1 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \\ 6 \\ 3 \end{bmatrix} \right\}$$

em que $m = 4$, pode ser particionado como:

$$\{v_i \ : \ i \in A_1\} = \underbrace{\left\{ \begin{bmatrix} 8 \\ 3 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 17 \\ 5 \\ 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \\ 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \\ 10 \\ 3 \end{bmatrix} \right\}}_{\sum_{i \in A_1} v_i = \begin{bmatrix} 45 \\ 18 \\ 30 \\ 16 \end{bmatrix}},$$

$$\{v_i \ : \ i \in A_2\} = \underbrace{\left\{ \begin{bmatrix} 8 \\ 3 \\ 6 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 12 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 6 \\ 1 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \\ 6 \\ 3 \end{bmatrix} \right\}}_{\sum_{i \in A_2} v_i = \begin{bmatrix} 29 \\ 12 \\ 30 \\ 10 \end{bmatrix}}$$

tendo valor de função objetivo dado por:

$$\max \left\{ \left| \begin{bmatrix} 45 \\ 18 \\ 30 \\ 16 \end{bmatrix} - \begin{bmatrix} 29 \\ 12 \\ 30 \\ 10 \end{bmatrix} \right| \right\} = \max \left\{ \left| \begin{bmatrix} 16 \\ 6 \\ 0 \\ 6 \end{bmatrix} \right| \right\} = 16$$

A partição ótima para esta instância é dada por:

$$\{v_i \ : \ i \in A_1\} \ = \ \underbrace{\left\{ \begin{bmatrix} 17 \\ 5 \\ 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \\ 10 \\ 5 \end{bmatrix}, \begin{bmatrix} 10 \\ 5 \\ 10 \\ 3 \end{bmatrix} \right\}}_{\sum_{i \in A_1} v_i = \begin{bmatrix} 37 \\ 15 \\ 30 \\ 13 \end{bmatrix}},$$

$$\{v_i \ : \ i \in A_2\} \ = \ \underbrace{\left\{ \begin{bmatrix} 8 \\ 3 \\ 6 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 12 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 0 \\ 3 \end{bmatrix}, \begin{bmatrix} 8 \\ 3 \\ 6 \\ 1 \end{bmatrix}, \begin{bmatrix} 5 \\ 3 \\ 6 \\ 3 \end{bmatrix} \right\}}_{\sum_{i \in A_2} v_i = \begin{bmatrix} 37 \\ 15 \\ 30 \\ 13 \end{bmatrix}}$$

tendo valor de função objetivo dado por:

$$\max \left\{ \left\| \begin{bmatrix} 37 \\ 15 \\ 30 \\ 13 \end{bmatrix} - \begin{bmatrix} 37 \\ 15 \\ 30 \\ 13 \end{bmatrix} \right\| \right\} = \max \left\{ \left\| \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \right\| \right\} = 0$$

□

2.5 Problema da Múltipla Partição de Números Multidimensional

O Problema da Múltipla Partição de Números Multidimensional (*Multidimensional Multi-Way Number Partitioning Problem* – MDMWNPP) é uma generalização tanto do MWNPP quanto do MDTWNPP. Neste problema, uma sequência de vetores em $\Re^m$ deve ser dividida em k subconjuntos, de modo que o diâmetro dos k vetores resultantes da soma dos vetores em cada parte seja o menor possível. O MDMWNPP é apresentado, originalmente, em Pop e Matei (2013b). Seu enunciado formal é apresentado na Definição 2.5.1.

Definição 2.5.1. *Seja $V = \{v_i\}_{i \in I_n}$ uma sequência de vetores em $\Re^m$, e considere k um número inteiro positivo. O MDMWNPP consiste em encontrar uma k-partição do conjunto de índices I_n, na forma $\{A_j\}_{j \in I_k}$, que minimize o diâmetro do multiconjunto $\left\{\sum_{i \in A_j} v_i\right\}_{j \in I_k}$, diâmetro este dado por:*

$$f(\{A_j\}_{j \in I_k}) = \max_{j_1, j_2 \in I_k} \left\{ \left\| \sum_{i \in A_{j_1}} v_i - \sum_{i \in A_{j_2}} v_i \right\|_\infty \right\} \tag{2.5}$$

Dada uma k-partição, o Exemplo 2.5.1 esclarece o significado do problema e do cálculo da função objetivo, dada pela Expressão (2.5).

Exemplo 2.5.1. *Considere a sequência*

$$V = \{(14,48,23)_1, (87,61,48)_2, (76,14,23)_3, (24,25,33)_4, (84,13,49)_5, (25,48,78)_6, (56,14,73)_7, (55,21,20)_8, (16,13,86)_9, (74,55,31)_{10}\},$$

em que $n = 10$ *e* $m = 3$.

Dada uma solução

$$S = \left\{ \overbrace{\{1,3,8,9\}}^{A_1}, \overbrace{\{2,4,7\}}^{A_2}, \overbrace{\{5,6,10\}}^{A_3} \right\},$$

para $k = 3$*, o cálculo da função objetivo é feito da seguinte maneira. A Tabela 2.3 mostra a separação dos vetores de* V *de acordo com a solução* S*. A última linha da Tabela 2.3 apresenta a soma dos elementos dos vetores contidos na mesma parte, gerando um conjunto de* 3 *vetores.*

Tabela 2.3: Exemplo 2.5.1: Avaliação da função objetivo

$v_i \in A_1$	$v_i \in A_2$	$v_i \in A_3$
$(14,48,23)_1$	$(87,61,48)_2$	$(84,13,49)_5$
$(76,14,23)_3$	$(24,25,33)_4$	$(25,48,78)_6$
$(55,21,20)_8$	$(56,14,73)_7$	$(74,55,31)_{10}$
$(16,13,86)_9$		
$(161,96,152)$	$(167,100,154)$	$(183,116,158)$

Usando os vetores da última linha da Tabela 2.3, a diferença entre o maior e menor elemento para cada coordenada é calculada. O maior destes valores, dado por 22*, é o diâmetro do conjunto e o valor da função objetivo associado à solução, como mostra a Tabela 2.4.*

Tabela 2.4: Exemplo 2.5.1: Cálculo da função objetivo

$\sum_{i \in A_1} v_i$	$(\underline{161}, \underline{96}, \underline{152})$
$\sum_{i \in A_2} v_i$	$(167, 100, 154)$
$\sum_{i \in A_3} v_i$	$(\overline{183}, \overline{116}, \overline{158})$
$\overline{v_{il}} - \underline{v_{i'l}}$	$\{22, 20, 6\}$
$f(\{A_1, A_2, A_3\})$	$\max\{22, 20, 6\}$

Para os valores de $k \in \{3,4,5\}$ *e a mesma sequência* V*, a Tabela 2.5 exibe soluções factíveis e as respectivas soluções ótimas.*

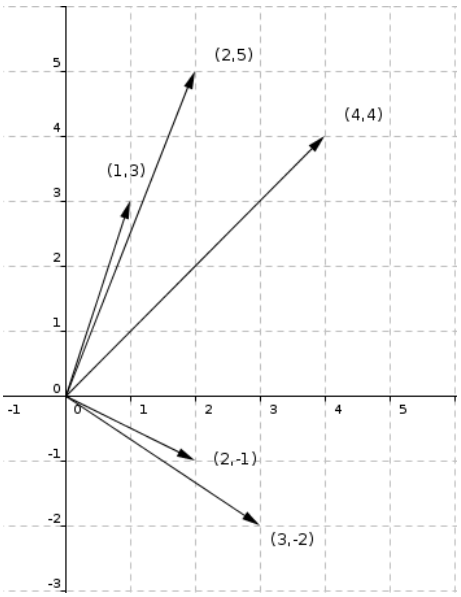
Tabela 2.5: Exemplo 2.5.1: Soluções factíveis e soluções ótimas para valores de $k \in \{3,4,5\}$

k	Factível	$f(.)$	Ótima	$f(.)$
3	$\{1,2,3,9\}, \{4,7,10\}, \{5,6,8\}$	54	$\{1,3,8,9\}, \{2,4,7\}, \{5,6,10\}$	22
4	$\{1,9,10\}, \{2,6\}, \{3,4,5\}, \{7,8\}$	81	$\{1,4,5\}, \{2,7\}, \{3,6\}, \{8,9,10\}$	44
5	$\{1,4,5\}, \{2\}, \{3,6\}, \{7,10\}, \{8,9\}$	59	$\{1,5\}, \{2,4\}, \{3,6\}, \{7,8\}, \{9,10\}$	51

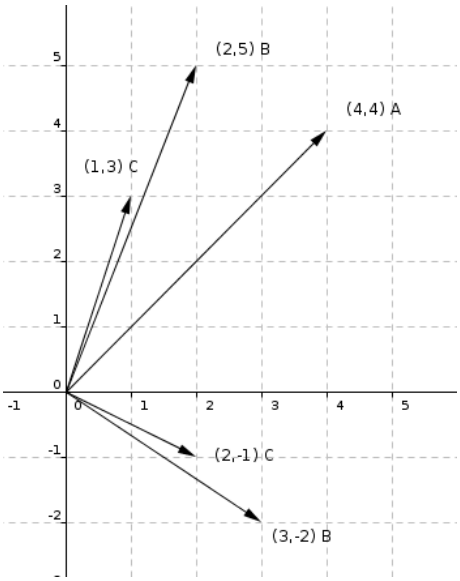
É importante notar que, apesar do que aparenta mostrar a Tabela 2.5, a suposição de o crescimento no valor de k implicar no aumento do valor ótimo da função objetivo é falsa. Um bom contraexemplo disso é uma sequência V com $n = 8$ elementos $v_i = (10, 10, 10)$, $\forall i \in I_n$. Quando k divide n, o valor ótimo é 0. Quando k não divide n, o valor ótimo é 10 □

A Figura 2.1 mostra uma 3-partição ótima para o conjunto de vetores $V = \{(1, 3), (4, 4), (3, -2), (2, 5), (2, -1)\}$. Esta figura é uma representação geométrica do exemplo encontrado em Pop e Matei (2013b).

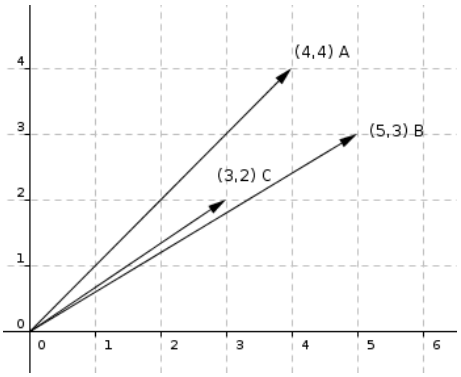
Figura 2.1: Exemplo geométrico de uma solução para o MDMWNPP.



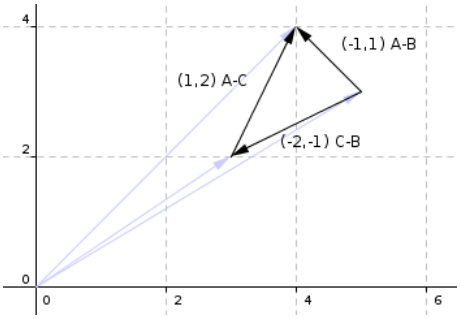
(a) Sequência de vetores V.



(b) 3-partição ótima



(c) $\{\sum_{i\in A} v_i, \sum_{i\in B} v_i, \sum_{i\in C} v_i\}$.



(d) $(\sum_{i\in A} v_i - \sum_{i\in C} v_i), (\sum_{i\in A} v_i - \sum_{i\in B} v_i), (\sum_{i\in C} v_i - \sum_{i\in B} v_i)$.

O valor de função objetivo é a maior coordenada em módulo da diferença entre os vetores resultantes, na forma:

$$\max_{l\in I_2} \left\{ \left| \sum_{i\in A} v_{il} - \sum_{i\in C} v_{il} \right|, \left| \sum_{i\in A} v_{il} - \sum_{i\in B} v_{il} \right|, \left| \sum_{i\in C} v_{il} - \sum_{i\in B} v_{il} \right| \right\} = 2 \qquad (2.6)$$

É interessante observar que, neste problema, não só o cálculo da função objetivo é particularmente complicado, como também é difícil fazer a busca completa em todo o espaço das soluções factíveis.

Capítulo 3

Revisão Bibliográfica

Este capítulo apresenta uma revisão bibliográfica acerca dos problemas da família de problemas de partição de números. Esta revisão literária está separada nas seções 3.1, 3.2, 3.3, 3.4 e 3.5. A Seção 3.6 descreve as publicações advindas dessa Tese.

3.1 Revisão Literária – TWNPP

O *Two-Way Number Partitioning Problem* (TWNPP) é um problema muito conhecido na literatura. Ele foi listado de modo formal em Karp (1972) como um dos problemas NP-completos básicos. Esta referência também demonstra uma série de equivalências entre o TWNPP e outros problemas NP-completos.

Os primeiros algoritmos exatos trivialmente adaptáveis para a solução do TWNPP foram devidos a Horowitz e Sahni (1974) e Schroeppel e Shamir (1981), ambos propostos para resolver o Problema da Mochila. Em Korf (1998) encontram-se duas propostas de métodos exatos do tipo *Branch and Bound* baseadas em heurística, quais sejam: *Complete Greedy Algorithm* (CGA), usando a heurística *Longest Processing Time* (LPT), proposta em Graham (1969), e o *Complete Karmarkar Karp Algorithm* (CKKA), usando o *Differencing Method*, proposto em Karmarkar e Karp (1982).

3.2 Revisão Literária – MWNPP

A primeira generalização do TWNPP é o *Multi-Way Number Partitioning Problem* (MWNPP), que expande o número de partes nas quais os índices da sequência V devem ser distribuídos. O MWNPP aparece enunciado explicitamente em Karmarkar e Karp (1982), no contexto da análise do *Differencing Method*, mais conhecido como Heurística de Karmarkar-Karp (HKK), proposta em Karmarkar e Karp (1982). Esta heurística está focada na ideia de dividir os maiores números em partes distintas, inserindo as diferenças entre os elementos retirados no conjunto dos números não-alocados enquanto este for não vazio. Gent e Walsh (1998) mostram que o MWNPP é um problema muito difícil de ser solucionado por meta-heurísticas de uso geral, como Algoritmos Genéticos, *Simulated Annealing* e outras. Em muitos casos, esses métodos perdem em tempo e desempenho para a HKK e até mesmo

para a heurística LPT. Os algoritmos exatos presentes em Korf (1998) apresentam versões adaptadas para resolver o MWNPP.

A primeira melhoria desses trabalhos acontece com o algoritmo *Recursive Number Partitioning* (RNP), proposto por Korf (2009), que trabalha com a resolução de subproblemas menores derivados do MWNPP. Por meio de sucessivas conversões do MWNPP de uma $(k-1)$-partição para uma k-partição, Moffitt (2013) propõe um algoritmo baseado em resolução de subproblemas menores. Atualmente, o estado da arte para o MWNPP é o algoritmo *Sequential Number Partitioning* (SNP), apresentado em Korf et al. (2013), e o algoritmo *Cached Iterative Weakening* (CIW), apresentado em Schreiber e Korf (2014). Schreiber (2014) consolida uma análise completa a respeito desses algoritmos.

3.3 Revisão Literária – MDTWNPP

A segunda generalização do TWNPP é o *Multidimensional Two-Way Number Partitioning Problem* (MDTWNPP). Esta variante considera uma sequência V de vetores, não mais números, de dimensão m. Esta generalização é proposta inicialmente em Kojić (2010). Neste mesmo artigo, um modelo de programação linear inteira para o MDTWNPP é proposto e solucionado usando CPLEX. Este é o único método exato para este problema proposto até o presente momento. Pop e Matei (2013b) também aborda o MDTWNPP, mas usando meta-heurísticas populacionais, como Algoritmo Memético (MA) e Algoritmo Genético (GA) para sua solução.

Já Kratica et al. (2014) apresenta implementações das meta-heurísticas *Variable Neighborhood Search* (VNS) e *Electromagnetism-like* (EM) para a solução do MDTWNPP e compara os resultados encontrados com os apresentados em Kojić (2010) e Pop e Matei (2013b). Os resultados mostram que a meta-heurística EM possui desempenho levemente superior às demais.

Outro importante artigo abordando o MDTWNPP é Rodriguez et al. (2017), em que soluciona-se este problema usando uma meta-heurística híbrida entre *Greedy Randomized Adaptive Search Procedure* (GRASP) e *Exterior Path-relinking*. Os resultados obtidos, a partir do mesmo conjunto de instâncias usado em Kojić (2010), Pop e Matei (2013b) e Kratica et al. (2014), mostram a superioridade do procedimento proposto.

3.4 Revisão Literária – MDTWNPP

A terceira generalização do TWNPP é o MDMWNPP, proposto por Pop e Matei (2013b), uma união do MWNPP e do MDTWNPP. O MDMWNPP aparece originalmente em Pop e Matei (2013b), com a resolução dos casos *three-way*, buscando uma 3-partição, e *four-way*, buscando uma 4-partição, usando MA. Deve-se ressaltar que este problema ainda é pouco estudado na literatura e a primeira formulação proposta para este problema foi desenvolvida nesta Tese, com a publicação do artigo (Faria et al., 2021).

Um trabalho com maiores esclarecimentos sobre o MDMWNPP e suas principais propriedades encontra-se em Faria et al. (2019a).

3.5 Revisão Literária – Sumário

Formulações matemáticas são muito empregadas na solução de problemas gerais. Este fato se deve à existência de muitos algoritmos e técnicas para resolução de problemas de programação inteira ou binária (*0-1 Integer Programming Problem* – IP), que também é um problema NP-completo listado em Karp (1972). Provavelmente, a redução polinomial mais conhecida e feita nos estudos publicados e livros acadêmicos seja a redução ao IP. Na maioria dos livros de Otimização Combinatória, encontra-se uma série de problemas e seus respectivos modelos matemáticos com restrições lineares, como, por exemplo, em Ausiello et al. (2003a). Também existem muitos livros, como Williams (2013), ensinando como construir um modelo matemático para um posterior uso de bibliotecas computacionais dedicadas à resolução de problemas de programação matemática.

Dado um modelo matemático para um problema, pode-se usar uma técnica de ramificação por inserção de Restrições do tipo *Local Branching*, proposta em Fischetti e Lodi (2003). Essas restrições são definidas a partir de uma solução inicial x^s, e determinam o quão próxima ou distante uma solução factível pode estar de x^s, criando, assim, uma região factível local baseada em um subproblema mais simples. Tal técnica possibilita muitas variantes, exatas e aproximadas, para resolver um problema de otimização combinatória. Uma delas é o *Variable Neighborhood Descent Branching* – (VNDB) que define suas vizinhanças usando Restrições *Local Branching*. Esta variante está descrita em Maniezzo et al. (2010). Na verdade, o algoritmo proposto em Fischetti e Lodi (2003) se assemelha bastante ao algoritmo de planos de corte proposto em Gomory (1960). A ideia desse procedimento é remover grandes partes do espaço de busca até que a região factível fique vazia e a solução ótima guardada em uma variável auxiliar do algoritmo.

A Tabela 3.1 resume esta revisão bibliográfica para o MDMWNPP. **Dim.** é a dimensão dos vetores a serem particionados, multidimensionais ou unidimensionais, $m \geq 1$. **Part.** é a cardinalidade da partição, $k \geq 2$.

3.6 Publicações

Os artigos derivados desta pesquisa estão listados abaixo com uma descrição sobre suas contribuições. Todos os artigos são fortemente relacionados ao MDMWNPP. As publicações iniciais resultaram numa construção de longo prazo para esclarecer definições, propriedades e problemas relacionados. As publicações finais são a consolidação das contribuições da pesquisa:

(i) Comparação entre Heurísticas Construtivas e ILS na Solução do Problema da k-partição de Números (Faria et al., 2017)

O artigo propõe a aplicação da meta-heurística *Iterated Local Search* (ILS) adaptada para ao MWPPN. O desempenho do método é satisfatório para problemas com $k \leq 6$ quando comparada aos métodos construtivos intensificados por uma enumeração inicial de elementos aplicada à heurística construtiva *Longest Processing Time* (LPT).

Tabela 3.1: Sumário da Bibliografia do MDMWNPP.

Item	Autor	Título	Ano	Dim.	Part.	Método	Tipo
1.	Kojić (2010)	Integer linear programming model for multidimensional two-way number partitioning problem	2010	MD	2W	ILP	ILP-Exato
2.	Pop e Matei (2013a)	A genetic algorithm approach for the multidimensional two-way number partitioning problem	2013	MD	2W	GA	Meta
3.	Pop e Matei (2013b)	A memetic algorithm approach for solving the multidimensional multi-way number partitioning problem	2013	MD	kW	MA	Meta
4.	Kratica et al. (2014)	Two metaheuristic approaches for solving multidimensional two-way number partitioning problem	2014	MD	2W	VNS e EM	Meta
5.	Rodriguez et al. (2017)	GRASP with exterior path-relinking and restricted local search for the multidimensional two-way number partitioning problem	2017	MD	2W	GRASP	Meta
6.	Hacibeyoglu et al. (2018)	A Comparative Analysis of Metaheuristic Approaches for Multidimensional Two-Way Number Partitioning Problem	2018	MD	2W	GA, SA, MBO e CSA	Meta
7.	Santucci et al. (2019)	Algebraic Differential Evolution and VNS for the multidimensional two way number_partitioning problem	2019	MD	2W	ADE + VNS	Meta
8.	Faria et al. (2019a)	A Variable Neighborhood Search Approach for Solving the Multidimensional Multi-Way Number Partitioning Problem	2019	MD	kW	VNS	Meta
9.	Faria et al. (2021)	A Mixed-Integer Linear Programming model to solve the Multidimensional Multi-Way Number Partitioning Problem	2021	MD	kW	ILP	ILP-Exato
10.	Santucci et al. (2021)	An improved memetic algebraic differential evolution for solving the multidimensional two-way number partitioning problem	2021	MD	2W	ADE + MA	Meta
11.	Nikolic et al. (2022)	New mixed-integer linear programming model for solving the multidimensional multi-way number partitioning problem.	2022	MD	kW	ILP	ILP-Exato
12.	Djukanović et al. (2022)	Self-adaptive cmsa for solving the multidimensional multi-way number partitioning problem.	2022	MD	kW	CMSA, VNS e ILP	Meta/Exato

As estruturas de vizinhanças usadas do ILS fundamentam boa parte dos algoritmos *Variable Neighborhood Descent* (VND) e *Variable Neighborhood Search* (VNS) propostos nos artigos Faria et al. (2018a, 2019a).

