



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE COMPUTAÇÃO
PROGRAMA DE MESTRADO E DOUTORADO EM CIÊNCIAS DA
COMPUTAÇÃO

JOEL CRUZ SOARES

O PROBLEMA DA ATRIBUIÇÃO CONEXA

FORTALEZA

2016

JOEL CRUZ SOARES

O PROBLEMA DA ATRIBUIÇÃO CONEXA

Dissertação de Mestrado apresentada ao Programa de Mestrado e Doutorado em Ciências da Computação, do Departamento de Computação da Universidade Federal do Ceará, como requisito parcial para obtenção do Título de Mestre em Ciências da Computação. Área de concentração: Algoritmos e Otimização.

FORTALEZA

2016

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S654p Soares, Joel Cruz.

O Problema da Atribuição Conexa / Joel Cruz Soares. – 2016.
99 f. : il. color.

Dissertação (mestrado) – Universidade Federal do Ceará, Centro de Ciências, Programa de Pós-Graduação em Ciência da Computação, Fortaleza, 2016.

Orientação: Prof. Dr. Manoel Bezerra Campêlo Neto.

1. Rede de Comunicações Móveis. 2. Complexidade Computacional. 3. Programação Inteira. 4. Combinatória Poliédrica. 5. Heurística.. I. Título.

CDD 005

Joel Cruz Soares

O Problema da Atribuição Conexa

Dissertação submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação, da Universidade Federal do Ceará, como parte dos requisitos exigidos para obtenção do grau de Mestre em Ciência da Computação.

Aprovada em

BANCA EXAMINADORA

Prof. Dr. Manoel Bezerra Campêlo Neto(Orientador)
MDCC/Universidade Federal do Ceará

Prof. Dr. Alexandre Salles da Cunha
Universidade Federal de Minas Gerais

Prof. Dr. Victor Almeida Campos
MDCC/Universidade Federal do Ceará

Prof. Dr. Tarcisio Ferreira Maciel
Universidade Federal do Ceará

Fortaleza/CE

2016

Resumo

Apresentamos um problema com aplicação em alocação de recursos em redes de comunicações móveis, que denominamos de Problema da Atribuição Conexa em Vetores (ACV). Este problema tem como entrada um conjunto de símbolos $I = \{1, 2, \dots, M\}$, um vetor v indexado por $J = \{1, 2, \dots, N\}$, e um valor de ganho ρ_{ij} ao alocar $i \in I$ à posição j de v . Desejamos encontrar uma atribuição dos símbolos ao vetor que tenha o maior ganho possível, sob a restrição de que símbolos repetidos sejam adjacentes no vetor. Demonstramos que ACV é um problema NP-Difícil a partir de uma redução do Problema de Recoloração Convexa de Caminhos (RCC). Apresentamos um algoritmo aproximativo para um caso particular deste problema (k -ACV). Propomos três formulações de Programação Inteira e comparamos teoricamente suas relaxações lineares. Estudamos o poliedro \mathcal{P} associado à formulação mais forte. Determinamos todas as desigualdades indutoras de facetas com lado direito igual a 1 e mostramos que elas, junto com as restrições de não-negatividade, descrevem \mathcal{P} quando $M = 2$ ou $N = 2$. Generalizamos essa classe de desigualdades válidas, mantendo a propriedade de que induzem facetas. Ao final, propomos 5 heurísticas para o problema e as comparamos através de resultados de experimentos computacionais.

Palavras-chaves: Rede de Comunicações Móveis. Complexidade Computacional. Programação Inteira. Combinatória Poliédrica. Heurística.

Abstract

We present a problem with application in resource allocation in mobile networks, that we name Connected Assignment in Arrays (CAA). This problem has as input a set of symbols $I = \{1, 2, \dots, M\}$, an array v indexed by $J = \{1, 2, \dots, N\}$, and a gain value ρ_{ij} of allocating $i \in I$ to position j of v . We want to find an assignment of symbols to positions so as to maximize the gain, under the constraint that repeated symbols are adjacent in the array. We demonstrate that CAA is an NP-Hard problem by a reduction from the Convex Path Recoloring Problem (CPR). We present an approximate algorithm for a particular case of this problem (k -CAA). We propose three ILP formulations and theoretically compare their linear relaxation. We study the polyhedron \mathcal{P} associated with the tightest formulation. We determine all facet-defining inequalities with right-hand side equal to 1 and show that they suffice, together with the non-negativeness constraints, to describe \mathcal{P} when $M = 2$ or $N = 2$. We generalize this class of valid inequalities while keeping the property of being facet inducing. Finally, we propose 5 heuristics for the problem and compare them by results of computational experiments.

Key-words: Mobile communications network. Computational complexity. Integer Programming. Polyhedral Combinatorics. Heuristic.

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Trabalhos Relacionados	3
1.3	Organização do Trabalho	4
2	Conceitos Preliminares	5
2.1	Teoria dos Grafos	5
2.2	Problemas de Otimização	9
2.2.1	PPLs e PLIs	9
2.2.2	Relaxação Lagrangeana	10
2.2.3	Algoritmos Aproximativos e Heurísticas	12
2.3	Poliedros	13
2.3.1	Faces	13
2.3.2	Dimensão de um Conjunto e Facetas	14
2.4	O Problema da Recoloração Convexa - RCC	15
3	O Problema ACV	17
3.1	Definições	17
3.2	Complexidade	18
3.2.1	Redução: RCC-D \rightarrow ACV-D	18
3.2.2	NP-Completeness	19
3.3	Algoritmo Aproximativo Para o Problema k -ACV	20
4	Formulações Matemáticas	24
4.1	Formulação Z	24
4.2	Formulação X	25
4.3	Formulação XY	26

4.4	Relacionamento Entre as Formulações	27
4.4.1	Relação Entre as Formulações Z e X	27
4.4.2	Relação Entre as Formulações XY e X	28
5	Estudo Poliédrico do Problema ACV	30
5.1	O Poliedro \mathcal{P} Associado ao Problema ACV	30
5.2	Todas as Desigualdades Fortes com Coeficientes Binários	32
5.3	Uma Generalização da Faceta de Coeficientes Binários	37
5.4	Descrição Completa de \mathcal{P} Para $M = 2$ ou $N = 2$	43
6	Heurísticas	46
6.1	A Heurística de Referência - H_0	46
6.2	Uma Heurística Baseada em Multa - H_1	49
6.3	A Heurística H_2	52
6.4	A Heurística H_3	54
6.5	Uma Heurística Gulosa - H_4	54
6.6	Uma Heurística Baseada em Relaxação Lagrangeana - H_5	56
7	Resultados Computacionais	60
7.1	Instâncias	60
7.2	Apresentação e Avaliação de Resultados	61
7.3	Resultados Para $N = 20$	61
7.4	Resultados Para $N = 60$	65
7.5	Resultados Para $N = 100$	69
7.6	Comportamento das Heurísticas em Função de N	71
7.7	Conclusões	74
8	Considerações Finais e Trabalhos Futuros	76
8.1	Considerações Finais	76
8.2	Trabalhos Futuros	77
A	Resultados de Experimentos Computacionais	79
	Referências Bibliográficas	83

Lista de Figuras

1.1	Matriz de ganho que representa uma instância do Problema ACV.	3
1.2	Exemplos de atribuição para a instância da Figura 1.1.	3
2.1	Exemplo de Grafo Simples.	5
2.2	Exemplo de Grafo Desconexo.	6
2.3	Exemplo de Grafo Bipartido.	7
2.4	Exemplo de Subgrafo Induzido. À Esquerda Temos o Grafo G e à Direita o Subgrafo de G Induzido Por T , onde $T = \{a, c, d\}$	7
2.5	Exemplo de Grafo de Intervalo e Sua Representação em Intervalos.	8
2.6	Exemplo de Instância do Problema RCC (Grafo Superior) e Sua Recoloração Convexa Ótima (Grafo Inferior)	15
3.1	Função ganho ρ da família de instâncias tratada no Teorema 3.14.	22
3.2	Atribuição inicial A do Algoritmo 1.	23
3.3	Na parte superior, atribuição \hat{A} após o laço. Na parte inferior, atribuição \hat{A} finalizada.	23
5.1	Grafo de Conflitos Para o Caso Onde $M = N = 2$	43
6.1	Ilustração das operações relacionadas à alocação virtual.	47
6.2	Regra R_1 para geração de Alocações Virtuais.	48
6.3	Regra R_2 para geração de Alocações Virtuais.	48
6.4	Regra R_3 para geração de Alocações Virtuais.	48
6.5	Ilustração da Ideia Básica Para a Definição da Função de Multa.	49
7.1	Gráfico da Razão de Aproximação para $N = 20$ (Instâncias Simuladas à Esquerda, Instâncias Aleatórias à Direita).	62
7.2	Gráficos do Tempo de Computação para $N = 20$ (Instâncias Simuladas).	62