(ii) Análise de Modelos Matemáticos para o Problema da Múltipla Partição de Números (Faria et al., 2018b)

O artigo apresenta quatro variantes de modelos matemáticos para o MWNPP. Os resultados descritivos mostram que algumas variantes das formulações propostas são capazes de encontrar soluções ótimas em tempo limitado sem esgotar a memória computacional. Os testes estatísticos realizados demonstram diferenças significativas entre os modelos matemáticos. O fator que mais influencia a qualidade da solução é a redução das variáveis com base na Equação (4.42):

$$x_{k,i} = 1 - \sum_{j=1}^{k-1} x_{j,i}, \quad \forall i \in I_n.$$

Este modelo refinado é usado no artigo Faria et al. (2019b). As deduções das restrições e tratamentos de variáveis desenvolvidos foram importantes para o desenvolvimento do artigo Faria et al. (2021).

(iii) *Variable Neighborhood Descent applied to Multi-way Number Partitioning Problem* (Faria et al., 2018a)

O artigo apresenta um algoritmo VND adaptado para a solução do MWNPP. O VND tem desempenho satisfatório em instâncias com $k \leq 6$. Realiza-se um estudo comparativo com outros dois algoritmos, KKH e LPT, utilizando instâncias geradas aleatoriamente e valores de funções objetivos. Os testes estatísticos mostram que os resultados do VND proposto são significativamente melhores do que os resultados dos métodos construtivos da literatura.

O VND apresentado é continuidade do ILS e generaliza as vizinhanças usadas em Faria et al. (2017), encontrando a melhor solução local em tempo polinomial.

(iv) *A Variable Neighborhood Search Approach for Solving the Multidimensional Multi-Way Number Partitioning Problem* (Faria et al., 2019a)

O artigo apresenta uma implementação da meta-heurística VNS para solução do MDMWNPP. O VNS proposto para a resolução do MDMWNPP é aplicado para resolução de instâncias com $k \in \{3, 4\}$. É realizado um estudo comparativo entre resultados encontrados a partir do VNS proposto e do Algoritmo Memético, *Memetic Algorithm* (MA), funcionando no mesmo intervalo de tempo e ajustado de acordo com métricas de processamento computacional. Embora os resultados médios sejam diferentes, os testes estatísticos mostram que os resultados do VNS proposto não são significativamente melhores do que o MA num conjunto de instâncias analisadas.

O VNS é uma continuação da ideia de Faria et al. (2018a). Este artigo representou uma fase de transição da pesquisa na direção dos métodos baseados em programação matemática devido ao insucesso das meta-heurísticas aplicadas ao MDMWNPP e à ausência de métodos exatos na literatura.

Esse artigo é citado e parcialmente reproduzido em Djukanović et al. (2022) onde também exibe resultados de menor desempenho quando comparado aos demais algoritmos da literatura.

(v) *Variable Neighborhood Descent Branching applied to the Multi-Way Number Partitioning Problem* (Faria et al., 2019b)

O artigo propõe uma aplicação do método *Local Branching* dentro de uma busca similar ao VND para resolver o MWNPP, chamado VNDB. Neste artigo, é apresentada uma decomposição do MWNPP em $n-1$ subproblemas usando restrições baseadas na distância *hamming*. Essa decomposição justifica a estrutura de vizinhança utilizada no VNDB. Os resultados mostram que, exceto para $k=4$, não há diferença estatística significativa entre o valor da função objetivo das soluções obtidas pelo VNDB e pela resolução da formulação matemática proposta em Faria et al. (2018b) utilizando um resolvedor com tempo limite. O tempo utilizado por ambos os métodos para atingir resultados similares é significativamente diferente, sendo favorável ao VNDB.

Para aplicar ideias do método *Local Branching* é necessário ter uma formulação matemática adequada. Nesse ponto, o foco foi a dedução de uma formulação matemática para o MDMWNPP apresentada em Faria et al. (2021) e subsequente aplicação do método *Local Branching* ao MDMWNPP.

Esse artigo é citado em Rodrigues et al. (2022) como uma estrutura de restrições locais mais bem ajustada à modelos matemáticos contendo a família de restrições:

$$\sum_{j \in I_k} x_{ji} = 1, \quad \forall i \in I_n.$$

(vi) *A Mixed-Integer Linear Programming model to solve the Multidimensional Multi-Way Number Partitioning Problem* (Faria et al., 2021)

O artigo propõe um MILP para resolver o MDMWNPP. Após a introdução do modelo proposto, são apresentadas algumas propriedades e são deduzidas as principais restrições. Para validar a formulação matemática proposta, compara-se o modelo proposto para $k=2$ e o apresentado em Kojić (2010) para a solução do MDTWNPP. O modelo proposto mostrou resultados competitivos e resolveu instâncias do problema para valores de $k \in \{3,4\}$.

Esse é o artigo mais citado desta pesquisa e foi completamente reproduzido em Nikolic et al. (2022), onde é proposta uma formulação matemática igual a descrita no modelo (4.31) (já presente no Projeto de Tese, mas não publicado). Os resultados desse artigo usando o modelo (4.23) são melhores que os do artigo original Faria et al. (2021). A segunda citação deste artigo foi feita por Santucci et al. (2021), que apenas se refere ao modelo (4.23) como um método exato para a solução do MDTWNPP na parte de revisão. A terceira citação foi feita por Djukanović et al. (2022), que usa os resultados dos experimentos do modelo (4.23) numa revisão que une métodos meta-heurísticos e exatos.

Capítulo 4

Formulações Matemáticas para o MDMWNPP

Este capítulo apresenta o desenvolvimento de modelos matemáticos para o Problema da Múltipla Partição de Números Multidimensional (*Multidimensional Multi-Way Number Partitioning Problem* – MDMWNPP). O caso específico de um modelo matemático para um problema NP-completo é interessante tanto do ponto de vista prático, para permitir sua resolução através de métodos de resolução baseados em programação matemática, quanto do ponto de vista teórico, para levar a redução do MDMWNPP ao problema de otimização linear inteira mista (MILP). Todas as técnicas usadas para desenvolver MILPs neste capítulo estão listadas em Williams (2013) e Hooker (2011). Este capítulo tem a seguinte organização: a Seção 4.1 mostra a equivalência entre a função objetivo exibida em Pop e Matei (2013b) e a função objetivo da Definição 2.5.1. As Seções 4.2 e 4.3 apresentam deduções das restrições que caracterizam o MILP associado ao MDMWNPP proposto neste trabalho. A Seção 4.4 relaciona o modelo matemático proposto com o modelo apresentado por Kojić (2010). A Seção 4.5 mostra como remover simetrias do modelo matemático proposto com uma nova família de restrições.

4.1 Análise da Função Objetivo

A função objetivo mostrada em Pop e Matei (2013b) e a função objetivo adotada neste trabalho são, na verdade, duas formas diferentes de calcular o diâmetro de um multiconjunto de vetores.

Antes de avançar nesta descrição, é necessário introduzir uma notação. Seja $V = \{v_i \in \Re^m \ : \ i \in I_n\}$. A função $g : \mathcal{P}(I_n) \to \Re^m$ é definida como:

$$X \in \mathcal{P}(I_n) \quad : \quad g(X) = \sum_{i \in X} v_i \tag{4.1}$$

Para se fazer referência à l-ésima coordenada de $g(X)$, usa-se a notação $g_l(X)$.

A função objetivo mostrada em Pop e Matei (2013b) é:

$$f(\{A_j\}_{j \in I_k}) = \max_{l \in I_m} \left\{ \left| \max_{j_1 \in I_k} \left\{ \sum_{i \in A_{j_1}} v_{il} \right\} - \min_{j_2 \in I_k} \left\{ \sum_{i \in A_{j_2}} v_{il} \right\} \right| \right\} \tag{4.2}$$

que pode ser reescrita, seguindo a notação apresentada pela Expressão (4.1), como:

$$f(\{A_j\}_{j\in I_k}) = \max_{l\in I_m}\left\{\left|\max_{j\in I_k} g_l(A_j) - \min_{j'\in I_k} g_l(A_{j'})\right|\right\}. \tag{4.3}$$

Primeiramente, é possível retirar o módulo da expressão sem qualquer perda, pois, fixando-se l, tem-se que

$$\max_{j\in I_k} g_l(A_j) - \min_{j'\in I_k} g_l(A_{j'}) \geq 0.$$

A ideia de denotar a função objetivo apenas como diâmetro do multiconjunto de vetores resultantes permite uma interpretação clara, que se aplica a todas as variantes do NPP listadas na literatura até o momento (TWNPP, MWNPP, MDTWNPP e MDMWNPP).

A Proposição 4.1.1 e o Corolário 4.1.1 mostram a dedução de um resultado conhecido a ser aplicado na demonstração de equivalência das duas expressões na Proposição 4.1.2 relacionada ao diâmetro induzido pela norma infinita.

Proposição 4.1.1. *Seja I um intervalo real limitado. Então*

$$\max_{x,y\in I}|x-y| = \max_{z\in I} z - \min_{z\in I} z \tag{4.4}$$

Prova: Como $x, y \in I$, tem-se que:

$$\min_{z\in I} z \leq x \leq \max_{z\in I} z \tag{4.5}$$

$$\min_{z\in I} z \leq y \leq \max_{z\in I} z \tag{4.6}$$

Multiplicando-se a Expressão (4.6) por -1:

$$-\max_{z\in I} z \leq -y \leq -\min_{z\in I} z \tag{4.7}$$

Somando-se as expressões (4.5) e (4.7), o resultado é

$$\min_{z\in I} z - \max_{z\in I} z \leq x - y \leq \max_{z\in I} z - \min_{z\in I} z \tag{4.8}$$

Então:

$$|x-y| \leq \max_{z\in I} z - \min_{z\in I} z \tag{4.9}$$

Portanto:

$$\max_{x,y\in I}|x-y| \leq \max_{z\in I} z - \min_{z\in I} z \tag{4.10}$$

A igualdade ocorre no valor máximo. Verifica-se este fato fixando-se $x = \max_{z\in I} z$ e $y = \min_{z\in I} z$. ■

Corolário 4.1.1. *Seja I um intervalo real limitado e considere uma sequência discreta $\{a_i\}_{i\in I_n} \subset I$. Então:*

$$\max_{i,j\in I_n}|a_i - a_j| = \max_{i\in I_n} a_i - \min_{j\in I_n} a_j \tag{4.11}$$

Proposição 4.1.2. *O diâmetro induzido pela norma infinita pode ser calculado por:*

$$diam_\infty(\{g(A_j)\}_{j\in I_k}) = \max_{l\in I_m}\{\max_{j\in I_k} g_l(A_j) - \min_{j'\in I_k} g_l(A_{j'})\} \tag{4.12}$$

Prova: O cálculo da norma infinita pode ser dado por:

$$\begin{aligned}\operatorname{diam}_\infty(\{g(A_j)\}_{j\in I_k}) &= \max_{j,j'\in I_k} \|g(A_j) - g(A_{j'})\|_\infty \\ &= \max_{j,j'\in I_k}\max_{l\in I_m} |g_l(A_j) - g_l(A_{j'})| \\ &= \max_{l\in I_m}\max_{j,j'\in I_k} |g_l(A_j) - g_l(A_{j'})| \end{aligned} \tag{4.13}$$

Pelo Corolário 4.1.1,

$$\max_{l\in I_m}\max_{j,j'\in I_k} |g_l(A_j) - g_l(A_{j'})| = \max_{l\in I_m}\{\max_{j\in I_k} g_l(A_j) - \min_{j'\in I_k} g_l(A_{j'})\} \tag{4.14}$$

■

4.2 Dedução das Restrições 1

Considere a função objetivo (2.5) da Definição 2.5.1. A partir dela, é possível deduzir os conjuntos de restrições para definir um MIP (*Mixed Integer Programming*).

Seja $y \geq 0$ o diâmetro a ser minimizado, então:

$$y = \max_{l\in I_m}\left\{\left|\max_{j_1,j_2\in I_k}\left\{\left|\sum_{i\in A_{j_1}} v_{il} - \sum_{i\in A_{j_2}} v_{il}\right|\right\}\right|\right\} \tag{4.15}$$

Pode-se deduzir, a partir da Expressão (4.15), que:

$$-y \leq \max_{j_1,j_2\in I_k}\left\{\left|\sum_{i\in A_{j_1}} v_{il} - \sum_{i\in A_{j_2}} v_{il}\right|\right\} \leq y, \quad \forall l \in I_m \tag{4.16}$$

e, da mesma maneira:

$$-y \leq \sum_{i\in A_{j_1}} v_{il} - \sum_{i\in A_{j_2}} v_{il} \leq y, \quad \forall l \in I_m, \quad \forall (j_1, j_2) \in I_k \times I_k \tag{4.17}$$

Seja x_{ji} uma variável binária tal que $x_{ji} = 1$, se o índice i do vetor v_i está em A_j, e $x_{ji} = 0$, caso contrário. Pode-se reescrever a Expressão (4.17) da seguinte maneira:

$$-y \leq \sum_{i=1}^{n} v_{il}x_{j_1i} - \sum_{i=1}^{n} v_{il}x_{j_2i} \leq y, \quad \forall l \in I_m, \quad \forall (j_1, j_2) \in I_k \times I_k \tag{4.18}$$

Colocando v_{il} em evidência, tem-se

$$-y \leq \sum_{i=1}^{n} v_{il}(x_{j_1i} - x_{j_2i}) \leq y, \quad \forall l \in I_m, \quad \forall (j_1, j_2) \in I_k \times I_k \tag{4.19}$$

Existe a possibilidade de se reduzir o número de restrições da Expressão (4.19) para menos da metade, sem perda de generalidade. Para isso, primeiro, quando $j_1 = j_2$, as desigualdades da Expressão (4.19) são sempre válidas, pois, como $y \geq 0$, então $-y \leq 0 \leq y$, ou seja:

$$\sum_{i=1}^{n} v_{il}(x_{j_1 i} - x_{j_1 i}) = 0 \tag{4.20}$$

Segundo, a validade da família de desigualdades dada pela Expressão (4.19) para o par (j_1, j_2) implica na validade para o par (j_2, j_1), devido à simetria. Multiplicando-se a Expressão (4.19) por (-1), tem-se que:

$$-y \leq \sum_{i=1}^{n} v_{il}(x_{j_1 i} - x_{j_2 i}) \leq y \Leftrightarrow -y \leq \sum_{i=1}^{n} v_{il}(x_{j_2 i} - x_{j_1 i}) \leq y \tag{4.21}$$

As restrições para garantir que uma solução seja uma k-partição são bem conhecidas e são dadas por:

$$\sum_{j=1}^{k} x_{ji} = 1, \ \forall i \in I_n \tag{4.22a}$$

$$\sum_{i=1}^{n} x_{ji} \geq 1, \ \forall j \in I_k \tag{4.22b}$$

O conjunto de restrições (4.22a) mantém as propriedades (ii) e (iii) da Definição 2.1.2 e o conjunto de restrições (4.22b) garante a propriedade (i) desta mesma definição.

Depois da dedução das restrições principais, chega-se ao modelo matemático proposto, que tem $mk(k-1)+n+k$ restrições e $nk+1$ variáveis, sendo nk variáveis binárias e 1 variável real não negativa. Este modelo é dado por:

$$\begin{aligned}
\min \quad & y && (4.23a)\\
\text{suj. a} \quad & \sum_{j=1}^{k} x_{ji} = 1, \ \forall i \in I_n && (4.23b)\\
& \sum_{i=1}^{n} x_{ji} \geq 1, \quad \forall j \in I_k && (4.23c)\\
& \sum_{i=1}^{n} v_{il}(x_{j_1 i} - x_{j_2 i}) \leq y, \quad \forall (1 \leq j_1 < j_2 \leq k), \forall l \in I_m && (4.23d)\\
& \sum_{i=1}^{n} v_{il}(x_{j_2 i} - x_{j_1 i}) \leq y, \quad \forall (1 \leq j_1 < j_2 \leq k), \forall l \in I_m && (4.23e)\\
& y \in \mathbb{R}_+ && (4.23f)\\
& x_{ji} \in \{0, 1\}, \quad \forall (j, i) \in I_k \times I_n && (4.23g)
\end{aligned}$$

A Expressão (4.23a) representa a função objetivo do problema e é o diâmetro do conjunto $\{g(A_j)\}_{j \in I_k}$. O objetivo do problema é minimizar y. As Equações (4.23b)

garantem que as partes formam uma coleção de subconjuntos mutuamente disjuntos. As Inequações (4.23c) asseguram que cada parte j é não vazia. As Inequações (4.23d) e (4.23e) calculam a diferença entre cada par de vetores resultantes da soma dos elementos de partes distintas. Cada uma destas diferenças é limitada pela variável y, definida pela relação (4.23f). A Relação de pertinência (4.23g) indica que $x_{ji} = 1$, se o vetor de índice i do conjunto V pertence à parte de índice j, e $x_{ji} = 0$, caso contrário.

4.3 Dedução das Restrições 2

Existe uma segunda forma de encontrar um conjunto de restrições diferente do apresentado na Seção 4.2, usando, como base, a função objetivo (4.2). A dedução é similar à apresentada na Seção 4.2.

Seja $y \geq 0$ o diâmetro a ser minimizado. Omitindo o módulo da Função (4.2), tem-se:

$$y = \max_{l \in I_m} \left\{ \max_{j_1 \in I_k} \left\{ \sum_{i \in A_{j_1}} v_{il} \right\} - \min_{j_2 \in I_k} \left\{ \sum_{i \in A_{j_2}} v_{il} \right\} \right\} \tag{4.24}$$

Defina dois conjuntos de variáveis reais $\{t_{1l}\}_{l \in I_m}$ e $\{t_{2l}\}_{l \in I_m}$ tais que:

$$t_{1l} = \min_{j_2 \in I_k} \left\{ \sum_{i \in A_{j_2}} v_{il} \right\} \tag{4.25}$$

$$t_{2l} = \max_{j_1 \in I_k} \left\{ \sum_{i \in A_{j_1}} v_{il} \right\} \tag{4.26}$$

Assim, da Equação (4.24), tem-se que:

$$y = \max_{l \in I_m} \{t_{2l} - t_{1l}\} \tag{4.27}$$

ou seja:

$$t_{2l} - t_{1l} \leq y, \quad \forall l \in I_m \tag{4.28}$$

Da mesma maneira que mostrado na Seção 4.2, deve-se definir x_{ji} como uma variável binária tal que $x_{ji} = 1$, se o índice i do vetor v_i está em A_j, e $x_{ji} = 0$, caso contrário. Desenvolve-se, a partir das Expressões (4.25) e (4.26), as restrições:

$$t_{1l} \leq \sum_{i=1}^{n} v_{il} x_{ji}, \quad \forall j \in I_k \tag{4.29}$$

$$t_{2l} \geq \sum_{i=1}^{n} v_{il} x_{ji}, \quad \forall j \in I_k \tag{4.30}$$

Nota-se que, fixada uma coordenada l, as variáveis t_{2l} e t_{1l} são limitantes superiores e inferiores, respectivamente, de $\{g_l(A_j)\}_{j \in I_k}$.

Este outro conjunto de restrições principais tem características distintas daquele mostrado na Seção 4.2, mas garante todas as propriedades do MDMWNPP. O número de restrições é igual a $(2k + 1)m + n + k$ e o número de variáveis é igual a

$nk+2m+1$, sendo uma variável real não negativa, $2m$ variáveis reais e nk variáveis binárias. O modelo é, portanto, dado por:

$$\begin{aligned}
\min \quad & y && (4.31a)\\
\text{suj. a} \quad & \sum_{i=1}^{k} x_{ji} = 1, \quad \forall j \in I_n && (4.31b)\\
& \sum_{i=1}^{n} x_{ji} \geq 1, \quad \forall j \in I_k && (4.31c)\\
& \sum_{i=1}^{n} v_{il} x_{ji} \geq t_{1l}, \quad \forall j \in I_k, \quad \forall l \in I_m && (4.31d)\\
& \sum_{i=1}^{n} v_{il} x_{ji} \leq t_{2l}, \quad \forall j \in I_k, \quad \forall l \in I_m && (4.31e)\\
& t_{2l} - t_{1l} \leq y, \quad \forall l \in I_m && (4.31f)\\
& y \in \mathbb{R}_+ && (4.31g)\\
& t_{2l}, t_{1l} \in \Re, \quad \forall l \in I_m && (4.31h)\\
& x_{ji} \in \{0,1\}, \quad \forall (j,i) \in I_k \times I_n && (4.31i)
\end{aligned}$$

A função objetivo é dada pela Variável (4.31a) e representa o diâmetro do conjunto $\{g(A_j)\}_{j\in I_k}$. O objetivo é minimizar y. As Restrições (4.31b) e (4.31c) definem uma k-partição. As Inequações (4.31d) e (4.31e) associam o limite superior e inferior de cada coordenada às variáveis t_{2l} e t_{1l}. O conjunto de Restrições (4.31f) limita todas as diferenças entre o maior e o menor valor de coordenada à variável y. A Relação (4.31g) define o diâmetro de $\{g(A_j)\}_{j\in I_k}$ como um número positivo. A Relação (4.31h) define os limitantes do conjunto $\{g_l(A_j)\}_{j\in I_k}$. A Relação (4.31i) indica que $x_{ji} = 1$, se o vetor de índice i do conjunto V pertence à parte de índice j, e $x_{ji} = 0$, caso contrário.

4.4 Comparação entre a Formulação Matemática do MDTWNPP e a Proposta

De acordo com Kojić (2010), o MDTWNPP é definido da seguinte maneira.

Definição 4.4.1. *Seja $V = \{w_i\}_{i\in I_n}$ uma sequência de vetores tal que $n = |V|$ e $w_i \in \Re^m$, $\forall\, i \in I_n$. O objetivo do MDTWNPP é particionar I_n em dois subconjuntos A_1 e A_2, sendo $A_1 \cup A_2 = I_n$ e $A_1 \cap A_2 = \emptyset$, minimizando a função:*

$$t = \max_{l\in I_m} \left| \sum_{i\in A_1} w_{il} - \sum_{i\in A_2} w_{il} \right| \tag{4.32}$$

Nota-se que não está citado no enunciado que A_1 e A_2 devem ser não vazios, apesar do termo "particionar" ser usado como nos livros clássicos Kreher e Stinson (1998) e Nijenhuis e Wilf (2014). Teoricamente, isso causa problemas na construção do modelo matemático, pois o enunciado se torna impreciso.

O MILP proposto por Kojić (2010) para o problema representado na Definição 4.4.1 é construído a partir da função objetivo (4.32). Assim, seja a expressão:

$$s_l = \sum_{i=1}^{n} w_{il}, \ \forall l \in I_m.$$

Defina x_i como uma variável binária, na forma:

$$x_i = \begin{cases} 1, & \text{se } \ i \in A_1 \\ 0, & \text{se } \ i \in A_2 \end{cases} \tag{4.33}$$

A variável t denota a distância induzida pela norma infinita entre $g(A_1)$ e $g(A_2)$. O modelo de otimização linear inteira proposto em Kojić (2010) é dado por:

$$\begin{aligned}
(MOD_{Kojic}) \quad \min \quad & t && \text{(4.34a)}\\
\text{suj. a} \quad & -0.5t + \sum_{i=1}^{n} w_{il}x_i \leq 0.5s_l, \ \forall l \in I_m && \text{(4.34b)}\\
& 0.5t + \sum_{i=1}^{n} w_{il}x_i \geq 0.5s_l, \ \forall l \in I_m && \text{(4.34c)}\\
& x_i \in \{0,1\}, \ \forall i \in I_n && \text{(4.34d)}\\
& t \in \Re && \text{(4.34e)}
\end{aligned}$$

A função objetivo (4.34a) representa a distância entre $g(A_1)$ e $g(A_2)$. As Restrições (4.34b) e (4.34c) derivam da linearização da função (4.32). O modelo (4.34) tem n variáveis binárias, uma variável real e $2m$ restrições.

Agora, usando o modelo matemático (4.23) e assumindo $k = 2$, o resultado encontrado é:

$$\begin{aligned}
\min \quad & y && \text{(4.35a)}\\
\text{suj. a} \quad & x_{1,i} + x_{2,i} = 1, \quad \forall i \in I_n && \text{(4.35b)}\\
& \sum_{i=1}^{n} x_{1,i} \geq 1 && \text{(4.35c)}\\
& \sum_{i=1}^{n} x_{2,i} \geq 1 && \text{(4.35d)}\\
& \sum_{i=1}^{n} v_{il}(x_{1,i} - x_{2,i}) \leq y, \quad \forall l \in I_m && \text{(4.35e)}\\
& \sum_{i=1}^{n} v_{il}(x_{2,i} - x_{1,i}) \leq y, \quad \forall l \in I_m && \text{(4.35f)}\\
& y \in \mathbb{R}_+ && \text{(4.35g)}\\
& x_{ji} \in \{0,1\}, \quad \forall (j,i) \in I_2 \times I_n && \text{(4.35h)}
\end{aligned}$$

Considerando $s_l = \sum_{i=1}^{n} v_{il}$ e fazendo a substituição $x_{1,i} = 1 - x_{2,i}$, baseada nas restrições (4.35b), então tem-se o modelo:

$$\min \quad y \tag{4.36a}$$

$$\text{suj. a} \quad \sum_{i=1}^{n} x_{1,i} \geq 1 \tag{4.36b}$$

$$\sum_{i=1}^{n}(1 - x_{1,i}) \geq 1 \implies \sum_{i=1}^{n} x_{1,i} \leq n - 1 \tag{4.36c}$$

$$\sum_{i=1}^{n} v_{il}(2x_{1,i} - 1) \leq y \implies -\frac{y}{2} + \sum_{i=1}^{n} v_{il}x_{1,i} \leq \frac{s_l}{2}, \ \forall l \in I_m \tag{4.36d}$$

$$\sum_{i=1}^{n} v_{il}(1 - 2x_{1,i}) \leq y \implies \frac{y}{2} + \sum_{i=1}^{n} v_{il}x_{1,i} \geq \frac{s_l}{2}, \ \forall l \in I_m \tag{4.36e}$$

$$y \in \mathbb{R}_+ \tag{4.36f}$$

$$x_{1,i} \in \{0, 1\}, \ \forall i \in I_n \tag{4.36g}$$

O modelo matemático (4.36) é diferente do modelo (4.34) apenas devido às restrições (4.36b) e (4.36c). Estas restrições são úteis para garantir a propriedade (i) da Definição 2.1.2, impedindo que uma solução tenha todos os elementos da sequência contidos em uma mesma parte ou a existência de uma parte vazia, já que o modelo considera $v_i \in \Re^m$. Fazendo as mesmas manipulações para o modelo matemático (4.31), obtém-se o mesmo resultado.

É possível mostrar exemplos de instâncias com vetores em $\Re^m$, contendo números negativos, em que o modelo (4.34), sem as restrições (4.36b) e (4.36c), gera uma solução ótima errada. Este caso é mostrado no Exemplo 4.4.1.

Exemplo 4.4.1. *Seja* $V = \{(3,3),(-1,-1),(-1,-1),(-1,-1)\}$, *sendo* $k = 2$, $n = 4$ *e* $m = 2$. *A solução ótima é* $\{(3,3),(-1,-1),(-1,-1)\},\{(-1,-1)\}$. *No entanto, o resultado encontrado pelo modelo matemático* (4.34) *é outro, qual seja,* $\{(3,3),(-1,-1),(-1,-1),(-1,-1)\},\{\}$. *Nota-se que o resultado contém uma parte vazia e não é uma* 2*-partição. Na verdade, existem diversos contraexemplos em que esta formulação falha, como:*

$$V = \begin{cases} v_i &= (-1,-1), \quad \forall i \in I_{n-1} \\ v_n &= (n-1, n-1) \end{cases}$$

$$V = \{(1,3),(-1,-2),(0-1),(6,5),(-3-3),(-3-2)\}$$

$$V = \{(a_1,0,0,0),(0,a_2,0,0),(0,0,a_3,0),(0,0,0,a_4)\}, \quad a_1,a_2,a_3,a_4 \in \Re$$

Se a soma dos vetores é igual ao vetor nulo, isto faz com que a função objetivo seja zero, pois a primeira solução inteira encontrada é sempre $x_i = 0, \forall i \in I_n$, *satisfazendo as restrições* (4.36d) *e* (4.36e) *e atingindo um critério de otimalidade do MIP, mesmo que uma das partes esteja vazia. O último contraexemplo é mais complexo, porém, todas as suas possíveis partições têm o mesmo valor de função objetivo. Neste caso, a solução padrão também é* $x_i = 0, \forall i \in I_n$, *pois esse valor é a relaxação inicial com a função objetivo igual a* $\max\{|a_i|\}$. □

Observando as instâncias usadas em Kojić (2010), nota-se que todos os números presentes são estritamente positivos e todas as coordenadas dos vetores são tais que $v_{il} > 0$. Esta hipótese será considerada para esclarecer se, apesar dos contraexemplos acima, os trabalhos que usam este modelo matemático sempre encontram particionamentos com partes não vazias nestas condições.

Proposição 4.4.1. *Sejam $x, y \in \Re_+^m$ vetores com coordenadas estritamente positivas. Então, $||x - y||_\infty < ||x + y||_\infty$.*

Prova: Considere uma coordenada fixa x_l e y_l dos vetores dados. Deseja-se provar que $|x_l - y_l| < |x_l + y_l|$, $\forall l \in I_m$, pois, logo em seguida, aplicando o máximo em ambos os lados, conclui-se a prova.

Sendo y_l e x_l estritamente positivos por hipótese, usa-se a definição de módulo: se $x_l \geq y_l$, então $|x_l - y_l| = (x_l - y_l)$. Portanto:

$$|x_l - y_l| = (x_l - y_l) < (x_l - y_l + 2y_l) = (x_l + y_l) = |x_l + y_l|.$$

Em outro caso, se $x_l < y_l$, então:

$$|x_l - y_l| = (y_l - x_l) < (y_l - x_l + 2x_l) = (y_l + x_l) = |y_l + x_l|.$$

Com a união dos dois casos acima, conclui-se que:

$$|x_l - y_l| < |x_l + y_l|, \quad \forall l \in I_m.$$

Aplicando o máximo em ambos os lados:

$$\max_{l \in I_m} \{|x_l - y_l|\} < \max_{l \in I_m} \{|x_l + y_l|\}.$$

Portanto:

$$||x - y|\,|_\infty < ||x + y|\,|_\infty.$$

■

Seja agora $\{A_1, A_2\}$ uma 2-partição de I_n. Pode-se usar a Proposição 4.4.1 para concluir que:

$$\left\| \sum_{i \in A_1} v_i - \sum_{i \in A_2} v_i \right\|_\infty < \left\| \sum_{i \in A_1} v_i + \sum_{i \in A_2} v_i \right\|_\infty = \left\| \sum_{i \in I_n} v_i \right\|_\infty. \tag{4.37}$$

Logo, as restrições (4.36c) e (4.36d) são dispensáveis nesse caso, pois qualquer 2-partição de I_n terá um valor de função objetivo estritamente menor que uma solução contendo todos os vetores em uma única parte, ao contrário de como ocorreu no Contraexemplo (4.4.1). Concluindo, o modelo apresentado em Kojić (2010) pode ser usado para comparação com o modelo proposto neste trabalho desde que a hipótese $v_{il} > 0$, $\forall (i, l) \in I_n \times I_m$, seja assumida.

4.5 Eliminação das Simetrias

A solução dos modelos matemáticos (4.23) e (4.31) implicam na solução do problema MDMWNPP, mas ainda há uma correspondência sobrejetora entre soluções factíveis desses modelos matemáticos e o conjunto das k-partições de I_n. Denota-se um modelo matemático $\mathcal{P}$ genérico, podendo ser os modelos (4.23) ou (4.31), pois as simetrias são iguais em ambos os modelos.

Primeiro, é importante notar que a ordem das partes não muda a solução. Portanto, toda permutação de linhas de uma solução representa a mesma k-partição. Uma ilustração é apresentada no Exemplo 4.5.1.

Exemplo 4.5.1. *Seja $k = 3$, $n = 5$ e $\{v_1, v_2, v_3, v_4, v_5\}$. Para qualquer valor que se queira atribuir aos v_i, as matrizes a seguir são soluções factíveis para o modelo matemático $\mathcal{P}$ e representam a mesma 3-partição $\{v_1, v_3\}, \{v_2, v_5\}, \{v_4\}$:*

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

A codificação clássica usada para k-partições é um vetor de tamanho n, cujas entradas são números de 1 até k. Mais precisamente, um vetor s tal que $s_i = j$ se $v_i \in A_j$. O vetor s é tal que $s_{i+1} \leq 1 + \max_{1 \leq h \leq i} s_h$ para evitar multiplicidade de soluções. A bijeção entre o conjunto das k-partições e o conjunto de todos os possíveis vetores s se encontra demonstrada em Orlov (2002), sendo:

(i) O índice 1 de v_1 deve estar sempre na parte 1;

(ii) Uma parte j sempre terá pelo menos um índice i menor que qualquer índice contido na parte $j + 1$. Isto vale para todo $j \in I_{k-1}$.

Portanto, as partes devem estar sempre ordenadas pelo valor de seu menor índice.

Exemplo 4.5.2. *Considere as seguintes codificações de uma 3-partição:*

$$[1, 1, 1, 3, 2] \text{ e } [1, 1, 1, 2, 3]$$

Observe que o primeiro vetor não satisfaz à propriedade (ii), enquanto o segundo vetor satisfaz. No entanto, note que a codificação $[1, 3, 1, 2, 3]$ corresponde à codificação $[1, 2, 1, 3, 2]$, representando a 3-partição $\{v_1, v_3\}, \{v_2, v_5\}, \{v_4\}$.

A partir deste fato, conclui-se que a propriedade desejada é que a matriz binária deve, além de satisfazer as restrições (4.22), ter a forma escada. Com o efeito da forma escada, pode-se fixar $x_{1,1} = 1$ como constante, excluindo uma variável e uma restrição do tipo (4.22a). Além disso, permite eliminar o triângulo inferior esquerdo de variáveis do modelo, na forma $x_{ji} = 0, \quad \forall j > i$, como no Exemplo 4.5.3 a seguir.

Exemplo 4.5.3.

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ & 0 & 0 & 0 & 1 \\ & & 0 & 1 & 0 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ & 0 & 1 & 1 & 0 \\ & & 0 & 0 & 1 \end{pmatrix}$$

Para deduzir as restrições que só permitem matrizes binárias na forma escada, usa-se a mesma ideia da redução do Problema da Satisfatibilidade Boleana (*Boolean Satisfiability Problem* – SAT) ao ILP vista em Cook (1971) e Karp (1972). Primeiro, escreve-se a propriedade como uma expressão lógica, para depois convertê-la em uma restrição com variáveis binárias:

$$x_{j+1,i} \implies \bigvee_{h=1}^{i-1} x_{j,h}, \quad \forall i \in I_n \setminus \{1\}, \quad \forall j \in I_{k-1} \tag{4.38}$$

Usando a implicação material $P \to Q \Leftrightarrow \neg P \vee Q$, tem-se:

$$\neg x_{j+1,i} \vee \bigvee_{h=1}^{i-1} x_{j,h}, \quad \forall i \in I_n \setminus \{1\}, \quad \forall j \in I_{k-1} \tag{4.39}$$

o que se transforma no conjunto de restrições:

$$(1 - x_{j+1,i}) + \sum_{h=1}^{i-1} x_{j,h} \geq 1, \quad \forall i \in I_n \setminus \{1\}, \quad \forall j \in I_{k-1} \tag{4.40}$$

Simplificando:

$$x_{j+1,i} - \sum_{h=1}^{i-1} x_{j,h} \leq 0, \quad \forall i \in I_n \setminus \{1\}, \quad \forall j \in I_{k-1} \tag{4.41}$$

Esta restrição não pode ser considerada uma desigualdade válida, porque diminui o número de soluções inteiras em $k!$ vezes. Estas restrições produzem uma correspondência bijetiva entre o conjunto das soluções factíveis e o conjunto das k-partições. Porém, ainda é possível eliminar variáveis extras, mais especificamente, $\{x_{ki}\}_{i \in I_n}$.

Nesta variante, se um elemento não pertence a nenhuma parte de índice 1 até $k-1$, então ele só pode pertencer à parte k. Portanto, a última linha do modelo matemático $\mathcal{P}$ é desnecessária. Basta usar a substituição:

$$x_{k,i} = 1 - \sum_{j=1}^{k-1} x_{j,i}, \quad \forall i \in I_n \tag{4.42}$$

para reconstruir o modelo. São eliminadas n variáveis e modificadas as restrições (4.22a) para:

$$\sum_{j=1}^{k-1} x_{ji} \leq 1, \quad \forall i \in I_n \tag{4.43}$$

Exemplo 4.5.4. *Considere duas soluções para um problema de* 3*−partição dadas pelas matrizes abaixo.*

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

Nestas matrizes, as colunas nulas representam os elementos da parte 3.

Capítulo 5

Método *Local Branching*

Este capítulo apresenta uma adaptação do método *Local Branching* para a solução do MDMWNPP (Fischetti e Lodi, 2003). Justifica-se a aplicação do método *Local Branching* pela sua capacidade de excluir soluções exploradas do espaço de busca com a simples adição de restrições ao modelo matemático.

O capítulo está organizado da seguinte maneira: a Seção 5.1 apresenta as restrições *Local Branching* e suas propriedades relacionadas à distância de *Hamming*. A Seção 5.2 mostra o objeto central para o funcionamento do método proposto e a particularidade da sua vizinhança em esferas. A Seção 5.3 apresenta um algoritmo baseado em inserção e remoção de restrições.

5.1 Restrições *Local Branching*

As restrições *Local Branching* restringem o espaço de busca a soluções que estão a uma certa distância de *Hamming* de uma solução dada, baseando-se no número de variáveis com valores distintos. Considere uma solução incumbente x^s para o modelo (4.23). Seja B o conjunto dos pares de índices (j,i) tais que $x^s_{ji} = 1$, $B = \{(j,i) \in I_k \times I_n | x^s_{ji} = 1\}$. As restrições *Local Branching* são definidas pelas seguintes expressões:

$$\Delta(x^s, x) := \sum_{(j,i)\in B} (1 - x_{ji}) \; + \sum_{(j,i)\notin B} x_{ji} \leq R \tag{5.1a}$$

$$\Delta(x^s, x) := \sum_{(j,i)\in B} (1 - x_{ji}) \; + \sum_{(j,i)\notin B} x_{ji} = R \tag{5.1b}$$

$$\Delta(x^s, x) := \sum_{(j,i)\in B} (1 - x_{ji}) \; + \sum_{(j,i)\notin B} x_{ji} \geq R \tag{5.1c}$$

A restrição (5.1a) significa que uma solução factível deve estar a uma distância de *Hamming* menor ou igual a R da solução x^s. Com isso, a vizinhança $N_R(x^s)$ é como uma bola de raio R e centro x^s, com relação à distância de *Hamming*. Na restrição (5.1b), apenas soluções com distância de Hamming de x^s exatamente igual a R são permitidas. Esta estrutura de vizinhança é uma esfera de centro x^s e raio igual a R. Finalmente, a restrição (5.1c) considera todo o espaço externo à esfera de raio R e centro x^s como solução factível.

É importante observar que nem sempre as restrições *Local Branching* definem a exata estrutura de vizinhança mencionada, pois sua forma dependerá da intersecção com a região factível do modelo matemático do problema estudado.

5.2 Construção do Subproblema Local

Dadas duas soluções factíveis distintas x^1 e x^2 do modelo (4.23), com a eliminação de simetrias, a primeira coluna da matriz de solução sempre possui valor $x^1_{11} = 1$ e $x^2_{11} = 1$. Portanto, restam $n-1$ colunas para que as variáveis assumam valores distintos. Da segunda coluna em diante pode-se acrescentar, no máximo, duas unidades por coluna à distância de *Hamming* total, pois, devido às restrições (4.22a), só existe um j tal que $x_{ji} = 1$ para cada coluna i. Portanto, a distância de *Hamming* entre duas soluções factíveis do modelo (4.23) é sempre um número par. Então, pode-se concluir que:

$$\forall x^1, x^2, \text{ em que } x^1 \neq x^2, \ \exists r \in I_{n-1}: \quad \Delta(x^1, x^2) = 2r \tag{5.2}$$

e, como consequência da Expressão (5.2), a distância de *Hamming* máxima entre duas soluções factíveis é igual a $2(n-1)$, ou seja:

$$\max\{\Delta(x^1, x^2)\} = 2(n-1) \tag{5.3}$$

Existe uma decomposição do MDMWNPP em subproblemas com restrições (5.1b). Usando a Equação (5.2), considere o modelo (4.23) acrescido da restrição $\Delta(x^s, x) = 2r$, e nomeado como $P_r(x^s)$:

$$(P_r(x^s)) \ \min \quad y \tag{5.4a}$$

$$\text{suj. a} \quad \sum_{j=1}^{k} x_{ji} = 1, \ \forall i \in I_n \tag{5.4b}$$

$$\sum_{i=1}^{n} x_{ji} \geq 1, \quad \forall j \in I_k \tag{5.4c}$$

$$\sum_{i=1}^{n} v_{il}(x_{j_1 i} - x_{j_2 i}) \leq y, \quad \forall (1 \leq j_1 < j_2 \leq k), \forall l \in I_m \tag{5.4d}$$

$$\sum_{i=1}^{n} v_{il}(x_{j_2 i} - x_{j_1 i}) \leq y, \quad \forall (1 \leq j_1 < j_2 \leq k), \forall l \in I_m \tag{5.4e}$$

$$\sum_{(j,i)\in B} (1 - x_{ji}) \ + \ \sum_{(j,i)\notin B} x_{ji} = 2r \tag{5.4f}$$

$$y \in \mathbb{R}_+ \tag{5.4g}$$

$$x_{ji} \in \{0, 1\}, \quad \forall (j, i) \in I_k \times I_n \tag{5.4h}$$

Sabe-se que o modelo matemático (5.4) só assume soluções factíveis quando $1 \leq r \leq n-1$ devido à Equação (5.3). Seja $X_r(x^s)$ o espaço de soluções viáveis do problema $P_r(x^s)$. Também é importante notar que $P_{r_1}(x^s)$ e $P_{r_2}(x^s)$ não têm nenhuma solução factível em comum sempre que $r_1 \neq r_2$, pois suas distâncias da solução x^s são distintas neste caso, ou seja, $X_{r_1}(x^s) \cap X_{r_2}(x^s) = \emptyset$, se $r_1 \neq r_2$. Portanto, a coleção de subproblemas $\{P_r(x^s)\}_{r=1}^{n-1}$ é uma decomposição do modelo matemático (4.23) e, portanto, uma decomposição do MDMWNPP.