LISTA DE FIGURAS

7.3	Gráficos do Tempo de Computação para $N = 20$ (Instâncias Aleatórias). . . .	63
7.4	Gráfico da Razão de Aproximação para $N = 60$ (Instâncias Simuladas à Esquerda, Instâncias Aleatórias à Direita).	66
7.5	Gráficos do Tempo de Computação para $N = 60$ (Instâncias Simuladas). . . .	66
7.6	Gráficos do Tempo de Computação para $N = 60$ (Instâncias Aleatórias). . . .	67
7.7	Gráficos da Razão de Aproximação para $N = 100$ (Instâncias Simuladas à Esquerda, Instâncias Aleatórias à Direita).	69
7.8	Gráficos do Tempo de Computação para $N = 100$ (Instâncias Simuladas). . .	70
7.9	Gráficos do Tempo de Computação para $N = 100$ (Instâncias Aleatórias). . .	70
7.10	Gráfico da Média Razão de Aproximação.	73
7.11	Gráficos de Média do Tempo de Computação (Instâncias Simuladas).	73
7.12	Gráficos de Média do Tempo de Computação (Instâncias Aleatórias).	74

Lista de Tabelas

7.1	Intervalos de Confiança Para a Razão de Aproximação (Instâncias Simuladas com $N = 20$).	64
7.2	Intervalos de Confiança Para a Razão de Aproximação (Instâncias Aleatórias com $N = 20$).	64
7.3	Resultados Médios Gerais Para $N = 20$ (Instâncias Simuladas).	64
7.4	Resultados Médios Gerais Para $N = 20$ (Instâncias Aleatórias).	65
7.5	Intervalos de Confiança Para a Razão de Aproximação (Instâncias Simuladas com $N = 60$).	67
7.6	Intervalos de Confiança Para a Razão de Aproximação (Instâncias Aleatórias com $N = 60$).	68
7.7	Resultados Médios Gerais para $N = 60$ (Instâncias Simuladas).	68
7.8	Resultados Médios Gerais para $N = 60$ (Instâncias Aleatórias).	68
7.9	Intervalos de Confiança Para a Razão de Aproximação (Instâncias Simuladas com $N = 100$).	71
7.10	Intervalos de Confiança Para a Razão de Aproximação (Instâncias Aleatórias com $N = 100$).	71
7.11	Resultados Médios Gerais para $N = 100$ (Instâncias Simuladas).	72
7.12	Resultados Médios Gerais para $N = 100$ (Instâncias Aleatórias).	72
A.1	Médias da Razão de Aproximação Para $N = 20$ (Instâncias Reais).	79
A.2	Médias da Razão de Aproximação Para $N = 20$ (Instâncias Artificiais).	79
A.3	Médias da Razão de Aproximação Para $N = 60$ (Instâncias Reais).	80
A.4	Médias da Razão de Aproximação Para $N = 60$ (Instâncias Artificiais).	80
A.5	Médias da Razão de Aproximação Para $N = 100$ (Instâncias Reais).	80
A.6	Médias da Razão de Aproximação Para $N = 100$ (Instâncias Artificiais).	80
A.7	Médias do Tempo de Computação Para $N = 20$ (Instâncias Reais).	81