5.3 Algoritmo *Local Branching*

O algoritmo proposto trabalha com a inserção e remoção de restrições *Local Branching*, visando, além da definição de vizinhanças, a memorização do espaço de busca já explorado usando expressões matemáticas.

Seja x^1 uma solução factível do problema (4.23) e seja x^2 a solução do subproblema $P_1(x^1)$. O problema inicial do Algoritmo *Local Branching* é $P_1(x^1)$. O próximo subproblema será o subproblema $P_2(x^1)$ se a solução x^1 é melhor que a solução x^2 ou o subproblema $P_1(x^2)$ com a restrição adicional $\Delta(x^1, x) \geq 4$, caso contrário. A mesma lógica se aplica a cada uma das possibilidades nas demais iterações.

De modo geral, na iteração T, as decisões do método são tomadas da seguinte maneira. Seja M um conjunto de restrições do tipo (5.1c) e $(x^p)_{p\in I_h}$ uma sequência de soluções encontradas pelo Algoritmo *Local Branching*. O método *Local Branching* continua seu progresso, partindo de um subproblema $P_r(x^h)$ com as restrições adicionais do conjunto M e encontra uma solução x^{h+1}, resolvendo este subproblema. Caso a solução x^{h+1} seja ótima e melhor que x^h, o subproblema *local branching* é atualizado, adicionando-se a restrição $\Delta(x^h, x) \geq 2(r+1)$ ao conjunto M, formando o conjunto M'. Este passo elimina a região factível já explorada do subproblema $P_r(x^h)$. Por último, substitui-se a restrição $\Delta(x^h, x) = 2r$ por $\Delta(x^{h+1}, x) = 2$. Este procedimento cria um problema na forma $P_1(x^{h+1})$ com restrições adicionais M'. Caso a solução x^{h+1} não seja melhor que x^h, troca-se a restrição $\Delta(x^h, x) = 2r$ por $\Delta(x^h, x) = 2(r+1)$ formando o subproblema $P_{r+1}(x^h)$ com restrições adicionais M. Em último caso, quando o subproblema for comprovadamente infactível, a solução ótima estará em x^{h+1}.

O ideal para o bom funcionamento do algoritmo é resolver os subproblemas com otimalidade garantida. Entretanto, quando é estabelecido um tempo limite de resolução de cada subproblema, é possível encontrar uma solução x^{h+1} melhor que a solução x^h, mas que não seja comprovadamente ótima. Neste caso, o procedimento proposto aqui escolhe substituir a restrição $\Delta(x^l, x) = 2r$ por $\Delta(x^{h+1}, x) = 2$ sem adicionar a restrição $\Delta(x^h, x) \geq 2(r+1)$ ao subproblema. Caso x^{h+1} seja factível e pior que x^h, a decisão do procedimento de resolução é sair do método *Local Branching* e continuar a resolver o MIP no estado corrente, esta função será denotada por $SolveMIP(\cdot)$. Neste caso, é resolvido o MIP (4.23) com as restrições *local branching* que eliminam as regiões já explorada.

O modelo matemático (5.5) mostra a forma do subproblema do *Local Branching* após passar por h soluções distintas e fazer $T = \sum_{p=1}^{h} r_p$ iterações aplicadas ao modelo (5.4).

$$
\begin{aligned}
(P_r(x^s)) \ \min \quad & y && && \text{(5.5a)}\\
\text{suj. a} \quad & \sum_{j=1}^{k} x_{ji} = 1, \ \forall i \in I_n && && \text{(5.5b)}\\
& \sum_{i=1}^{n} x_{ji} \geq 1, \quad \forall j \in I_k && && \text{(5.5c)}\\
& \sum_{i=1}^{n} v_{il}(x_{j_1 i} - x_{j_2 i}) \leq y, \quad \forall (1 \leq j_1 < j_2 \leq k), \forall l \in I_m && && \text{(5.5d)}
\end{aligned}
$$

$$\sum_{i=1}^{n} v_{il}(x_{j_2 i} - x_{j_1 i}) \leq y, \quad \forall (1 \leq j_1 < j_2 \leq k), \forall l \in I_m \tag{5.5e}$$

$$\sum_{(j,i)\in B_h} (1 - x_{ji}) \; + \sum_{(j,i)\notin B_h} x_{ji} = 2r_h \tag{5.5f}$$

$$\sum_{(j,i)\in B_p} (1 - x_{ji}) \; + \sum_{(j,i)\notin B_p} x_{ji} \geq 2r_p, \quad \forall p \in I_{h-1} \tag{5.5g}$$

$$y \in \mathbb{R}_+ \tag{5.5h}$$

$$x_{ji} \in \{0, 1\}, \quad \forall (j, i) \in I_k \times I_n \tag{5.5i}$$

O subproblema (5.5) tem h restrições a mais que o modelo matemático (4.23). A restrição (5.5f) representa a vizinhança atual da iteração. As $h - 1$ restrições (5.5g) tornam infactíveis as regiões de busca dos subproblemas antigos. O aumento de h com o número de iterações depende da solução inicial escolhida e do tamanho da estrutura de vizinhanças. Os $r_p \in I_{h-1}$ definem por quantas vizinhanças cada solução passou sem ser superada.

O Algoritmo 1 resume as explicações anteriores. O critério de parada ocorre pelo esgotamento das $n - 1$ vizinhanças em uma determinada solução, pelo número máximo de iterações $Itermax$ (ou tempo máximo) e por encontrar o ótimo global. A função $SolveSP(\cdot)$ também tem um limite de tempo para resolver o subproblema (5.5). Então, devem existir desvios condicionais para todos os possíveis *status* da solução: $\{\text{factível}, \text{ótimo}, \text{infactível}\}$. Quando o método $SolveSP(\cdot)$ retorna uma solução de melhora factível, mas não ótima, não é possível excluir a região do subproblema, pois existe uma parte do espaço não explorada que ainda pode conter a solução ótima do problema geral.

Algorithm 1 *Local Branching* Adaptado.

1: **function** LocalBranching($\dot{x}$) ▷ Solução inicial
2: $r \leftarrow 1$ ▷ Vizinhança inicial
3: $t \leftarrow 1$ ▷ Contador
4: **while** $t < Itermax$ **do**
5: *add* $\Delta(\dot{x}, x) = 2r$
6: $\ddot{x} \leftarrow SolveSP()$ ▷ Solução do subproblema
7: *remove* $\Delta(\dot{x}, x) = 2r$
8: **if** *optimal* **then**
9: **if** $f(\ddot{x}) \geq f(\dot{x})$ **then** ▷ Sem melhoria
10: $r \leftarrow r + 1$ ▷ Aumento da vizinhança
11: **else** ▷ Melhoria da solução
12: *add* $\Delta(\dot{x}, x) \geq 2(r + 1)$ ▷ Exclusão do espaço vizinho de $\dot{x}$
13: $r \leftarrow 1$ ▷ Retorno para a primeira vizinhança
14: $\dot{x} \leftarrow \ddot{x}$ ▷ Atualiza solução
15: **end if**
16: **else if** *feasible* **then** ▷ Não reduzir o espaço de busca
17: **if** $f(\ddot{x}) \geq f(\dot{x})$ **then**
18: **return** $SolveMIP()$
19: **else**
20: $r \leftarrow 1$
21: $\dot{x} \leftarrow \ddot{x}$
22: **end if**
23: **else** ▷ O espaço de busca é vazio
24: **return** $\dot{x}$
25: **end if**
26: *add* $\Delta(\dot{x}, x) = 2r$
27: $t \leftarrow t + 1$
28: **end while**
29: **return** $\dot{x}$
30: **end function**

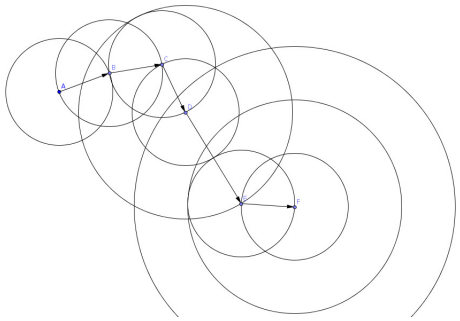


Figura 5.1: Representação da exploração do Algoritmo 1 assumindo que todos os subproblemas encontram somente soluções factíveis de melhora sem otimalidade comprovada

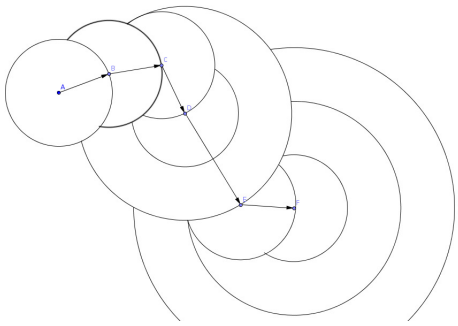


Figura 5.2: Efeito das restrições adicionadas na linha 12 do algoritmo 1 no espaço de busca do Algoritmo 1 assumindo que todos os subproblemas encontram soluções ótimas de melhora.

As Figuras 5.1 e 5.2 mostram a exploração do espaço de busca, de modo abstrato, feita pelo Algoritmo *Local Branching*. Cada vez que o Algoritmo entra nos desvios condicionais das linhas 11 ou 19, gera-se uma nova solução melhor que a anterior. Esta sequência de soluções e suas respectivas vizinhanças são exemplificadas nas Figuras 5.1 e 5.2. Os centros são as soluções aceitas pelo *Local Branching* e as circunferências são definidas pelas restrições $\Delta(\dot{x}, x) = 2r$. A Figura 5.1 mostra como seria o percurso do *Local Branching* no espaço de busca caso todas as soluções encontradas tivessem o *status* factível. A Figura 5.2 exibe o caso no qual todas as novas soluções têm *status* otimo. Neste último caso, nota-se que os subproblemas têm um espaço de busca mais limitado, devido às restrições $\Delta(\dot{x}, x) \geq 2r$.

Capítulo 6

Resultados

Este capítulo apresenta as informações produzidos nesta pesquisa e a análise dos resultados obtidos. A Seção 6.1 apresenta a metodologia geral de execução e análise dos experimentos computacionais. A Seção 6.2 analisa os resultados dos modelos matemáticos (4.23), (4.31) e (4.34). A Seção 6.3 avalia o desempenho do método *Local Branching*. A Seção 6.4 faz uma análise estatística dos resultados mostrados em Nikolic et al. (2022) para os modelos (4.23), (4.31).

6.1 Configuração geral dos experimentos e das análises estatísticas

Os testes computacionais são realizados usando um único núcleo em cada tarefa para manter o controle do ambiente experimental. A quantidade de memória RAM é igual para todas as tarefas. Além disso, as instâncias para os experimentos são retiradas da literatura e foram propostas por Kojić (2010). As características das instâncias são classificadas de acordo com suas propriedades, de modo a permitir o isolamento de causa e efeito durante a avaliação dos resultados.

A meta dos experimentos computacionais é fazer uma comparação entre cada formulação proposta com e sem os tratamentos desenvolvidos durante a pesquisa teórica e verificar se a aplicação das técnicas de melhoria dos modelos matemáticos são efetivas para o desempenho do resolvedor. Os experimentos tem tempo limite fixo para garantir equidade entre as execuções. As medidas de comparação estudadas são: tempo de execução em segundos denotado por t, limite superior do valor de função objetivo alcançado denotado por ub, limite inferior do valor de função objetivo alcançado denotado por lb e $status$ da solução. Os resultados são sumarizados seguindo o modelo da Tabela 6.1. Durante a coleta de dados as execuções que usarem memória `swap` serão retirados da análise para uma comparação mais justa.

Os dados do experimento são independentes por construção e, portanto, cabe uma análise estatística. Testam-se as hipóteses sobre o desempenho dos algoritmos com a verificação da diferença estatística significativa entre os resultados. As medidas p_{Mi} a serem comparadas são: média, desvio padrão ou proporção de algum resultado dos experimentos computacionais como ub, lb, t ou número de soluções

Tabela 6.1: Dados descritivos sumarizados.

Status	Medida	ub	lb	t
Factível: a	Min.	u_1	l_1	t_1
Ótimo :$T-a$	Qu1.	u_2	l_2	t_2
	Mediana	u_3	l_3	t_3
	Média	u_4	l_4	t_4
	Qu3.	u_5	l_5	t_5
	Max.	u_6	l_6	t_6

ótimas obtidas. As hipóteses são descritas matematicamente por

$$\begin{cases} H_0: \ p_{M1} \geq p_{M2} \\ H_1: \ p_{M1} < p_{M2} \end{cases} \tag{6.1}$$

ou

$$\begin{cases} H_0: \ \forall i,j : p_{Mi} = p_{Mj} \\ H_1: \ \exists i,j : p_{Mi} \neq p_{Mj}. \end{cases} \tag{6.2}$$

A o valor p_{Mi} é uma medida de desempenho do método Mi, uma proporção do número de vezes que Mi tem sucesso. A hipótese nula H_0 afirma que não há diferença entre os métodos e a hipótese alternativa H_1 é descrita como a negação lógica H_0. A H_0 é assumida como verdade probabilística, a decisão de rejeita-la dependerá do $p-valor$ encontrado no teste. O nível de significância dos testes usados nos resultados é sempre de 95%, um $p-valor < 0.05$ rejeita H_0.

Usam-se as variantes do Teste Exato de Fisher seguidas de Testes de Múltiplas Comparações para a análise do *status* da solução e o Teste de Wilcoxon para avaliar as diferenças entre os resultados numéricos (Dalgaard, 2008). O delineamento dos experimentos computacionais é baseado no seguinte: cada modelo matemático é um grupo e os resultados das instâncias resolvidas por cada modelo constituem as réplicas, classificações ou elemento numérico. Portanto, o número de métodos testados é a quantidade de grupos em todos os testes de hipóteses apresentados. A análise estatística destes dados será feita usando as bibliotecas para a execução dos testes da linguagem `R` (R Core Team, 2018).

A Tabela 6.2 mostra o formato dos dados para comparar se a proporção de ótimos alcançados por algum método M_i se destaca entre os demais com significância estatística. A quantidade k_i é o número de ótimos e N_i é o número de instâncias usadas nas execuções do método M_i. Além disso, $k = \sum_{i=1}^{n} k_i$ e $N = \sum_{i=1}^{n} N_i$.

Tabela 6.2: Tabela de contingência.

	Ótimo	Factível	total lin.
M_1	k_1	$N_1 - k_1$	N_1
M_2	k_2	$N_2 - k_2$	N_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$
M_n	k_n	$N_n - k_n$	N_n
total col.	k	$N - k$	N

O resultado da realização dos Testes de Múltiplas Comparações com p-valor ajustado pelo método de Benjamini e Hochberg (1995) será apresentado no formato da Tabela 6.3.

Tabela 6.3: Múltiplas comparações usando Teste de Wilcoxon ou Fisher.

p-valor	M_1	M_2	M_3	...	$M_{(n-1)}$
M_2	p_{12}	-	-	...	-
M_3	p_{13}	p_{23}	-	...	-
M_4	p_{14}	p_{24}	p_{34}	...	-
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
M_n	p_{1n}	p_{2n}	p_{3n}	...	$p_{n-1,n}$

Para reproduzir estes resultados, bastará gerar novas instâncias com características mais diversas e repetir os mesmos experimentos. Todo o material necessário para reprodução deste trabalho será postado em repositório de acesso público.[1]

6.2 Experimentos das Formulações

Este primeiro conjunto de experimentos computacionais foram realizados em um computador com CPU Intel Core i7-3770, 3.4 GHz, 8 núcleos, 32 GB de RAM e sistema operacional Ubuntu 16.04 de 64 bits. Os modelos matemáticos (4.23) e (4.34) foram implementados em `C++` usando a biblioteca ILOG Concert Technology e executados através do resolvedor CPLEX versão 12.7, com a configuração padrão, na versão do compilador `gcc` 5.4.0.

As instâncias usadas nos experimentos foram geradas inicialmente por Kojić (2010) e fornecidas pelos autores dos artigos Kratica et al. (2014) e Pop e Matei (2013b). Todos os artigos citados na Tabela 3.1 usam o mesmo conjunto de instâncias, com exceção de Hacibeyoglu et al. (2018) e Hacibeyoglu et al. (2014). Cada sequência V usada para os testes contém n vetores com a dimensão m, sendo $n \in \{50, 100, 200, 300, 400, 500\}$ e $m \in \{2, 3, 4, 5, 10, 15, 20\}$. Existem 5 tipos de instâncias, indexadas na forma $r \in \{a, b, c, d, e\}$, para cada combinação (n, m), totalizando 210 possíveis sequências diferentes. As instâncias são complementadas pela associação de um valor $k \in \{2, 3, 4\}$, que é o tamanho da partição. O nome das instâncias tem o seguinte formato: k_n_mr.

Usa-se um único núcleo para cada execução. O tempo limite dado ao solucionador para cada instância de teste é de 1800 segundos. A memória máxima usada para teste é de 5 GB. Após cada execução, o algoritmo grava uma linha com os seguintes atributos da solução encontrados: Status $\in \{\text{Infeasible}, \text{Feasible}, \text{Optimal}\}$, um número real UB (limite superior), um número real LB (limite inferior) e o tempo de execução em segundos. Cada linha representa uma instância resolvida. O ambiente computacional de teste usado é o mesmo para resolver os modelos (4.34) e (4.23). O artigo Kojić (2010) foi completamente reproduzido da parte teórica aos experimentos.

[1] https://github.com/AlexandreFrias

6.2.1 Resultados para $k = 2$

As Tabelas 6.4 e 6.5 apresentam os resultados sumarizados para o modelo (4.34), proposto por Kojić (2010), e para o modelo (4.23), proposto neste trabalho, respectivamente, para $k = 2$. Os dados completos são apresentados nas Tabelas A.1-A.6 no Apêndice A. Estes resultados correspondem a solução do problema MDTWNPP assim como em Kojić (2010). Conforme mostra a Tabela 3.1, esta é a única referência que utiliza modelo matemático para a solução desse problema.

As Tabelas 6.4 e 6.5 tem como objetivo facilitar a análise e a compreensão dos resultados encontrados. A Tabela 6.4 mostra que uma solução ótima foi encontrada pelo modelo (4.34) na instância 2_50_20c. Nesse caso, os modelos (4.34) e (4.23) têm o mesmo valor de UB, mas o modelo (4.23) não certifica a otimalidade da solução. É possível fazer uma comparação entre as medidas apresentadas nas tabelas 6.4 e 6.5. Comparando a média de UB para todas as 210 instâncias testadas, é mostrado que o modelo (4.23) é melhor que o modelo (4.34). Nota-se que a diferença é muito pequena em todas as medidas das tabelas e isso não significa uma diferença estatística significativa entre as medidas dos dois modelos nas análises de quantil.

Para verificar se existe uma diferença estatística entre os valores de UB e de LB obtidos pelos modelos (4.23) e (4.34), bem como em relação ao tempo de execução dos testes experimentais com estes modelos, utiliza-se o Teste de Wilcoxon pareado com correção de continuidade (Dalgaard, 2008). Estes testes foram aplicados numa amostra composta pelos resultados encontrados para todas as instâncias com $k = 2$. A Tabela 6.6 mostra que, para o conjunto de instâncias utilizadas neste trabalho, os valores de UB encontrados através do modelo (4.23) não são inferiores aos valores de UB do modelo (4.34), os valores de LB encontrados pelo modelo (4.23) não são superiores aos valores de LB referentes ao modelo (4.34), e existe uma diferença significativa entre o tempo de execução destes dois modelos. A diferença está principalmente relacionada com a solução ótima encontrada para o exemplo 2_50_20c, com o tempo de execução de $1271,12$ segundos.

Tabela 6.4: Resultados obtidos através do modelo (4.34).

Status	Medida	UB	LB	Tempo
Factível: 209	Min.	0.25	0	1271
Ótimo : 1	Qu1.	121.06	0	1797
	Mediana	1899.38	0	1800
	Média	14481.81	1263	1795
	Qu3.	29184.12	0	1800
	Max.	58791.40	50556	1800

Outra análise descritiva dos resultados para $k = 2$ é fornecida pela Tabela 6.7, que contém o número de soluções encontradas pelo modelo (4.23) que são melhores, piores ou iguais ao modelo (4.34).

O modelo (4.23) é mais efetivo em encontrar limites superiores melhores enquanto o modelo (4.34) apresenta melhores limites inferiores.

Tabela 6.5: Resultados obtidos através do modelo (4.23) usando $k = 2$.

Status	Medida	UB	LB	Time
Factível: 210	Min.	0.13	0	1761
	Qu1.	146.02	0	1798
	Mediana	1951.33	0	1799
	Média	14303.46	1248	1797
	Qu3.	29153.33	0	1800
	Max.	56535.80	45184	1800

Tabela 6.6: Teste de Wilcoxon pareado ($\Delta = \mu_{k=2} - \mu_{kojic}$).

Medida	W	p-valor	H_1
UB	8613	0.0955	$\Delta < 0$
LB	26	0.5808	$\Delta > 0$
$Time$	7207	0.0002	$\Delta \neq 0$

Tabela 6.7: Contagem de resultados obtidos pelo modelo (4.23) em relação ao modelo (4.34).

	Melhor	Pior	Igual
UB	103	93	14
LB	4	6	200

6.2.2 Resultados para $k = 3$ e $k = 4$

As Tabelas 6.8 e 6.9 resumem os resultados obtidos pelo modelo (4.23) para $k = 3$ e $k = 4$, respectivamente. O conjunto completo de resultados estão nas Tabelas A.7-A.12 disponíveis no Apêndice A.

Tabela 6.8: Resultados do modelo (4.23) para $k = 3$.

Status	Medida	UB	LB	Time
Factível: 210	Min.	39.71	0.0	1707
	Qu1.	1749.71	0.0	1799
	Mediana	9102.22	0.0	1799
	Média	28504.28	377.1	1789
	Qu3.	59631.55	0.0	1800
	Max.	96176.20	18418.0	1800

De acordo com os resultados apresentados nestas tabelas, não foi provada a otimalidade de nenhuma solução encontrada. Uma análise mais detalhada dos resultados do modelo (4.23) para $k = 3$ e $k = 4$ será feita na Seção 6.3, que apresenta a comparação deste procedimento de resolução através de um resolvedor e o procedimento de resolução usando o algoritmo *Local Branching* (Algoritmo 1), e na Seção 6.4, que apresenta a comparação dos resultados obtidos por esta formulação com os resultados obtidos pelo modelo (4.31). Os resultados de ambas as seções foram obtidos por experimentos computacionais que usam um ambiente computacional diferente do ambiente computacional referente aos resultados da Tabelas 6.8 e 6.9.

Tabela 6.9: Resultados do modelo (4.23) para $k = 4$.

Status	Medida	UB	LB	Time
Factível: 210	Min.	254.9	0.0	1755
	Qu1.	4213.8	0.0	1799
	Mediana	15769.0	0.0	1799
	Média	37841.8	443.1	1798
	Qu3.	77292.1	0.0	1799
	Max.	113213.0	19955.0	1800

6.3 Experimentos usando o algoritmo *Local Branching*

Este conjunto de experimentos computacionais foi realizado em um computador com um CPU Intel(R) Xeon(R) E5620 @ 2.40GHz, 16-core, 128GB RAM, e sistema operacional CentOS Linux de 64 bits 7.9.2009 (Core). O modelo matemático (4.23), foi implementado em `C++` usando a biblioteca ILOG Concert Technology e executado através do resolvedor `CPLEX` versão 12.6, com a configuração padrão, usando o compilador `gcc` versão 5.4.0. Outra diferença em relação aos experimentos da Seção 6.2 é a memória máxima usada para o teste, que é de 6 GB por instância.

Os experimentos computacionais do método *Local Branching* foram realizados em uma máquina diferente e, por esse motivo, usam-se tabelas diferentes para sumarizar os resultados do modelo matemático. O conjunto de instâncias utilizado nestes experimentos é o mesmo descrito na Seção 6.2.

As Tabelas 6.10, 6.11 e 6.12 apresentam os resultados do modelo (4.23) para valores de $k \in \{3, 4, 5\}$. As Tabelas 6.13, 6.14 e 6.15 apresentam os resultados do Algoritmo 1, usando o modelo (4.23) como base, descrito no Capítulo 5, para valores de $k \in \{3, 4, 5\}$. Nota-se que os resultados do modelo (4.23) são ligeiramente piores do que os resultados das Tabelas 6.8 e 6.9, tanto para o limite superior quanto o limite inferior, devido a diferença de ambiente computacional. Mas serão usados os resultados das Tabelas 6.10-6.12 nas análises feitas nesta seção para garantia de isonomia do ambiente computacional.

Tabela 6.10: Sumário de resultados para o MILP (4.23) com $k = 3$.

Status	Medida	UB	LB	Time
Factível:210	Min.	55.32	0.00	1788
	Qu1.	1982.77	0.00	1794
	Mediana	10258.95	0.00	1796
	Média	30065.82	25.14	1796
	Qu3.	65261.62	0.00	1798
	Max.	100423.00	3931.88	1799

Todas as medidas das tabelas apresentadas para o UB do MILP se mostram maiores que as do algoritmo *Local Branching*. No entanto, apenas com a análise dos resultados sumarizados nenhuma conclusão é possível. Portanto, foram feitas algumas análises estatísticas dos resultados. A Tabela 6.16 mostra que, quando

Tabela 6.11: Sumário de resultados para o MILP (4.23) com $k = 4$.

Status	Medida	UB	LB	Time
Factível:210	Min.	326.6	0.0	1791
	Qu1.	4679.0	0.0	1794
	Mediana	17164.5	0.0	1796
	Média	40285.9	119.9	1796
	Qu3.	83234.2	0.0	1798
	Max.	116353.0	8365.2	1799

Tabela 6.12: Sumário de resultados para o MILP (4.23) com $k = 5$.

Status	Medida	UB	LB	Time
Factível:210	Min.	656.7	0.00	1791
	Qu1.	8304.7	0.00	1795
	Mediana	24634.8	0.00	1797
	Média	47890.7	95.26	1796
	Qu3.	94371.5	0.00	1798
	Max.	133908.0	7600.34	1799

Tabela 6.13: Sumário de resultados para o algoritmo *Local Branching* com $k = 3$.

Status	Medida	UB	LB	Time
Factível:210	Min.	46.74	0.0	1786
	Qu1.	2028.70	0.0	1793
	Mediana	9828.77	0.0	1796
	Média	29696.39	164.5	1795
	Qu3.	63111.80	0.0	1798
	Max.	93892.70	11188.6	1799

Tabela 6.14: Sumário de resultados para o algoritmo *Local Branching* com $k = 4$.

Status	Medida	UB	LB	Time
Factível:210	Min.	178.7	0.0	1786
	Qu1.	4098.2	0.0	1792
	Mediana	16595.9	0.0	1796
	Média	39757.5	314.3	1795
	Qu3.	81332.2	0.0	1797
	Max.	117767.0	14628.1	1799

comparados num teste estatístico pareado, todos os resultados do Algoritmo 1 obtém um desempenho melhor que o MILP. Quando a medida avaliada é o LB, essa diferença estatística só existe nas instâncias com $k = 4$.

Tabela 6.15: Sumário de resultados para o algoritmo *Local Branching* com $k = 5$.

Status	Medida	UB	LB	Time
Factível:210	Min.	364.4	0.0	1787
	Qu1.	7368.9	0.0	1792
	Mediana	24552.6	0.0	1796
	Média	46160.5	505.3	1795
	Qu3.	89708.6	0.0	1797
	Max.	129403.0	20855.9	1799

Tabela 6.16: Resultados da aplicação do Teste de Wilcoxon pareado sobre os valores de UB, para a hipótese $H_1 : \Delta_{UB} = \mu_{Local} - \mu_{MILP} < 0$, e sobre os valores de LB, para a hipótese $H_1 : \Delta_{LB} = \mu_{Local} - \mu_{MILP} > 0$.

Medida	W_{UB}	p-valor (UB)	W_{LB}	p-valor (LB)
$k = 3$	9336	0.02415	376	0.2519
$k = 4$	9025	0.009969	1205	0.03141
$k = 5$	7650	5.072e-05	2062.5	0.07376

6.4 Análise dos resultados para comparação com o modelo (4.31)

Os resultados apresentados no artigo Nikolic et al. (2022) merecem uma maior atenção neste capítulo porque o artigo demonstra uma reprodução completa dos modelos matemáticos (4.34), (4.23) e (4.31).

Todos os algoritmos propostos foram implementados em `C++` usando o `gcc` 5.4.2. Todas os experimentos computacionais foram realizadas num *cluster* de máquinas com Linux com CPUs Intel Xeon E5649 com 2.53 GHz e um limite de memória de 8 GB em um único núcleo. O tempo máximo de computação permitido para cada execução foi limitado a 1200 segundos. Os modelos matemáticos foram implementados utilizando a biblioteca ILOG Concert Technology e executados através do resolvedor `CPLEX` versão 12.5.1.

As instâncias usadas nessas análises são uma extensão daquelas propostas por (Kojić, 2010). Cada sequência V usada para os testes contém n vetores com a dimensão m, sendo $n \in \{30, 40, 50, 100, 200, 300, 400, 500\}$ e $m \in \{2, 3, 5, 10, 15, 20\}$. Existem 5 tipos de instâncias, indexadas na forma $r \in \{a, b, c, d, e\}$, para cada combinação (n, m, r), totalizando 240 possíveis sequências diferentes. As instâncias são complementadas pela associação de um valor $k \in \{2, 3, 4, 5, 10, 20, 30\}$, que representa o tamanho da partição.

Devido ao fato dos resultados apresentarem muitas soluções ótimas, permite-se uma análise estatística não paramétrica com o Teste Exato de Fisher. Esses testes podem evitar falsas conclusões derivadas de uma precipitação por comparação numérica "intuitiva". Aqui, para facilitar o entendimento das tabelas, denotam-se as formulações matemáticas descritas no Capítulo 4 de acordo com as seguintes siglas, as mesmas do artigo (Nikolic et al., 2022)

- Kojic: MILP (4.34) apresentado em Kojić (2010);

- FdSdS: MILP (4.23) apresentado em Faria et al. (2021);
- New: MILP (4.31) apresentado em Nikolic et al. (2022).

Os principais dados de análise são as quantidades de soluções ótimas atingidas por cada modelo para cada valor de k, que são apresentadas na Tabela 6.17. Para cada linha da Tabela 6.17, realiza-se um Teste de Fisher Exato para verificar se a proporção de ótimos na amostra obtida por um modelo é significativamente maior que na amostra do outro. O padrão da leitura do teste estatístico aqui descrito é a leitura das amostras na ordem, da esquerda para direita. Por exemplo, na linha 3 da Tabela 6.17, a pergunta é: "*A proporção de melhores resultados atingidos pela amostra de FdSdS é diferente da de New?*"

Tabela 6.17: Número de vezes em que o modelo atinge o melhor UB da literatura.

k	Kojic	FdSdS	New	p-valor
2	138	125	95	0.000
3	n/a	151	119	0.004
4	n/a	137	125	0.313
5	n/a	119	137	0.120
10	n/a	88	161	0.000
20	n/a	53	206	0.000
30	n/a	46	235	0.000

Ignorando a primeira linha da Tabela 6.17 e fazendo a análise entre FdSdS e New, é possível afirmar que FdSdS atinge uma maior proporção de melhores resultados maior que New para valores de $k = 3$. Para $k \in \{4, 5\}$, não existe diferença significativa nessa medida de resultados. O modelo New é significativamente melhor para $k \in \{10, 20, 30\}$.

Observe que na primeira linha da Tabela 6.17 contém resultados de 3 modelos. Como explicado na Seção 6.1, neste caso, deve ser aplicado o Teste de Fisher Exato 3×2 e caso exista uma diferença entre as 3 amostras, deve-se conduzir uma comparação por pares como na Tabela 6.3.

Tabela 6.18: Proporção de melhor solução alcançada.

MILP	Alcança	Não Alcança
Kojic	138	102
FdSdS	125	115
New	95	145

A Tabela 6.18 apresenta os resultados referentes à quantidade de melhor solução obtida para cada modelo. O Teste de Fisher referente a estes resultados tem p-valor igual a 0.0002957. Isso demonstra que existe diferença estatística entre os resultados obtidos e, portanto, um teste dois a dois deve ser realizado para esclarecer quais são os melhores ou piores resultados. A Tabela 6.19 apresenta os resultados da análise dos pares. Nesta tabela, a primeira coordenada do par de amostras comparadas vem da linha e a segunda coordenada vem da coluna, ou seja, a primeira pergunta a ser respondida aqui é: "*A proporção de melhores resultados atingidos pela amostra de*

Kojic é maior do que a de FdSdS?" e o p-valor nessa coordenada é a resposta do teste. Os resultados na Tabela 6.19 indicam que, para $k = 2$, não existe diferença entre Kojic e FdSdS, mas ambos superam o modelo New.

Tabela 6.19: Resultados dos p-valores ajustados para a proporção de melhor solução alcançada por pares.

p-valor	Kojic	FdSdS
FdSdS	0.136	-
New	0.000	0.006

Quanto a proporção de soluções ótimas encontradas, exibidos na Tabela 6.20 , os modelos não demonstram diferença estatística significativa alguma.

Tabela 6.20: Número de vezes em que o modelo atinge o resultado ótimo.

k	Kojic	FdSdS	New	p-valor
2	45	44	43	0.992
3	n/a	29	31	0.890
4	n/a	18	18	1.000
5	n/a	15	14	1.000
10	n/a	8	11	0.641
20	n/a	7	13	0.253
30	n/a	30	35	0.594

Os resultados aqui tratados não consideram meta-heurísticas devido ao padrão estabelecido nas comparações. Análises comparativas entre heurísticas e métodos exatos iniciais encontram-se em Faria et al. (2021). Mostram-se resultados mais profundos e completos em Djukanović et al. (2022).

Capítulo 7

Conclusões Gerais

Esta Tese apresentou um estudo do MDMWNPP com foco em técnicas baseadas em programação matemática. O principal foco do trabalho foi o desenvolvimento de modelos matemáticos para a solução do MDMWNPP, ausentes na literatura até a publicação do artigo Faria et al. (2021). A partir de um núcleo duro de restrições caracterizadoras de k-partições, bastou encontrar o conjunto de restrições auxiliares derivadas da função objetivo do enunciado 2.5.1. Partindo dos modelos matemáticos desenvolvidos, criou-se uma base para o uso de métodos baseados na inserção e remoção de pseudo-cortes criados a partir de subproblemas locais, técnica chamada de *Local Branching*, que visa reduzir o espaço de soluções factíveis até o vazio e encontrar a solução nesse processo.

As análises descritivas e testes estatísticos sobre as amostras de resultados obtidos pelos métodos aqui propostos ou reproduzidos, validam os modelos 4.23, 4.31 e o Algoritmo 1, assim como o modelo reproduzido da literatura 4.34. Os resultados são consistentes e pode-se dizer que os objetivos da Seção 1.3 foram alcançados. O trabalho desta Tese apresentou contribuições efetivas para o campo de pesquisa, tendo muitos de seus resultados corroborados pelos seus pares em publicações revisadas e, também, serviu como modelo para artigos posteriores como Nikolic et al. (2022).

Este capítulo apresenta o estado atual da pesquisa e suas conclusões. A Seção 7.1 reapresenta as publicações resultantes desta pesquisa, incluindo artigos em andamento. A Seção 7.2 mostra algumas considerações e discussões baseadas naquilo que já foi testado e reproduzido nesta pesquisa. A Seção 7.3 apresenta propostas de trabalhos futuros envolvendo, ainda, muitas técnicas de programação matemática e futuras melhorias metodológicas.

7.1 Publicações

(i) Comparação entre Heurísticas Construtivas e ILS na Solução do Problema da k-partição de Números (Faria et al., 2017)

(ii) Análise de Modelos Matemáticos para o Problema da Múltipla Partição de Números (Faria et al., 2018b)

(iii) *Variable Neighborhood Descent applied to Multi-way Number Partitioning Pro-*

blem (Faria et al., 2018a)

(iv) *A Variable Neighborhood Search Approach for Solving the Multidimensional Multi-Way Number Partitioning Problem* (Faria et al., 2019a)

(v) *Variable Neighborhood Descent Branching applied to the Multi-Way Number Partitioning Problem* (Faria et al., 2019b)

(vi) *A Mixed-Integer Linear Programming model to solve the Multidimensional Multi-Way Number Partitioning Problem* (Faria et al., 2021)

(vii) *Local Branching Based Algorithm for Solving Multidimensional Multi-Way Number Partitioning Problem*. Em andamento.

O artigo fornece um algoritmo para resolver o MDMWNPP utilizando um modelo matemático (4.23) e uma abordagem baseada na construção de subproblemas que reduzem o espaço viável em cada iteração que, por sua vez, consistem em resolver o modelo matemático sobre uma esfera induzida pela distância *hamming* entre as soluções. A estrutura de subproblemas usada é a mesma apresentada no Capítulo 5. Foram realizados experimentos computacionais com instâncias da literatura. O método proposto mostrou resultados competitivos contra o clássico *Branch-and-Cut*.

7.2 Considerações Finais

No desenvolvimento desta Tese, novas formulações matemáticas foram desenvolvidas para o MDMWNPP. Antes deste trabalho começar a ser desenvolvido, só se encontravam na literatura abordagens via meta-heurísticas para tratar este problema. Por outro lado, tanto a abordagem via programação matemática quanto via meta-heurísticas são insuficientes para encontrar soluções ótimas em uma proporção satisfatória nos conjuntos de instâncias da literatura. A metodologia de solução via *Local Branching* já se demonstra um pouco mais efetiva do que o simples uso de um MILP para o MDMWNPP.

Os resultados ruins causaram o abandono de meta-heurísticas e rápida migração para a programação matemática ou métodos híbridos. Isso pesou positivamente para o trabalho mesmo sendo um insucesso. Os trabalhos com meta-heurísticas ainda são um complemento aos métodos exatos como o *Local Branching*.

Devido aos diversos imprevistos no decorrer desta pesquisa, os resultados se dispersaram por vários ambientes computacionais e em nenhum momento as diversas abordagens foram testadas em um mesmo experimento e máquina.

Existem questões metodológicas a serem melhor avaliadas. Deve-se comparar os tempos usados por dois algoritmos para alcançarem a solução ótima, nas instâncias em que isso ocorre em ambos? Ou seria melhor ignorar o status da solução e avaliar o tempo, mesmo esgotado, para todas as instâncias? As duas escolhas são válidas mas possuem limitações. Uma alternativa não trivial seria encontrar o maior subconjunto de instâncias sem diferença significativa de UB, LB e proporção de ótimos atingidos para realizar a comparação de tempo.

Os testes estatísticos precisam usar um novo conjunto de instâncias como amostra. Um que seja mais bem organizado pelas suas características e que possibilite um

melhor delineamento dos testes estatísticos. A ideia é ter uma amostra significativa para cada (n, m, k).

7.3 Trabalhos Futuros

Os primeiros trabalhos futuros são aqueles com material para serem feitos, como o artigo em andamento aplicando o *Local Branching*. Falta ainda uma comparação entre o MILP 4.31 e o Algoritmo 1. Todos os demais pares possíveis foram comparados e analisados. A pesquisa realizada ainda não finalizou todas as ideias desenvolvidas como pode ser visto no Apêndice B. Além destas formulações prontas e não publicadas, existe também a análise de simetrias, descritas na Seção 4.5, aplicada aos modelos matemáticos (4.23), (4.31) e (B.2) a serem testadas. Esses testes devem medir o efeito da inserção das restrições de eliminação das simetrias em cada modelo.

Outra sugestão de trabalho futuro é verificar a influência do método *Local Branching*, apresentado no Capítulo 5, aplicado aos modelos com ou sem os tratamentos da Seção 4.5, sobre a qualidade da solução encontrada pelas duas formulações. Uma possibilidade muito ampla é o uso de heurísticas para definir a prioridade da expansão de nós na árvore de busca dentro dos métodos de programação matemática.