LISTA DE TABELAS

A.8 Médias do Tempo de Computação Para $N = 20$ (Instâncias Artificiais).	81
A.9 Médias do Tempo de Computação Para $N = 60$ (Instâncias Reais).	81
A.10 Médias do Tempo de Computação Para $N = 60$ (Instâncias Artificiais).	81
A.11 Médias do Tempo de Computação Para $N = 100$ (Instâncias Reais).	82
A.12 Médias do Tempo de Computação Para $N = 100$ (Instâncias Artificiais).	82

Lista de Abreviaturas e Siglas

ACM	Atribuição Conexa em Matrizes, 78
ACV	Atribuição Conexa em Vetores, 1
CNMAI	Congresso Nacional de Matemática Aplicada à Indústria, 1
PLI	Programação Linear Inteira, 10
PLM	Programação Linear Misto, 10
PPL	Problema de Programação Linear, 9
RCC	Recoloração Convexa de Caminhos, 15

Lista de Símbolos

\mathcal{P}	Poliedro Associado ao Problema ACV
ρ	Função Ganho
G	Grafo de Conflitos Associado ao Poliedro \mathcal{P}
I	Conjunto de Símbolos
J	Conjunto de Índices do Vetor
$J(p)$	Conjunto de Posições Ativas no Padrão p
M	Número de Símbolos
N	Número de Posições do Vetor
P	Número de Padrões Possíveis Relativo a N Posições do Vetor / Último Padrão de Atribuição / Índice do Padrão que Aloca Todas as Posições do Vetor
$P_{\cap}(p)$	Conjunto de Padrões que Interceptam p
$P_{\supset}(p)$	Conjunto de Padrões que Contém p
Q	Conjunto de Índices de Padrões
T	Matriz de Padrões de Atribuição

Capítulo 1

Introdução

Neste capítulo, introduzimos o Problema da Atribuição Conexa, ACV, de uma maneira pouco formal. Apresentamos as motivações do problema, definimos propriedades básicas e mostramos os elementos que compõem uma instância de ACV. Além disso, apresentamos alguns trabalhos relacionados ao problema e definimos a organização do trabalho na última seção deste capítulo.

Alguns resultados apresentados neste trabalho foram publicados nos anais do Congresso Nacional de Matemática Aplicada à Indústria (CNMAI 2015) [1]. O artigo apresentado no CNMAI tratou de demonstrar que ACV é NP-Difícil, resultado no Capítulo 3, e apresentar resultados de testes computacionais com três heurísticas propostas, identificadas no Capítulo 6 como H_1 , H_3 e H_4 , duas delas baseadas na formulação proposta em [2]. Nesta dissertação, expandimos tais resultados propondo duas outras formulações, duas novas heurísticas ampliando os experimentos numéricos e realizando um estudo do poliedro associado ao problema.

1.1 Motivação

Nas últimas décadas, os sistemas de comunicações celulares têm se tornado bastante populares, a ponto de serem atualmente de fundamental importância em uma sociedade moderna. Esta crescente popularização tem tornado este mercado a cada dia mais competitivo e, devido a esta tendência, vários serviços e novas tecnologias têm sido desenvolvidas. Para enfrentar o constante aumento de tráfego, muitas mudanças no núcleo da rede foram realizadas e novas técnicas foram desenvolvidas. Algumas destas mudanças tratam-se do uso de múltiplas antenas nos nós da rede e da utilização de algoritmos de alocação de recursos de rádio. Estes algoritmos devem gerenciar da forma mais eficiente possível tais recursos que, de modo geral, são escassos e podem ser, por exemplo, intervalos de tempo para transmissão

(*time slots*) ou faixas de frequência.

O Problema da Atribuição Conexa em Vetores, ACV, que definimos no presente trabalho, consiste de um problema de otimização combinatória com aplicação dentro do contexto de alocação de recursos de rádio. Uma das técnicas utilizadas nesse contexto considera blocos de frequência, ou faixas de frequência, de mesmo tamanho que são disponibilizados aos clientes. Por particularidades relacionadas ao projeto do sistema, se vários blocos são alocados a um mesmo cliente, então eles devem ser adjacentes. O intuito principal é alocar os blocos de frequência respeitando tal restrição de alocação de modo a maximizar a taxa de transmissão total.