Referências Bibliográficas

Ausiello, G.; Crescenzi, P.; Kann, V.; Marchetti-sp,; Gambosi, Giorgio e Spaccamela, Alberto M. (2003)a. *Complexity and Approximation: Combinatorial Optimization Problems and Their Approximability Properties*, Capítulo Sequential Algorithms for Partitioning Problems, p. 50–60. Springer-Verlag New York, Inc., 1st edição. 16

Ausiello, Giorgio; Crescenzi, Pierluigi; Gambosi, Giorgio; Kann, Viggo; Marchetti-Spaccamela, Alberto e Protasi, Marco. (2003)b. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer. 2

Bazaraa, Mokhtar S.; Jarvis, John J. e Sherali, Hanif D. (2009). *Linear Programming and Network Flows*. Wiley. 1

Benjamini, Yoav e Hochberg, Yosef. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B (Methodological)*, v. 57, n. 1, p. 289–300. ISSN 00359246. 39

Bertsimas, Dimitris e Weismantel, Robert. (2005). *Optimization over integers*. Athena Scientific. ISBN 978-0-97591-462-5. 3

Cook, Stephen A. (1971). The complexity of theorem-proving procedures. *Proceedings of the Third annual ACM Symposium on Theory of Computing*, p. 151–158. ACM, (1971). 2, 29

Crescenzi, Pierluigi; Kann, Viggo; Halldórsson, Magnús; Karpinski, Marek e Woeginger, Gerhard. A compendium of NP optimization problems. URL: https://www.nada.kth.se/ viggo/problemlist/compendium.html, July(1997). 2

Dalgaard, Peter. (2008). *Introductory statistics with R*. Springer, New York, NY, USA, 2 edição. 38, 40

Djukanović, Marko; Kartelj, Aleksandar e Blum, Christian. Self-adaptive CMSA for solving the multidimensional multi-way number partitioning problem. http://dx.doi.org/10.2139/ssrn.4241190, Sept.(2022). 17, 18, 19, 46

Faria, Alexandre Frias; de Souza, Sérgio Ricardo; Souza, Marcone Jamilson Freitas; Silva, Carlos Alexandre e Nazário Coelho, Vitor. (2019)a. A variable neighborhood search approach for solving the multidimensional multi-way number partitioning

problem. Sifaleras, Angelo; Salhi, Said e Brimberg, Jack, editors, *Variable Neighborhood Search*, p. 243–258, Cham. Springer International Publishing. ISBN 978-3-030-15843-9. 15, 17, 18, 48

Faria, Alexandre Frias; de Souza, Sérgio Ricardo e de Sá, Elisangela Martins. (2021). A mixed-integer linear programming model to solve the multidimensional multi-way number partitioning problem. *Computers & Operations Research*, v. 127, p. 105133. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2020.105133. URL `https://www.sciencedirect.com/science/article/pii/S0305054820302501`. 15, 17, 18, 19, 45, 46, 47, 48

Faria, Alexandre Frias; de Souza, Sérgio Ricardo; de Sá, Elisangela Martins e Silva, Carlos Alexandre. (2019)b. Variable neighborhood descent branching applied to the multi-way number partitioning problem. *Electronic Notes in Theoretical Computer Science*, v. 346, p. 437 – 447. ISSN 1571-0661. doi: https://doi.org/10.1016/j.entcs.2019.08.039. URL `http://www.sciencedirect.com/science/article/pii/S1571066119300908`. The proceedings of Lagos 2019, the tenth Latin and American Algorithms, Graphs and Optimization Symposium (LAGOS 2019). 18, 19, 48

Faria, Alexandre Frias; de Souza, Sérgio Ricardo e Silva, Carlos Alexandre. (2017). Comparação entre heurísticas construtivas e ils na solução do problema da k-partição de números. *Anais do XLIX SBPO*, Blumenau - Brasil. 16, 18, 47

Faria, Alexandre Frias; de Souza, Sérgio Ricardo e Silva, Carlos Alexandre. (2018)a. Variable neighborhood descent applied to multi-way number partitioning problem. *Electronic Notes in Discrete Mathematics*, v. 66, p. 103 – 110. ISSN 1571-0653. doi: https://doi.org/10.1016/j.endm.2018.03.014. URL `http://www.sciencedirect.com/science/article/pii/S157106531830060X`. 5th International Conference on Variable Neighborhood Search. 17, 18, 48

Faria, Alexandre Frias; de Souza, Sérgio Ricardo; Silva, Carlos Alexandre e de Sá, Elisangela Martins. (2018)b. Análise de modelos matemáticos para o problema da múltipla partição de números. *Anais do L SBPO*, Rio de Janeiro - Brasil. 17, 19, 47

Fischetti, Matteo e Lodi, Andrea. (2003). Local branching. *Mathematical programming*, v. 98, n. 1-3, p. 23–47. 16, 31

Gent, Ian P. e Walsh, Toby. (1998). Analysis of heuristics for number partitioning. *Computational Intelligence*, v. 14, n. 3, p. 430–451. 6, 14

Gomory, Ralph. (1960). An algorithm for the mixed integer problem. Relatório técnico, RAND CORP SANTA MONICA CA. 16

Graham, Ronald L. (1969). Bounds on multiprocessing timing anomalies. *SIAM Journal on Applied Mathematics*, v. 17, n. 2, p. 416–429. 14

Hacibeyoglu, M.; Tongur, V. e Alaykiran, K. October(2014). Solving the bi-dimensional two-way number partitioning problem with heuristic algorithms. *Proceedings of the 2014 IEEE 8th International Conference on Application of Infor-*

mation and Communication Technologies (AICT), p. 1–5, October(2014). doi: 10.1109/ICAICT.2014.7035916. 39

Hacibeyoglu, Mehmet; Alaykiran, Kemal; Acilar, Ayse Merve; Tongur, Vahit e Ulker, Erkan. December(2018). A comparative analysis of metaheuristic approaches for multidimensional two-way number partitioning problem. *Arabian Journal for Science and Engineering*, v. 43, n. 12, p. 7499–7520. doi: 10.1007/s13369-018-3155-9. 3, 17, 39

Hooker, John. (2011). *Logic-based methods for optimization: combining optimization and constraint satisfaction*, volume 2. John Wiley & Sons. 3, 20

Horowitz, Ellis e Sahni, Sartaj. (1974). Computing partitions with applications to the knapsack problem. *Journal of the ACM (JACM)*, v. 21, n. 2, p. 277–292. 14

Karmarkar, Narendra e Karp, Richard M. (1982). The differencing method of set partition. Report UCB/CSD 81/113, Computer Science Division, University of California, Berkeley, CA. 7, 14

Karp, Richard M. (1972). Reducibility among combinatorial problems. Miller, Raymond E.; Thatcher, James W. e Bohlinger, Jean D., editors, *Proceedings of a Symposium on the Complexity of Computer Computations*, p. 85–103. Springer US, (1972). 2, 6, 14, 16, 29

Kojić, Jelena. (2010). Integer linear programming model for multidimensional two-way number partitioning problem. *Computers & Mathematics with Applications*, v. 60, n. 8, p. 2302–2308. vi, viii, 4, 8, 15, 17, 19, 20, 25, 26, 27, 28, 37, 39, 40, 44, 61, 62, 63, 64, 65, 66

Korf, Richard E. (1998). A complete anytime algorithm for number partitioning. *Artificial Intelligence*, v. 106, n. 2, p. 181–203. 14, 15

Korf, Richard E. (2009). Multi-way number partitioning. *IJCAI*, p. 538–543. Citeseer, (2009). 7, 15

Korf, Richard E; Schreiber, Ethan L e Moffitt, Michael D. (2013). Optimal sequential multi-way number partitioning. *International Symposium on Artificial Intelligence and Mathematics (ISAIM-2014)*, (2013). 7, 15

Kratica, Jozef; Kojić, Jelena e Savić, Aleksandar. (2014). Two metaheuristic approaches for solving multidimensional two-way number partitioning problem. *Computers & Operations Research*, v. 46, p. 59 – 68. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2014.01.003. 15, 17, 39

Kreher, Donald L e Stinson, Douglas R. (1998). *Combinatorial algorithms: generation, enumeration, and search*, volume 7. CRC press. 25

Lima, Elon Lages. (2004). *Análise real*. IMPA, Rio de Janeiro, Brasil. 5

Maniezzo, Vittorio; Stützle, Thomas e Voß, Stefan, editors. (2010). *Matheuristics: Hybridizing Metaheuristics and Mathematical Programming*, volume 10 of *Annals of Information Systems*. Springer US, Boston, MA, EUA. 16

Moffitt, Michael D. (2013). Search strategies for optimal multi-way number partitioning. *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, p. 623–629. AAAI Press, (2013). 15

Nijenhuis, Albert e Wilf, Herbert S. (2014). *Combinatorial algorithms: for computers and calculators*. Elsevier. 25

Nikolic, Bojan; Djukanović, Marko e Matic, Dragan. April(2022). New mixed-integer linear programming model for solving the multidimensional multi-way number partitioning problem. *Journal of Computational and Applied Mathematics*, v. 41. doi: 10.1007/s40314-022-01825-2. 4, 17, 19, 37, 44, 45, 47

Orlov, Michael. (2002). Efficient generation of set partitions. Tech. rep, University of Ulm, Alemanha. URL `http://www.informatik.uni-ulm.de/ni/Lehre/WS03/DMM/Software/partitions.pdf`. 29

Pedroso, João Pedro e Kubo, Mikio. (2010). Heuristics and exact methods for number partitioning. *European Journal of Operational Research*, v. 202, n. 1, p. 73–81. 6

Piccialli, Veronica; Sudoso, Antonio M e Wiegele, Angelika. (2022). Sos-sdp: an exact solver for minimum sum-of-squares clustering. *INFORMS Journal on Computing*, v. 34, n. 4. doi: 10.1287/ijoc.2022.1166. 3

Pop, P. C. e Matei, O. (2013)a. A genetic algorithm approach for the multidimensional two-way number partitioning problem. Nicosia, Giuseppe e Pardalos, Panos, editors, *Learning and Intelligent Optimization*, p. 81–86, Berlin, Heidelberg. Springer. 17

Pop, Petrică C e Matei, Oliviu. (2013)b. A memetic algorithm approach for solving the multidimensional multi-way number partitioning problem. *Applied Mathematical Modelling*, v. 37, n. 22, p. 9191–9202. 3, 4, 10, 12, 15, 17, 20, 39

R Core Team,. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, (2018). URL `https://www.R-project.org/`. 38

Rodrigues, Filipe; Agra, Agostinho; Hvattum, Lars Magnus e Requejo, C. 06(2022). Weighted iterated local branching for mathematical programming problems with binary variables. *Journal of Heuristics*, v. 28, p. 1–22. doi: 10.1007/s10732-022-09496-2. 19

Rodriguez, Francisco J.; Glover, Fred; García-Martínez, Carlos; Martí, Rafael e Lozano, Manuel. (2017). Grasp with exterior path-relinking and restricted local search for the multidimensional two-way number partitioning problem. *Computers & Operations Research*, v. 78, p. 243–254. ISSN 0305-0548. doi: https://doi.org/10.1016/j.cor.2016.09.005. URL `https://www.sciencedirect.com/science/article/pii/S0305054816302209`. 15, 17

Santucci, Valentino; Baioletti, Marco e Di Bari, Gabriele. (2021). An improved memetic algebraic differential evolution for solving the multidimensional two-way number partitioning problem. *Expert Systems with Applications*, v. 178, p. 114938. ISSN 0957-4174. doi: https://doi.org/10.1016/j.eswa.2021.114938. 17, 19

Santucci, Valentino; Baioletti, Marco; Di Bari, Gabriele e Milani, Alfredo. (2019). A binary algebraic differential evolution for the multidimensional two-way number partitioning problem. Liefooghe, Arnaud e Paquete, Luís, editors, *Evolutionary Computation in Combinatorial Optimization*, p. 17–32, Cham. Springer International Publishing. ISBN 978-3-030-16711-0. 17

Schreiber, Ethan L. *Optimal Multi-Way Number Partitioning*. PhD thesis, University of California Los Angeles, (2014). 7, 15

Schreiber, Ethan L e Korf, Richard E. (2014). Cached iterative weakening for optimal multi-way number partitioning. *Proceedings of the Twenty-Eighth Annual Conference on Artificial Intelligence (AAAI-14) Quebec City, Canada*, (2014). 15

Schrijver, A. (1980). On cutting planes. Hammer, Peter L., editor, *Combinatorics 79*, volume 9 of *Annals of Discrete Mathematics*, p. 291–296. Elsevier. doi: https://doi.org/10.1016/S0167-5060(08)70085-2. URL `http://www.sciencedirect.com/science/article/pii/S0167506008700852`. 3

Schroeppel, Richard e Shamir, Adi. (1981). A $T = O(2^{n/2})$, $S = O(2^{n/4})$ algorithm for certain NP-complete problems. *SIAM journal on Computing*, v. 10, n. 3, p. 456–464. 14

Stanley, Richard P. (1997). *Enumerative Combinatorics. Vol. 1, vol. 49 of Cambridge Studies in Advanced Mathematics*, volume 1 of *Cambridge Studies in Advanced Mathematics*. Cambridge University Press, Cambridge, 2nd edição. 5

Williams, H Paul. (2013). *Model building in mathematical programming*. John Wiley & Sons. 3, 16, 20

Apêndice A

Comparação entre modelos

A.1 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 50$

Tabela A.1: Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 50$

	Model (MOD_{Kojic})				Model (MOD_k) using $k = 2$			
Instance	Status	UB	LB	time(s)	Status	UB	LB	time(s)
2_50_2a	Feasible	7.504	0	1796.98	Feasible	4.448	0	1796.72
2_50_2b	Feasible	112.559	0	1799.79	Feasible	111.611	0	1799.94
2_50_2c	Feasible	4.467	0	1798.6	Feasible	7.513	0	1799.95
2_50_2d	Feasible	0.774971	0	1799.95	Feasible	222.954	0	1798.97
2_50_2e	Feasible	6.947	0	1799.95	Feasible	9.015	0	1799.94
2_50_3a	Feasible	185.82	0	1799.92	Feasible	181.706	0	1799.4
2_50_3b	Feasible	137.887	0	1799.27	Feasible	137.887	0	1799.95
2_50_3c	Feasible	258.492	0	1799.41	Feasible	258.494	0	1799.95
2_50_3d	Feasible	88.831	0	1799.95	Feasible	87.563	0	1799.17
2_50_3e	Feasible	253.919	0	1799.96	Feasible	258.692	0	1799.94
2_50_4a	Feasible	1208.73	0	1799.93	Feasible	1220.93	0	1799.38
2_50_4b	Feasible	1279.75	0	1799.89	Feasible	1279.75	0	1799.9
2_50_4c	Feasible	946.785	0	1799.97	Feasible	943.731	0	1799.96
2_50_4d	Feasible	967.16	0	1799.96	Feasible	967.16	0	1799.36
2_50_4e	Feasible	843.079	0	1799.96	Feasible	852.229	0	1799.96
2_50_5a	Feasible	2523.86	0	1799.94	Feasible	2523.85	0	1799.5
2_50_5b	Feasible	1789.99	0	1799.95	Feasible	1789.99	0	1799.94
2_50_5c	Feasible	2561.65	0	1799.96	Feasible	2755.23	0	1799.95
2_50_5d	Feasible	2163.26	0	1799.97	Feasible	2560.62	0	1799.5
2_50_5e	Feasible	2021.72	0	1799.96	Feasible	2030.88	0	1799.93
2_50_10a	Feasible	17399.4	0	1799.59	Feasible	17393.3	0	1799.03
2_50_10b	Feasible	18550.1	0	1799.67	Feasible	18556.2	0	1799.64
2_50_10c	Feasible	17827.7	0	1799.74	Feasible	17833.9	0	1799.74
2_50_10d	Feasible	17055.1	0	1799.74	Feasible	17055.1	0	1799.19
2_50_10e	Feasible	15108.7	0	1799.73	Feasible	15114.8	0	1799.74
2_50_15a	Feasible	30311	10668.8	1799.44	Feasible	30311	12420.7	1798.83
2_50_15b	Feasible	33499	7874.82	1799.32	Feasible	32625.3	10886.2	1799.55
2_50_15c	Feasible	27787.8	11994.7	1799.65	Feasible	27787.8	12703.5	1799.61
2_50_15d	Feasible	33823.9	14160.6	1799.44	Feasible	33823.9	14168.3	1798.67
2_50_15e	Feasible	31718.8	14641.3	1799.4	Feasible	31718.8	13597.8	1799.58
2_50_20a	Feasible	52826.3	35826.2	1799.58	Feasible	52826.3	36452.6	1799.21
2_50_20b	Feasible	51917.9	40903.7	1799.77	Feasible	51917.9	39533.9	1799.51
2_50_20c	Optimal	50560.9	50555.8	1271.12	Feasible	50560.9	45183.7	1799.75
2_50_20d	Feasible	53956	40208.7	1799.63	Feasible	53956	40775.8	1798.73
2_50_20e	Feasible	48281.5	38447.7	1799.7	Feasible	48281.5	36424.2	1799.74

A.2 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 100$

Tabela A.2: Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 100$

	Model (MOD_{Kojic})				Model (MOD_k) using $k = 2$			
Instance	Status	UB	LB	time(s)	Status	UB	LB	time(s)
2_100_2a	Feasible	3.669	0	1799.92	Feasible	4.159	0	1799.4
2_100_2b	Feasible	0.925	0	1799.93	Feasible	1.079	0	1799.92
2_100_2c	Feasible	2.176	0	1799.94	Feasible	6.442	0	1799.9
2_100_2d	Feasible	1.343	0	1799.94	Feasible	0.509277	0	1799.3
2_100_2e	Feasible	4.51	0	1799.91	Feasible	3.204	0	1799.9
2_100_3a	Feasible	151.339	0	1799.93	Feasible	142.179	0	1799.48
2_100_3b	Feasible	169.791	0	1799.94	Feasible	143.062	0	1799.96
2_100_3c	Feasible	6.103	0	1799.95	Feasible	154.879	0	1799.93
2_100_3d	Feasible	3.051	0	1799.96	Feasible	6.101	0	1799.24
2_100_3e	Feasible	108.622	0	1799.93	Feasible	105.57	0	1799.94
2_100_4a	Feasible	939.942	0	1799.84	Feasible	916.488	0	1799.67
2_100_4b	Feasible	1330.55	0	1799.89	Feasible	1348.86	0	1799.92
2_100_4c	Feasible	785.547	0	1799.93	Feasible	782.501	0	1799.92
2_100_4d	Feasible	928.469	0	1799.88	Feasible	952.881	0	1799.17
2_100_4e	Feasible	1241.95	0	1799.88	Feasible	1248.05	0	1799.88
2_100_5a	Feasible	2844.34	0	1799.72	Feasible	2847.39	0	1799.66
2_100_5b	Feasible	1908.29	0	1799.83	Feasible	1914.38	0	1799.9
2_100_5c	Feasible	1282.8	0	1799.88	Feasible	1285.85	0	1799.96
2_100_5d	Feasible	926.167	0	1799.92	Feasible	913.957	0	1799.21
2_100_5e	Feasible	1983.71	0	1799.84	Feasible	1983.7	0	1799.91
2_100_10a	Feasible	14184.2	0	1798.87	Feasible	14181.2	0	1799.36
2_100_10b	Feasible	15382.8	0	1799.34	Feasible	15379.7	0	1799.59
2_100_10c	Feasible	16068.8	0	1799.56	Feasible	16062.7	0	1799.6
2_100_10d	Feasible	17493.3	0	1799.64	Feasible	17493.2	0	1798.27
2_100_10e	Feasible	15761.7	0	1799.64	Feasible	15749.5	0	1799.57
2_100_15a	Feasible	30478.6	0	1789.9	Feasible	30487.8	0	1789.28
2_100_15b	Feasible	29768.4	0	1790.81	Feasible	25609.7	0	1790.78
2_100_15c	Feasible	22243.7	0	1793.04	Feasible	22255.9	0	1792.58
2_100_15d	Feasible	30072.6	0	1789.66	Feasible	31065.6	0	1787.15
2_100_15e	Feasible	27579.4	0	1790.88	Feasible	24487.2	0	1792.21
2_100_20a	Feasible	49216.5	0	1789.26	Feasible	50336.6	0	1788.78
2_100_20b	Feasible	46688.4	0	1789.4	Feasible	47534.3	0	1789.4
2_100_20c	Feasible	49807.5	0	1790.69	Feasible	47662.6	0	1789.38
2_100_20d	Feasible	43780.6	0	1789	Feasible	43786.7	0	1788.32
2_100_20e	Feasible	44792.4	0	1788.96	Feasible	43926.3	0	1789.2

A.3 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 200$

Tabela A.3: Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 200$

	Model (MOD_{Kojic})				Model (MOD_k) using $k = 2$			
Instance	Status	UB	LB	time(s)	Status	UB	LB	time(s)
2_200_2a	Feasible	1.373	0	1799.91	Feasible	5.547	0	1799.67
2_200_2b	Feasible	1.368	0	1799.92	Feasible	3.214	0	1799.91
2_200_2c	Feasible	0.587074	0	1799.92	Feasible	1.236	0	1799.94
2_200_2d	Feasible	0.251879	0	1799.93	Feasible	1.686	0	1799
2_200_2e	Feasible	0.502961	0	1799.91	Feasible	1.238	0	1799.96
2_200_3a	Feasible	116.614	0	1799.95	Feasible	81.695	0	1799.72
2_200_3b	Feasible	105.565	0	1799.94	Feasible	158.341	0	1799.9
2_200_3c	Feasible	100.995	0	1799.93	Feasible	77.263	0	1799.9
2_200_3d	Feasible	125.962	0	1799.92	Feasible	100.452	0	1798.96
2_200_3e	Feasible	119.666	0	1799.89	Feasible	92.379	0	1799.89
2_200_4a	Feasible	898.189	0	1799.8	Feasible	907.337	0	1799.59
2_200_4b	Feasible	995.062	0	1799.72	Feasible	992.02	0	1799.91
2_200_4c	Feasible	1072.39	0	1799.83	Feasible	1087.64	0	1799.76
2_200_4d	Feasible	1082.54	0	1799.88	Feasible	1010.67	0	1799.19
2_200_4e	Feasible	18.31	0	1799.85	Feasible	1042.52	0	1799.88
2_200_5a	Feasible	1863.36	0	1799.46	Feasible	1832.84	0	1799.49
2_200_5b	Feasible	1483.64	0	1799.8	Feasible	1474.48	0	1799.69
2_200_5c	Feasible	1758.8	0	1799.7	Feasible	1763.96	0	1799.82
2_200_5d	Feasible	1489.74	0	1799.81	Feasible	2268.29	0	1799
2_200_5e	Feasible	2268.28	0	1799.79	Feasible	2265.23	0	1799.63
2_200_10a	Feasible	18863.4	0	1798.98	Feasible	14272.3	0	1798.61
2_200_10b	Feasible	16706.9	0	1798.89	Feasible	16694.7	0	1798.88
2_200_10c	Feasible	12766.2	0	1799	Feasible	17290.2	0	1798.76
2_200_10d	Feasible	12791.1	0	1799.07	Feasible	11745.5	0	1798.44
2_200_10e	Feasible	16297.2	0	1798.84	Feasible	14621.6	0	1799.11
2_200_15a	Feasible	28911.2	0	1791.2	Feasible	27746.6	0	1790.81
2_200_15b	Feasible	29275.1	0	1789.35	Feasible	31319.6	0	1791
2_200_15c	Feasible	27604.1	0	1792.91	Feasible	30788.3	0	1790.34
2_200_15d	Feasible	32500.1	0	1792.13	Feasible	30035.2	0	1788.33
2_200_15e	Feasible	32784.7	0	1793.49	Feasible	28660.8	0	1787.76
2_200_20a	Feasible	49890	0	1795.33	Feasible	46416.1	0	1794.83
2_200_20b	Feasible	47532.1	0	1794.31	Feasible	51462.9	0	1794.82
2_200_20c	Feasible	50830.7	0	1795.35	Feasible	50139	0	1792.71
2_200_20d	Feasible	46651.6	0	1793.38	Feasible	45235.4	0	1796.2
2_200_20e	Feasible	37156.1	0	1797.55	Feasible	49838.6	0	1793.98