Resumidamente, ACV tem como entrada um conjunto de símbolos $I = \{1, 2, \dots, M\}$ que representa os clientes, um vetor v indexado por $J = \{1, 2, \dots, N\}$ que indica os blocos de frequência, e uma função ganho ρ tal que $\rho(i, j)$ (ou simplesmente ρ_{ij}) define o ganho ao alocar $i \in I$ à posição j de v . Desejamos encontrar uma atribuição conexa que tenha o maior ganho possível. Informalmente, uma atribuição é conexa se um símbolo repetido define um único bloco (subintervalo) do vetor, composto exclusivamente por este símbolo. Iremos nos referir a esta restrição, como restrição de adjacência ou restrição de conexidade. No Capítulo 3, definimos formalmente o problema ACV e mostramos que é um problema NP-Difícil.

A Figura 1.1 mostra uma entrada do Problema ACV. A matriz mostrada representa a função ganho ρ da instância. Cada linha i define os ganhos do símbolo i nas posições de v , e cada entrada (i, j) especifica o ganho de i na posição j de v . Dessa maneira, podemos concluir que $I = \{1, 2, 3, 4\}$ e $J = \{1, 2, 3, 4, 5, 6\}$, isto é, consideramos a alocação de 4 símbolos em um vetor de 6 posições.

Observe as atribuições da Figura 1.2. A atribuição A_1 aloca o símbolo 1 nas 4 primeiras posições, o símbolo 3 na quinta e o símbolo 4 na sexta. O ganho desta atribuição é a soma dos ganhos dos símbolos nas respectivas posições em que foram alocados. Portanto, A_1 tem ganho correspondente à soma $1 + 2 + 1 + 2 + 2 + 4$, isto é, tal atribuição tem ganho igual a 12. Além disso, notamos que esta respeita a restrição de adjacência imposta pelo problema e, sendo assim, representa uma solução viável para a instância considerada. A atribuição A_2 é uma solução inviável, pois viola a restrição de adjacência, e tem ganho de valor 15; as atribuições A_3 e A_4 tem valor 13 e são soluções ótimas para a instância considerada.

$i \setminus j$	1	2	3	4	5	6
1	1	2	1	2	0	0
2	0	0	3	0	1	0
3	2	2	0	0	2	1
4	2	0	0	2	0	4

Figura 1.1: Matriz de ganho que representa uma instância do Problema ACV.

$$\begin{aligned}
 A_1 &\rightarrow \boxed{1 \mid 1 \mid 1 \mid 1 \mid 3 \mid 4} & A_2 &\rightarrow \boxed{3 \mid 3 \mid 2 \mid 1 \mid 3 \mid 4} \\
 A_3 &\rightarrow \boxed{3 \mid 3 \mid 2 \mid 4 \mid 4 \mid 4} & A_4 &\rightarrow \boxed{3 \mid 1 \mid 2 \mid 4 \mid 4 \mid 4}
 \end{aligned}$$

Figura 1.2: Exemplos de atribuição para a instância da Figura 1.1.

1.2 Trabalhos Relacionados

O problema matemático que denominamos ACV está sendo definido neste trabalho. No entanto, em [2], ACV é estudado diretamente dentro do contexto da aplicação, isto é, no campo da engenharia de telecomunicações. Naquele trabalho é desenvolvido um simulador que modela um ambiente celular moderno e pode ser utilizado para gerar instâncias “reais” do problema. Além disso, são apresentadas variações de ACV e heurísticas para resolvê-las, sendo também descritos modelos de programação inteira.

Em [3] é estudado o problema da Recoloração Convexa de Caminhos (RCC). Neste problema, é dado um caminho (grafo) com uma coloração arbitrária nos vértices. O objetivo é recolorir um número mínimo de vértices de modo a obter uma coloração convexa. Uma coloração é convexa se, para cada cor, o subgrafo induzido pelos vértices desta cor é conexo. O referido trabalho apresenta um algoritmo aproximativo, faz um estudo poliédrico do problema RCC e apresenta resultados de simulações computacionais.

Observe a semelhança entre o problema ACV e o problema RCC. Relacionando as posições do vetor aos vértices do caminho, vemos que são equivalentes as restrições de conectividade na atribuição e de convexidade da coloração.

1.3 Organização do Trabalho

Antes de apresentarmos propriamente nosso estudo do problema, resumimos no Capítulo 2 os principais conceitos teóricos de Teoria dos Grafos, Problemas de Otimização e Poliedros que serão utilizados neste trabalho.

No Capítulo 3, buscamos apresentar de maneira mais formal as definições e convenções utilizadas. Mostramos que o Problema ACV é NP-Difícil, mesmo que a matriz de ganho seja uma matriz binária, com no máximo duas entradas não nulas por linha e uma única entrada não nula por coluna. Ao final desse capítulo, apresentamos um algoritmo aproximativo para o caso onde a matriz de ganho tem no máximo k entradas não nulas por linha, que denominamos k -ACV.

No Capítulo 4, propomos três formulações de programação inteira para o problema ACV: Formulação Z; Formulação X; e Formulação XY. Ao final deste capítulo, mostramos que a Formulação X pode ser obtida a partir da aplicação da Decomposição de Dantzig-Wolfe à Formulação Z e que um fortalecimento da Formulação XY a torna equivalente à Formulação X.

No Capítulo 5, definimos o poliedro \mathcal{P} associado à Formulação X e estudamos suas propriedades. Iniciamos por demonstrar que \mathcal{P} tem dimensão plena; posteriormente introduzimos uma classe de desigualdades válidas com lado direito igual a 1 e mostramos que esta define todas as facetas com coeficientes binários. Generalizamos esta desigualdade obtendo uma classe de desigualdades mais ampla indutora de facetas. Ao final, caracterizamos a envoltória convexa para os casos onde $M = 2$ ou $N = 2$.

No Capítulo 6, propomos 5 heurísticas para o Problema ACV e apresentamos uma heurística encontrada na literatura, H_0 , que será utilizada como referência de comparação nos testes computacionais. Algumas dessas heurísticas são baseadas nas formulações apresentadas no Capítulo 4.

No Capítulo 7, mostramos os resultados de experimentos computacionais que foram realizados com as heurísticas propostas e os analisamos fazendo comparação com os resultados obtidos para a heurística encontrada na literatura, H_0 .

O Capítulo 8 trata das considerações finais deste trabalho e projeta as possibilidades de estudos futuros relacionados ao problema ACV.

Capítulo 2

Conceitos Preliminares

Neste capítulo, introduzimos os principais conceitos abordados neste trabalho. Buscamos mostrar propriedades, teoremas e resultados gerais relacionados a Teoria dos Grafos, Problemas de Otimização e Poliedros. As demonstrações de teoremas e propriedades podem ser consultadas em: [4] e [5], para Teoria dos Grafos; [6], [7] e [8], para Problemas de Otimização; [9] e [6], para Poliedros.

2.1 Teoria dos Grafos

Um **Grafo** G é uma dupla que consiste de um conjunto de vértices $V(G)$ e um conjunto de arestas $E(G)$, onde cada aresta é definida por dois vértices (não necessariamente distintos), chamados de extremos. Um **laço** é uma aresta cujos extremos são iguais. **Arestas múltiplas** são arestas que têm os mesmos extremos. Dizemos que um grafo é simples se não tem laços nem aresta múltipla. Na Figura 2.1, temos um grafo G simples, onde $V(G) = \{a, b, c, d\}$ e $E(G) = \{ab, ad, bc, bd\}$.

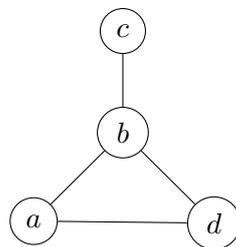


Figura 2.1: Exemplo de Grafo Simples.

Dois vértices, u e v , são **vizinhos** (ou adjacentes) se são extremos de uma mesma aresta. Denotamos por $N(u)$ a **vizinhança** do vértice u , isto é, $N(u) = \{v \in V(G) : v \text{ é adjacente a } u\}$. Dizemos que um grafo é **completo** se quaisquer dois vértices são adja-

centes.