A.4 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 300$

Tabela A.4: Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 300$

	Model (MOD_{Kojic})				Model (MOD_k) using $k = 2$			
Instance	Status	UB	LB	time(s)	Status	UB	LB	time(s)
2_300_2a	Feasible	0.455	0	1799.93	Feasible	0.132022	0	1799.72
2_300_2b	Feasible	4.578	0	1799.93	Feasible	3.67	0	1799.92
2_300_2c	Feasible	2.646	0	1799.94	Feasible	2.382	0	1799.93
2_300_2d	Feasible	2.131	0	1799.91	Feasible	3.508	0	1799.24
2_300_2e	Feasible	0.746	0	1799.91	Feasible	2.456	0	1799.95
2_300_3a	Feasible	3.049	0	1799.94	Feasible	75.294	0	1799.65
2_300_3b	Feasible	90.855	0	1799.9	Feasible	164.825	0	1799.93
2_300_3c	Feasible	112.914	0	1799.92	Feasible	115.962	0	1799.92
2_300_3d	Feasible	125.236	0	1799.93	Feasible	88.248	0	1799.41
2_300_3e	Feasible	53.453	0	1799.94	Feasible	68.117	0	1799.92
2_300_4a	Feasible	1159.3	0	1799.72	Feasible	509.55	0	1799.62
2_300_4b	Feasible	4.572	0	1799.97	Feasible	1040.59	0	1799.63
2_300_4c	Feasible	900.614	0	1799.82	Feasible	816.459	0	1799.8
2_300_4d	Feasible	13.731	0	1799.91	Feasible	960.002	0	1799.11
2_300_4e	Feasible	1034.2	0	1799.88	Feasible	1104.97	0	1799.8
2_300_5a	Feasible	2805.02	0	1799.46	Feasible	1909.82	0	1799.68
2_300_5b	Feasible	1866.07	0	1799.48	Feasible	2047.72	0	1799.24
2_300_5c	Feasible	2744.77	0	1799.49	Feasible	1918.96	0	1799.58
2_300_5d	Feasible	1844.71	0	1799.4	Feasible	2226.06	0	1799.13
2_300_5e	Feasible	7.629	0	1799.81	Feasible	2230.65	0	1799.45
2_300_10a	Feasible	11863.5	0	1781.75	Feasible	11842.1	0	1799.01
2_300_10b	Feasible	19635.1	0	1798.83	Feasible	17988.5	0	1799.02
2_300_10c	Feasible	21692.7	0	1796.43	Feasible	20207.4	0	1761.17
2_300_10d	Feasible	21040.9	0	1799.02	Feasible	18249.7	0	1798.48
2_300_10e	Feasible	23732.6	0	1798.87	Feasible	13091.3	0	1798.71
2_300_15a	Feasible	33756.7	0	1787.96	Feasible	30466.8	0	1790.37
2_300_15b	Feasible	30245.3	0	1788.28	Feasible	37664.2	0	1779.76
2_300_15c	Feasible	27543.1	0	1788.7	Feasible	36755	0	1783.11
2_300_15d	Feasible	29571.5	0	1789.36	Feasible	35387.5	0	1779.54
2_300_15e	Feasible	36320.2	0	1788.11	Feasible	30198.6	0	1780.96
2_300_20a	Feasible	47270	0	1795.26	Feasible	46991.3	0	1799.12
2_300_20b	Feasible	55996.3	0	1797.31	Feasible	56535.8	0	1799.08
2_300_20c	Feasible	47315.5	0	1796.37	Feasible	43529.6	0	1791.61
2_300_20d	Feasible	58791.4	0	1799.21	Feasible	49340	0	1795.37
2_300_20e	Feasible	55596.8	0	1799.18	Feasible	50129	0	1798.97

A.5 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 400$

Tabela A.5: Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 400$

	Model (MOD_{Kojic})				Model (MOD_k) using $k = 2$			
Instance	Status	UB	LB	time(s)	Status	UB	LB	time(s)
2_400_2a	Feasible	1.972	0	1799.94	Feasible	116.347	0	1799.93
2_400_2b	Feasible	3.415	0	1799.93	Feasible	2.225	0	1799.93
2_400_2c	Feasible	1.571	0	1799.89	Feasible	1.133	0	1799.7
2_400_2d	Feasible	1.367	0	1799.9	Feasible	1.787	0	1799.82
2_400_2e	Feasible	0.715	0	1799.93	Feasible	2.886	0	1799.91
2_400_3a	Feasible	118.141	0	1799.9	Feasible	75.556	0	1799.93
2_400_3b	Feasible	135.906	0	1799.88	Feasible	136.046	0	1799.9
2_400_3c	Feasible	6.104	0	1799.89	Feasible	91.76	0	1799.68
2_400_3d	Feasible	135.912	0	1799.87	Feasible	115.328	0	1799.91
2_400_3e	Feasible	149.167	0	1799.88	Feasible	78.609	0	1799.91
2_400_4a	Feasible	881.653	0	1799.73	Feasible	880.218	0	1799.73
2_400_4b	Feasible	891.301	0	1799.54	Feasible	912.663	0	1799.6
2_400_4c	Feasible	610.67	0	1799.81	Feasible	743.438	0	1799.63
2_400_4d	Feasible	1480.13	0	1799.58	Feasible	1252.97	0	1799.81
2_400_4e	Feasible	514.393	0	1799.78	Feasible	819.661	0	1799.71
2_400_5a	Feasible	10.686	0	1799.9	Feasible	1890.48	0	1799.42
2_400_5b	Feasible	2301.44	0	1798.37	Feasible	1915.99	0	1799.27
2_400_5c	Feasible	2544.21	0	1799.34	Feasible	2539.38	0	1799.12
2_400_5d	Feasible	1486.69	0	1799.11	Feasible	2033.83	0	1799.32
2_400_5e	Feasible	1971.49	0	1799.65	Feasible	2534.14	0	1799.35
2_400_10a	Feasible	22542.5	0	1798.86	Feasible	19035.6	0	1798.91
2_400_10b	Feasible	20077.6	0	1797.43	Feasible	20364.7	0	1798.84
2_400_10c	Feasible	14403.6	0	1798.84	Feasible	14412.8	0	1798.28
2_400_10d	Feasible	19027	0	1798.91	Feasible	18021.9	0	1798.64
2_400_10e	Feasible	20773.1	0	1798.73	Feasible	20445.2	0	1798.72
2_400_15a	Feasible	28656.7	0	1788.97	Feasible	29555.5	0	1788.65
2_400_15b	Feasible	30357.8	0	1788.05	Feasible	32100.9	0	1789.32
2_400_15c	Feasible	36823	0	1788.23	Feasible	31755.1	0	1788.83
2_400_15d	Feasible	29978.8	0	1789.11	Feasible	32766.1	0	1788.23
2_400_15e	Feasible	32964.4	0	1788.81	Feasible	29317.5	0	1788.55
2_400_20a	Feasible	47452.9	0	1792.29	Feasible	51048.1	0	1797.88
2_400_20b	Feasible	50420	0	1791.65	Feasible	47661.2	0	1797.78
2_400_20c	Feasible	49353.1	0	1792.05	Feasible	56310.1	0	1798.62
2_400_20d	Feasible	44643.2	0	1792.76	Feasible	52217.2	0	1798.29
2_400_20e	Feasible	53079.7	0	1793.01	Feasible	45866	0	1792.36

A.6 Modelos (MOD_{Kojic}) e (MOD_k) para $k = 2$ e $n = 500$

Tabela A.6: Resultados: modelos (MOD_{Kojic}) e (MOD_k) usando $k = 2$ e $n = 500$

	Model (MOD_{Kojic})				Model (MOD_k) using $k = 2$			
Instance	Status	UB	LB	time(s)	Status	UB	LB	time(s)
2_500_2a	Feasible	2.59	0	1799.94	Feasible	5.909	0	1799.9
2_500_2b	Feasible	2.902	0	1798.2	Feasible	4.342	0	1799.89
2_500_2c	Feasible	4.435	0	1799.85	Feasible	1.201	0	1799.67
2_500_2d	Feasible	2.36	0	1799.92	Feasible	403.755	0	1799.93
2_500_2e	Feasible	0.567	0	1799.94	Feasible	0.663	0	1799.92
2_500_3a	Feasible	107.081	0	1799.88	Feasible	52.991	0	1799.93
2_500_3b	Feasible	152.894	0	1798.1	Feasible	86.54	0	1799.9
2_500_3c	Feasible	78.459	0	1799.9	Feasible	162.631	0	1799.48
2_500_3d	Feasible	116.253	0	1799.91	Feasible	97.261	0	1799.92
2_500_3e	Feasible	97.935	0	1799.88	Feasible	62.009	0	1799.92
2_500_4a	Feasible	482.635	0	1799.77	Feasible	751.607	0	1799.53
2_500_4b	Feasible	654.78	0	1798.37	Feasible	749.372	0	1799.73
2_500_4c	Feasible	570.741	0	1799.58	Feasible	1104.98	0	1798.57
2_500_4d	Feasible	1088.59	0	1799.84	Feasible	606.985	0	1799.78
2_500_4e	Feasible	1225.18	0	1799.78	Feasible	701.553	0	1799.33
2_500_5a	Feasible	1709.02	0	1799.75	Feasible	887.853	0	1799.3
2_500_5b	Feasible	2083.46	0	1796.74	Feasible	1235.63	0	1799.62
2_500_5c	Feasible	1614.87	0	1799.4	Feasible	2522.59	0	1798.42
2_500_5d	Feasible	2232.3	0	1799.45	Feasible	1690.47	0	1798.9
2_500_5e	Feasible	1890.48	0	1799.41	Feasible	1752.8	0	1799.26
2_500_10a	Feasible	17624.1	0	1764.56	Feasible	19184.9	0	1765.61
2_500_10b	Feasible	17966	0	1760.21	Feasible	19208.4	0	1768.18
2_500_10c	Feasible	20602.2	0	1763.96	Feasible	19555.8	0	1763.64
2_500_10d	Feasible	22524.7	0	1798.4	Feasible	19944.4	0	1797.38
2_500_10e	Feasible	23812.6	0	1796.19	Feasible	15217.2	0	1798.56
2_500_15a	Feasible	33077.8	0	1784.89	Feasible	31842.2	0	1789.5
2_500_15b	Feasible	33210.2	0	1787.21	Feasible	32756.9	0	1788.97
2_500_15c	Feasible	33929.8	0	1788.97	Feasible	32709.3	0	1788.01
2_500_15d	Feasible	35875.1	0	1789.24	Feasible	34283.4	0	1789.04
2_500_15e	Feasible	35861.5	0	1788.95	Feasible	26418.7	0	1788.86
2_500_20a	Feasible	54786.9	0	1793.32	Feasible	47784.8	0	1792.52
2_500_20b	Feasible	44374.9	0	1792.31	Feasible	53358.3	0	1793.78
2_500_20c	Feasible	39986.4	0	1793.36	Feasible	49051.1	0	1791.26
2_500_20d	Feasible	52962.7	0	1792.8	Feasible	52449.4	0	1792.34
2_500_20e	Feasible	53636.3	0	1792.87	Feasible	47057	0	1792.47

A.7 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 50$

Tabela A.7: Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 50$

	$k = 3$				$k = 4$			
Instance	Status	UB	LB	Time(s)	Status	UB	LB	Time(s)
3_50_2a	Feasible	108.376	0	1798.15	Feasible	759.295	0	1797.98
3_50_2b	Feasible	233.444	0	1799.9	Feasible	1016.41	0	1798.67
3_50_2c	Feasible	153.758	0	1799.98	Feasible	326.054	0	1799.67
3_50_2d	Feasible	275.211	0	1799.92	Feasible	817.197	0	1799.72
3_50_2e	Feasible	199.015	0	1799.75	Feasible	788.127	0	1799.58
3_50_3a	Feasible	1727.11	0	1799.88	Feasible	4561.99	0	1799.56
3_50_3b	Feasible	1619.99	0	1799.93	Feasible	4671.89	0	1799.44
3_50_3c	Feasible	1637.95	0	1799.76	Feasible	3653.37	0	1799.2
3_50_3d	Feasible	1531.66	0	1799.64	Feasible	3810.06	0	1799.6
3_50_3e	Feasible	1174.1	0	1799.63	Feasible	6097.69	0	1799.54
3_50_4a	Feasible	4843.93	0	1799.8	Feasible	7864.45	0	1799.58
3_50_4b	Feasible	4811.94	0	1799.6	Feasible	12383.4	0	1798.68
3_50_4c	Feasible	4512.36	0	1799.6	Feasible	10605.7	0	1798.49
3_50_4d	Feasible	3918.02	0	1799.8	Feasible	12398.4	0	1799.51
3_50_4e	Feasible	6685.06	0	1799.33	Feasible	10761	0	1799.62
3_50_5a	Feasible	10242.3	0	1799.61	Feasible	14442.1	0	1799.64
3_50_5b	Feasible	11453.3	0	1799.65	Feasible	17474.4	0	1798.78
3_50_5c	Feasible	10644.5	0	1799.68	Feasible	18586.2	0	1798.66
3_50_5d	Feasible	9918.96	0	1799.71	Feasible	20639	0	1799.57
3_50_5e	Feasible	10939.4	0	1799.23	Feasible	19519.9	0	1799.62
3_50_10a	Feasible	37497.1	0	1799.74	Feasible	52094.9	0	1799.03
3_50_10b	Feasible	38845.7	0	1799.7	Feasible	55590.9	0	1797.83
3_50_10c	Feasible	33154.1	0	1799.81	Feasible	47480.6	0	1797.65
3_50_10d	Feasible	31332.9	0	1799.71	Feasible	56615.4	0	1798.61
3_50_10e	Feasible	33244.7	0	1799.29	Feasible	58775	0	1799
3_50_15a	Feasible	65740.3	0	1799	Feasible	82603.6	5409.36	1799.13
3_50_15b	Feasible	67606.9	0	1798.83	Feasible	81396.8	4316.24	1798.48
3_50_15c	Feasible	63972.2	0	1799.16	Feasible	87756	3380.3	1798.03
3_50_15d	Feasible	65884.7	0	1799.27	Feasible	88641.8	4416	1799.01
3_50_15e	Feasible	53260	0	1798.58	Feasible	84652.7	5736.9	1799.09
3_50_20a	Feasible	86148.1	15876.5	1799.5	Feasible	97629.8	17292.4	1799.42
3_50_20b	Feasible	81418.2	17304.7	1799.3	Feasible	102400	4438.93	1798.47
3_50_20c	Feasible	88259.7	17472.9	1799.35	Feasible	108617	14259.2	1798.15
3_50_20d	Feasible	86192.5	18418	1799.26	Feasible	102946	19955	1799.35
3_50_20e	Feasible	87280.8	10118.4	1798.72	Feasible	104535	13843.4	1799.46

A.8 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 100$

Tabela A.8: Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3, 4\}$ e $n = 100$

	$k = 3$				$k = 4$			
Instance	Status	UB	LB	Time(s)	Status	UB	LB	Time(s)
3_100_2a	Feasible	230.358	0	1727.21	Feasible	594.436	0	1799.27
3_100_2b	Feasible	188.676	0	1722.81	Feasible	744.755	0	1798.59
3_100_2c	Feasible	138.306	0	1723.01	Feasible	586.926	0	1797.22
3_100_2d	Feasible	229.948	0	1716.48	Feasible	638.946	0	1799.32
3_100_2e	Feasible	259.944	0	1749.81	Feasible	924.913	0	1760.27
3_100_3a	Feasible	2121.11	0	1772.62	Feasible	3929.83	0	1799.34
3_100_3b	Feasible	1671.87	0	1761.11	Feasible	4402.19	0	1798.91
3_100_3c	Feasible	1202.61	0	1780.96	Feasible	3722.65	0	1797.82
3_100_3d	Feasible	1171.16	0	1772.71	Feasible	3738.01	0	1799.45
3_100_3e	Feasible	1980.93	0	1749.35	Feasible	3898.88	0	1799.32
3_100_4a	Feasible	3493.16	0	1799.28	Feasible	9997.65	0	1799.45
3_100_4b	Feasible	5293.17	0	1799.25	Feasible	7934.6	0	1799.15
3_100_4c	Feasible	4503.36	0	1799.29	Feasible	9129.63	0	1797.8
3_100_4d	Feasible	5347.41	0	1798.39	Feasible	7630.58	0	1799.55
3_100_4e	Feasible	4933.83	0	1798.75	Feasible	8781.43	0	1799.47
3_100_5a	Feasible	7234.48	0	1799.5	Feasible	15472.7	0	1799.49
3_100_5b	Feasible	9195	0	1799.33	Feasible	18923.1	0	1799.23
3_100_5c	Feasible	8792.01	0	1799.52	Feasible	18709	0	1798.66
3_100_5d	Feasible	10214.8	0	1799.34	Feasible	15904.6	0	1799.53
3_100_5e	Feasible	8241.72	0	1798.76	Feasible	17489.9	0	1798.94
3_100_10a	Feasible	38760.5	0	1799.7	Feasible	51816.8	0	1798.95
3_100_10b	Feasible	34028.2	0	1799.75	Feasible	56949.7	0	1798.99
3_100_10c	Feasible	37067.5	0	1799.75	Feasible	48903.4	0	1798.84
3_100_10d	Feasible	36873.9	0	1799.79	Feasible	44704.3	0	1799.03
3_100_10e	Feasible	37968.2	0	1799.17	Feasible	51129.9	0	1799.73
3_100_15a	Feasible	58854	0	1799.8	Feasible	78459	0	1799.13
3_100_15b	Feasible	62369.2	0	1799.84	Feasible	74455	0	1798.89
3_100_15c	Feasible	72196.2	0	1799.77	Feasible	77818.4	0	1798.38
3_100_15d	Feasible	57447.8	0	1799.73	Feasible	84866.8	0	1799.09
3_100_15e	Feasible	61952.9	0	1799.46	Feasible	84472.4	0	1799.07
3_100_20a	Feasible	81959.9	0	1799.22	Feasible	95612.7	0	1799.46
3_100_20b	Feasible	89503.4	0	1799.31	Feasible	91895.1	0	1799.15
3_100_20c	Feasible	79363.2	0	1799.24	Feasible	107789	0	1798.77
3_100_20d	Feasible	84352.8	0	1799.43	Feasible	107969	0	1799.34
3_100_20e	Feasible	85619.1	0	1799.07	Feasible	98138.3	0	1799.45

A.9 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 200$

Tabela A.9: Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 200$

	$k = 3$				$k = 4$			
Instance	Status	UB	LB	Time(s)	Status	UB	LB	Time(s)
3_200_2b	Feasible	249.313	0	1714.96	Feasible	629.785	0	1773.3
3_200_2c	Feasible	169.712	0	1712.18	Feasible	900.571	0	1768.18
3_200_2d	Feasible	72.815	0	1716.7	Feasible	650.07	0	1796.61
3_200_2e	Feasible	69.019	0	1714.56	Feasible	351.272	0	1795.5
3_200_3a	Feasible	1408.54	0	1736.42	Feasible	2032.06	0	1799.11
3_200_3b	Feasible	1409.48	0	1736.63	Feasible	5030.79	0	1794.29
3_200_3c	Feasible	1839.59	0	1736.47	Feasible	3206.77	0	1798.66
3_200_3d	Feasible	1865.9	0	1750.44	Feasible	3086.72	0	1799.36
3_200_3e	Feasible	2925.48	0	1735.87	Feasible	2980.31	0	1799.28
3_200_4a	Feasible	3680.86	0	1789.42	Feasible	9780.88	0	1798.97
3_200_4b	Feasible	5473.03	0	1767.65	Feasible	8490.88	0	1798.76
3_200_4c	Feasible	5273.1	0	1771.79	Feasible	7903.95	0	1798.84
3_200_4d	Feasible	2875.36	0	1770.78	Feasible	5749.82	0	1799.36
3_200_4e	Feasible	3988.26	0	1769.89	Feasible	9247.18	0	1799.48
3_200_5a	Feasible	8274.47	0	1799.13	Feasible	14249.7	0	1799.27
3_200_5b	Feasible	8489.2	0	1799.23	Feasible	14030.2	0	1798.04
3_200_5c	Feasible	10119.1	0	1799.21	Feasible	10720.3	0	1799.14
3_200_5d	Feasible	6974.41	0	1799.28	Feasible	14026.7	0	1799.16
3_200_5e	Feasible	10487.1	0	1798.89	Feasible	14228.1	0	1799.65
3_200_10a	Feasible	34901.8	0	1799.7	Feasible	55588.9	0	1799.06
3_200_10b	Feasible	31203.6	0	1799.69	Feasible	55113.2	0	1799.34
3_200_10c	Feasible	37618.4	0	1799.71	Feasible	48135.4	0	1798.47
3_200_10d	Feasible	33559	0	1799.69	Feasible	49832.5	0	1799.73
3_200_10e	Feasible	40131	0	1799.36	Feasible	56188.4	0	1799.74
3_200_15a	Feasible	59119.6	0	1799.76	Feasible	80413.3	0	1799.52
3_200_15b	Feasible	61926.6	0	1799.69	Feasible	69755.6	0	1799.69
3_200_15c	Feasible	63285.8	0	1799.75	Feasible	79558.3	0	1799.21
3_200_15d	Feasible	64840.9	0	1799.83	Feasible	75775.8	0	1799.13
3_200_15e	Feasible	62643.9	0	1799.58	Feasible	77088.9	0	1799.74
3_200_20a	Feasible	80097	0	1799.37	Feasible	100031	0	1799.6
3_200_20b	Feasible	82983.8	0	1799.89	Feasible	100805	0	1799.48
3_200_20c	Feasible	79399.4	0	1799.88	Feasible	106623	0	1799.21
3_200_20d	Feasible	84969.7	0	1799.87	Feasible	103685	0	1799.45
3_200_20e	Feasible	82492.7	0	1799.62	Feasible	102098	0	1799.7

A.10 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 300$

Tabela A.10: Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 3000$

	$k=3$				$k=4$			
Instance	Status	UB	LB	Time(s)	Status	UB	LB	Time(s)
3_300_2a	Feasible	103.03	0	1799.47	Feasible	313.148	0	1799.11
3_300_2b	Feasible	317.844	0	1715.54	Feasible	666.654	0	1798.94
3_300_2c	Feasible	138.287	0	1706.92	Feasible	351.477	0	1756.37
3_300_2d	Feasible	153.483	0	1706.58	Feasible	823.647	0	1762.54
3_300_2e	Feasible	138.752	0	1714.41	Feasible	269.92	0	1799.27
3_300_3a	Feasible	1471.47	0	1735.59	Feasible	2440.1	0	1799.04
3_300_3b	Feasible	1444.62	0	1735.26	Feasible	3096.27	0	1798.88
3_300_3c	Feasible	1739.72	0	1737.47	Feasible	1688.65	0	1798.65
3_300_3d	Feasible	1276.1	0	1733.72	Feasible	3893.89	0	1799.17
3_300_3e	Feasible	1645.78	0	1729.11	Feasible	2992.1	0	1799.23
3_300_4a	Feasible	3279.57	0	1774.14	Feasible	10171.6	0	1798.83
3_300_4b	Feasible	4905.09	0	1765.96	Feasible	7831.64	0	1798.69
3_300_4c	Feasible	4440.1	0	1771.05	Feasible	11128.2	0	1798.67
3_300_4d	Feasible	4221.08	0	1769.64	Feasible	6688.7	0	1799.11
3_300_4e	Feasible	5834.08	0	1764.85	Feasible	8471.32	0	1799.42
3_300_5a	Feasible	9014.91	0	1799.04	Feasible	15542.9	0	1798.65
3_300_5b	Feasible	8270.16	0	1799.19	Feasible	20697.6	0	1798.73
3_300_5c	Feasible	9919.95	0	1797.16	Feasible	16959.8	0	1798.64
3_300_5d	Feasible	10297.6	0	1799.32	Feasible	16461.2	0	1799.04
3_300_5e	Feasible	9100.96	0	1797.07	Feasible	16141.3	0	1798.95
3_300_10a	Feasible	37028.6	0	1799.61	Feasible	49298.5	0	1799.32
3_300_10b	Feasible	33925.4	0	1799.74	Feasible	58197.1	0	1799.57
3_300_10c	Feasible	43682.6	0	1799.68	Feasible	52429.6	0	1799.65
3_300_10d	Feasible	35065.5	0	1799.67	Feasible	44749.9	0	1799.25
3_300_10e	Feasible	37841.1	0	1799.47	Feasible	51797.3	0	1799.8
3_300_15a	Feasible	61105.6	0	1799.75	Feasible	79954	0	1798.38
3_300_15b	Feasible	58358.4	0	1799.61	Feasible	87728.4	0	1799.7
3_300_15c	Feasible	64863.9	0	1799.77	Feasible	79070.6	0	1799.25
3_300_15d	Feasible	67186.6	0	1799.71	Feasible	67247.7	0	1799.49
3_300_15e	Feasible	53469.3	0	1799.64	Feasible	73494.7	0	1799.91
3_300_20a	Feasible	89533.1	0	1799.83	Feasible	113213	0	1799.03
3_300_20b	Feasible	96176.2	0	1799.82	Feasible	97957.2	0	1799.36
3_300_20c	Feasible	83349.4	0	1799.8	Feasible	95144.3	0	1799.59
3_300_20d	Feasible	79232	0	1799.28	Feasible	113188	0	1799.56
3_300_20e	Feasible	88667.9	0	1799.43	Feasible	106096	0	1799.55

A.11 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 400$

Tabela A.11: Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 50$

	$k=3$				$k=4$			
Instance	Status	UB	LB	Time(s)	Status	UB	LB	Time(s)
3_400_2a	Feasible	93.905	0	1799.23	Feasible	601.744	0	1798.69
3_400_2b	Feasible	100.099	0	1799.21	Feasible	690.419	0	1799.01
3_400_2c	Feasible	158.381	0	1799.21	Feasible	254.95	0	1799.11
3_400_2d	Feasible	182.881	0	1799.39	Feasible	476.441	0	1799.11
3_400_2e	Feasible	327.772	0	1799.19	Feasible	666.861	0	1799.08
3_400_3a	Feasible	2062.86	0	1798.86	Feasible	3472.03	0	1798.38
3_400_3b	Feasible	1747.87	0	1799.29	Feasible	4373.07	0	1798.71
3_400_3c	Feasible	1755.22	0	1799.33	Feasible	2495.82	0	1799.15
3_400_3d	Feasible	1153.8	0	1799.14	Feasible	2883.38	0	1799.18
3_400_3e	Feasible	1545.31	0	1799.12	Feasible	3577.91	0	1799.03
3_400_4a	Feasible	6917.87	0	1799.11	Feasible	8596.1	0	1798.42
3_400_4b	Feasible	5651.3	0	1799.2	Feasible	10012.5	0	1798.45
3_400_4c	Feasible	3366	0	1799.19	Feasible	7912.11	0	1798.93
3_400_4d	Feasible	5661.21	0	1799.07	Feasible	9518.91	0	1799.06
3_400_4e	Feasible	5973.58	0	1799.02	Feasible	10377.7	0	1799.08
3_400_5a	Feasible	9103.47	0	1799.13	Feasible	14301.2	0	1797.72
3_400_5b	Feasible	8814.49	0	1799.11	Feasible	15513.2	0	1798.98
3_400_5c	Feasible	9366.14	0	1799.16	Feasible	16039.5	0	1798.87
3_400_5d	Feasible	9344.93	0	1798.64	Feasible	14145.1	0	1798.72
3_400_5e	Feasible	8186.98	0	1799.06	Feasible	15633.4	0	1798.83
3_400_10a	Feasible	32611.9	0	1799.63	Feasible	53364	0	1798.87
3_400_10b	Feasible	33280.4	0	1799.54	Feasible	51815.4	0	1799.15
3_400_10c	Feasible	37574	0	1799.51	Feasible	43721.2	0	1799.78
3_400_10d	Feasible	35355.6	0	1799.28	Feasible	44755.5	0	1799.77
3_400_10e	Feasible	38396.5	0	1799.37	Feasible	54871.1	0	1799.13
3_400_15a	Feasible	65681.9	0	1799.8	Feasible	77390.6	0	1798.72
3_400_15b	Feasible	62965.4	0	1799.42	Feasible	86242.4	0	1799.96
3_400_15c	Feasible	58118.1	0	1799.79	Feasible	82750.8	0	1799.32
3_400_15d	Feasible	68745.8	0	1799.64	Feasible	79647.6	0	1799.64
3_400_15e	Feasible	63347.5	0	1799.62	Feasible	84181.6	0	1800.01
3_400_20a	Feasible	84340.9	0	1799.92	Feasible	111878	0	1798.73
3_400_20b	Feasible	85859.4	0	1799.52	Feasible	99060.8	0	1799.79
3_400_20c	Feasible	87625.7	0	1799.73	Feasible	112145	0	1800
3_400_20d	Feasible	72477.2	0	1799.74	Feasible	97210.7	0	1800.05
3_400_20e	Feasible	91552.3	0	1799.39	Feasible	106559	0	1799.77

A.12 Modelos de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 500$

Tabela A.12: Resultados usando instâncias de Kojić (2010) e (MOD_k) para $k \in \{3,4\}$ e $n = 500$

	$k=3$				$k=4$			
Instance	Status	UB	LB	Time(s)	Status	UB	LB	Time(s)
3_500_2a	Feasible	116.497	0	1799.14	Feasible	425.876	0	1797.96
3_500_2b	Feasible	160.295	0	1799.19	Feasible	329.433	0	1799.21
3_500_2c	Feasible	224.893	0	1799.18	Feasible	367.294	0	1799.02
3_500_2d	Feasible	61.047	0	1798.9	Feasible	393.62	0	1799.05
3_500_2e	Feasible	39.714	0	1799.06	Feasible	440.448	0	1799.02
3_500_3a	Feasible	1429.18	0	1799.28	Feasible	4278.56	0	1797.39
3_500_3b	Feasible	1688.88	0	1799.23	Feasible	3192.41	0	1799.11
3_500_3c	Feasible	1020.23	0	1799.09	Feasible	3115.03	0	1799.01
3_500_3d	Feasible	1233	0	1798.93	Feasible	4075.95	0	1799.15
3_500_3e	Feasible	706.12	0	1798.84	Feasible	4192.18	0	1799.07
3_500_4a	Feasible	5928.38	0	1799.18	Feasible	10749.4	0	1797.29
3_500_4b	Feasible	2930.39	0	1799.15	Feasible	8320.57	0	1799.1
3_500_4c	Feasible	3650.3	0	1799.16	Feasible	10950	0	1799.06
3_500_4d	Feasible	3266.17	0	1798.89	Feasible	12999.8	0	1798.89
3_500_4e	Feasible	4125.47	0	1798.91	Feasible	8850.88	0	1799
3_500_5a	Feasible	5566.55	0	1799.06	Feasible	12879.6	0	1797.09
3_500_5b	Feasible	7005.89	0	1799.07	Feasible	13007.9	0	1799.04
3_500_5c	Feasible	8932.97	0	1799	Feasible	17142.6	0	1799.15
3_500_5d	Feasible	7963.28	0	1798.95	Feasible	18104.2	0	1799
3_500_5e	Feasible	11284	0	1798.87	Feasible	13586.2	0	1798.84
3_500_10a	Feasible	44880.2	0	1799.54	Feasible	44425.6	0	1798.84
3_500_10b	Feasible	38800.9	0	1799.66	Feasible	45546.1	0	1799.98
3_500_10c	Feasible	36549.5	0	1798.98	Feasible	43597.3	0	1799.9
3_500_10d	Feasible	32786.4	0	1799.41	Feasible	50462.5	0	1799.15
3_500_10e	Feasible	42432.6	0	1799.32	Feasible	51362.8	0	1799.94
3_500_15a	Feasible	70379.1	0	1799.88	Feasible	72442.5	0	1799.35
3_500_15b	Feasible	65216.2	0	1799.81	Feasible	77359.9	0	1800.06
3_500_15c	Feasible	66522.9	0	1799.81	Feasible	89428.7	0	1800.11
3_500_15d	Feasible	67032.5	0	1799.6	Feasible	81164.6	0	1799.46
3_500_15e	Feasible	59802.2	0	1799.81	Feasible	81246.5	0	1800.08
3_500_20a	Feasible	76546.9	0	1799.95	Feasible	109933	0	1799.14
3_500_20b	Feasible	77116.9	0	1799.49	Feasible	96878.6	0	1799.77
3_500_20c	Feasible	83956	0	1799.96	Feasible	111243	0	1800
3_500_20d	Feasible	83892.2	0	1799.65	Feasible	109301	0	1800
3_500_20e	Feasible	89721.6	0	1800.02	Feasible	111790	0	1800.15

Apêndice B

Modelos Extras

B.1 Dedução das Restrições 3

Proposição B.1.1. *Seja* $(a_i)_{i\in I}$ *uma sequência de números reais.*

$$\max_{i\in I} a_i - \min_{i\in I} a_i = \frac{1}{2}\left(|a_1 - a_n| + \sum_{i=1}^{n-1} |a_i - a_{i+1}|\right) \tag{B.1}$$

Prova: A fórmula é invariante em relação a toda permutação $a_{\sigma(i)}$. Para provar isso, basta notar que a troca de ordem de dois elementos da sequência não altera o resultado da fórmula. Como toda permutação é um produto de transposições, então o resultado valerá para toda permutação. Cada índice $i \in I$ aparece duas vezes na soma B.1. Para todo índice i, tem-se $...|a_{i-1} - \underbrace{a_i}| + |\underbrace{a_i} - a_{i+1}|....$ Os índices 1 e n aparecem pela segunda vez no módulo $|a_1 - a_n|$. Como a ordem dos elementos não altera o resultado da expressão B.1 do lado direito, fixa-se a ordem decrescente $a_i \geq a_{i+1},\ \forall i \in I_{n-1}$. Pode-se remover o módulo dos termos da soma e o resultado é

$$\frac{1}{2}\left(a_1 - a_n + \sum_{i=1}^{n-1} a_i - a_{i+1}\right) = \frac{1}{2}\left((a_1 - a_n) + (a_1 - a_n)\right) = a_1 - a_n.$$

ou seja, o maior elemento da sequência, a_1, menos o menor elemento, a_n. ■

A Proposição B.1.1 permite um novo modelo matemático (B.2) por meio de novas restrições para limitar o diâmetro da solução.

$$\min \quad T \tag{B.2a}$$

$$\text{suj. a} \quad \sum_{j=1}^{k} x_{ji} = 1, \quad \forall i \in I_n \tag{B.2b}$$

$$\sum_{i=1}^{n} x_{ji} \geq 1, \quad \forall j \in I_k \tag{B.2c}$$

$$-y_{lj} \leq \sum_{i=1}^{n} v_{il}(x_{j,i} - x_{j+1,i}) \leq y_{lj}, \quad \forall (l,j) \in I_m \times I_{k-1} \tag{B.2d}$$

$$-y_{lk} \leq \sum_{i=1}^{n} v_{il}(x_{1,i} - x_{k,i}) \leq y_{lk}, \quad \forall l \in I_m \tag{B.2e}$$

$$\sum_{j=1}^{k} y_{lj} \leq 2.T, \quad \forall l \in I_m \tag{B.2f}$$

$$y_{lj} \in \mathbb{R}_+, \quad \forall (l,j) \in I_m \times I_k \tag{B.2g}$$

$$x_{ji} \in \{0,1\}, \quad \forall (j,i) \in I_k \times I_n \tag{B.2h}$$

$$T \in \mathbb{R}_+ \tag{B.2i}$$

As Variáveis (B.2g) são os termos em módulo do somatório em B.1. As Restrições (B.2d) representam o módulo da diferença das partes. Por fim, (B.2f) é o somatório que representa a equivalência provada na Proposição B.1.1.

B.2 Dedução das Restrições 4

O quarto modelo matemático para o MDMWNPP é baseado na mudança do significado das variáveis. Seja V a sequência de vetores m-dimensionais. Os índices i e j agora só indexam elementos da sequência a ser k-particionada. A variável $y_{i,j} = 1$, se i e j estão juntos numa mesma parte ou $y_{ij} = 0$, caso contrário. Não é necessário indexar a parte da partição neste modelo, pois esta será identificada pelo elemento líder da parte, ou seja, aquele com o menor índice dentre todos os elementos no mesmo subconjunto. A variável binária $z_i = 1$, se i é índice de um elemento líder; ou $z_i = 0$, caso contrário. Pode-se escrever a nova variável binária em termos de $x_{j,i}$ do modelo 4.23 da seguinte maneira:

$$y_{i,j} = \sum_{h=1}^{k} x_{h,i}.x_{h,j} \tag{B.3}$$

Para reconstruir as restrições apropriadas ao MDMWNPP com esta nova variável, deve-se considerar que as novas variáveis y_{ij} têm uma propriedade conhecida como relação de transitividade. Quando $y_{ij} = 1$ e $y_{jh} = 1$, obrigatoriamente tem-se $y_{ih} = 1$. Para encontrar as restrições que garantem a transitividade do conjunto de pares (i,j), deve-se partir de uma expressão lógica e usar algumas ideias da redução do SAT ao Programa de programação binária.

$$y_{ij} \wedge y_{jk} \implies y_{ik}, \quad \forall (i,j,k) : 1 \leq i < j < k \leq n \tag{B.4}$$

• Equivalência da condicional:

$$\neg(y_{ij} \wedge y_{jk}) \vee y_{ik}, \quad \forall (i,j,k) : 1 \leq i < j < k \leq n \tag{B.5}$$

• De Morgan:

$$\neg y_{ij} \vee \neg y_{jk} \vee y_{ik}, \quad \forall (i,j,k) : 1 \leq i < j < k \leq n \tag{B.6}$$

Partindo dessa forma disjuntiva, derivam-se as inequações que garantem a relação de transitividade:

$$(1 - y_{ij}) + (1 - y_{jk}) + y_{ik} \geq 1, \quad \forall (i,j,k) : 1 \leq i < j < k \leq n \tag{B.7}$$

$$y_{ij} + y_{jk} - y_{ik} \leq 1, \quad \forall (i,j,k) : 1 \leq i < j < k \leq n \tag{B.8}$$

A variável z_i, que define se o elemento indexado por i é, ou não, líder de uma parte. Por definição $z_1 = 1$, sem perda de generalidade. Outra propriedade importante a ser garantida pelas restrições é que devem existir pelo menos um conjunto de k elementos que não estão juntos dois a dois. A ideia é impor uma restrição que garanta $z_{ij} = 0$ sempre que i ou j forem líderes de uma parte. Isso formará uma grande "clique"de zeros.

$$z_i \wedge z_j \implies \neg y_{ij}, \quad \forall (i,j) : 1 \leq i < j \leq n \tag{B.9}$$

$$\neg(z_i \wedge z_j) \vee \neg y_{ij}, \quad \forall (i,j) : 1 \leq i < j \leq n \tag{B.10}$$

$$\neg z_i \vee \neg z_j \vee \neg y_{ij}, \quad \forall (i,j) : 1 \leq i < j \leq n \tag{B.11}$$

$$(1 - z_i) + (1 - z_j) + (1 - y_{ij}) \geq 1, \quad \forall (i,j) : 1 \leq i < j \leq n \tag{B.12}$$

$$z_i + z_j + y_{ij} \leq 2, \quad \forall (i,j) : 1 \leq i < j \leq n \tag{B.13}$$

O modelo matemático B.14 usa uma variável representado pelo produto interno de dois vetores de variáveis binárias dos modelos B.2, mesmo assim, com a devida interpretação das propriedades de sua solução, é um MILP.

$$\begin{aligned}
\min \quad & T && \text{(B.14a)}\\
\text{suj. a} \quad & y_{ij} + y_{jh} - y_{ih} \leq 1, \ \forall (i,j,h) : 1 \leq i < j < h \leq n && \text{(B.14b)}\\
& z_i + \sum_{h=1}^{i-1} y_{hi} + \sum_{l=i+1}^{n} y_{ih} \geq 1, \ \forall i \in I_{n-1} \setminus \{1\} && \text{(B.14c)}\\
& z_i + z_j + y_{ij} \leq 2, \quad \forall (i,j) : 1 \leq i < j \leq n && \text{(B.14d)}\\
& \sum_{i \in I_n} z_i = k && \text{(B.14e)}\\
& t_{1l} \leq v_{il} z_i + \sum_{h=1}^{i-1} v_{hl} y_{hi} + \sum_{h=i+1}^{n} v_{hl} y_{ih} \leq t_{2l}, \quad \forall (i,l) \in I_n \times I_m && \text{(B.14f)}\\
& t_{1l} - t_{2l} \leq T \quad \forall l \in I_m && \text{(B.14g)}\\
& z_i \in \{0,1\}, \quad \forall i \in I_n && \text{(B.14h)}\\
& t_{1l}, t_{2l}, T \in \mathbb{R}_+, \quad \forall l \in I_m && \text{(B.14i)}\\
& y_{ij} \in \{0,1\}, \quad \forall (j,i) : 1 \leq i < j \leq n && \text{(B.14j)}
\end{aligned}$$

A função objetivo é dada pela Expressão (B.14a) e representa o diâmetro do conjunto $\{g(A_j)\}_{j \in I_k}$. O objetivo é minimizar T. As restrições (B.14b) estabelecem a relação de transitividade entre os pares (i,j). As restrições (B.14c) definem que cada elemento i deve estar com algum outro ou ser o líder isolado de uma parte. As restrições (B.14d) definem que dois líderes nunca estão juntos na mesma parte. A restrição (B.14e) fixa exatamente k elementos líderes. As restrições (B.14f) e (B.14g) são auxiliares para o cálculo do diâmetro de $\{g(A_j)\}_{j \in I_k}$. As variáveis (B.14h) significam $z_i = 1$, se o vetor de índice i é elemento líder de uma parte, e $z_i = 0$, caso contrário. As variáveis (B.14i) são limitantes de somas das partes e diâmetro de $\{g(A_j)\}_{j \in I_k}$. A relação (B.14j) indica que $y_{ji} = 1$, se o vetor de índice i e o vetor de índice j pertencem à parte, e $y_{ji} = 0$, caso contrário.