O **complementar** \bar{G} de um grafo simples G é um grafo simples com vértices $V(G)$ e arestas $E(\bar{G})$ tais que $uv \in E(\bar{G})$ se, e somente se, $uv \notin E(G)$, para $u, v \in V(G), u \neq v$. Uma **clique** de um grafo é um conjunto de vértices dois-a-dois adjacentes. A cardinalidade de uma clique máxima em um grafo G é chamada de **número de clique** e denotada por $\omega(G)$. Um **conjunto independente** (ou conjunto estável) em um grafo é um conjunto de vértices dois-a-dois não adjacentes. A cardinalidade de um conjunto independente máximo em um grafo é chamada de **número de independência** e denotada por $\alpha(G)$. No exemplo anterior, o conjunto $C = \{a, b, d\}$ é uma clique de tamanho 3 e o conjunto $I = \{a, c\}$ é um conjunto independente de tamanho 2.

Um **passeio** em um grafo é uma sequência finita de vértices (v_1, v_2, \dots, v_k) tal que v_i é adjacente a v_{i+1} para todo i onde $0 < i < k$. Se nenhum vértice aparece mais de uma vez na sequência, dizemos que o passeio é também um **caminho**. Se $v_1 = v_k$ e não há mais nenhuma repetição na sequência de vértices, então o passeio é um **ciclo**. O ciclo é dito par se envolve um número par de arestas, caso contrário, dizemos que o ciclo é ímpar.

Um grafo é **conexo** se, e somente se, existe um caminho (v_1, v_2, \dots, v_k) para todo par de vértices v_1 e v_k de $V(G)$. Caso o grafo G não seja conexo, chamamos G de grafo **desconexo**. O grafo da Figura 2.1 trata-se de um grafo conexo, enquanto que o grafo mostrado na Figura 2.2 é um grafo desconexo.

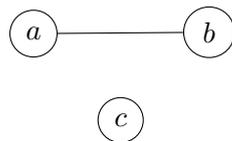


Figura 2.2: Exemplo de Grafo Desconexo.

Um grafo G é **bipartido** se $V(G)$ é a união de dois conjuntos independentes disjuntos, U' e U'' , chamados de partições de G . Se todo $u' \in U'$ é adjacente a todos os vértices de U'' , dizemos que G é um grafo bipartido completo. É possível mostrar que um grafo é bipartido se, e somente se, não possui ciclos ímpares (Teorema de König). Na Figura 2.3, temos um grafo bipartido com partições $U' = \{a, b, c, d\}$ e $U'' = \{e, f, g\}$.

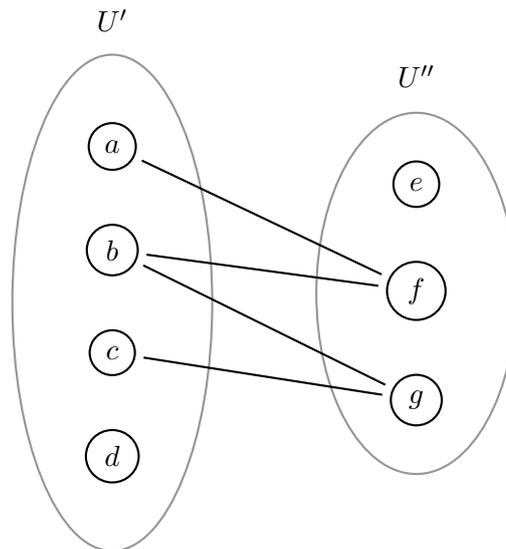


Figura 2.3: Exemplo de Grafo Bipartido.

Um **subgrafo** de um grafo G é um grafo H tal que $V(H) \subseteq V(G)$ e $E(H) \subseteq E(G)$. Então, escrevemos $H \subseteq G$ e dizemos que G contém H . Um **subgrafo induzido** por um conjunto de vértices $T \subseteq V(G)$ é um subgrafo cujo conjunto de vértices é T e cujas arestas são exatamente aquelas de G com ambos os extremos em T . Denotamos por $G[T]$ o subgrafo de G induzido por T . As figuras 2.4(a) e 2.4(b) ilustram estes conceitos. Considerando o grafo G da Figura 2.4(a) e sendo $T = \{a, c, d\}$, mostramos na Figura 2.4(b) o subgrafo de G induzido por T .



Figura 2.4: Exemplo de Subgrafo Induzido. À Esquerda Temos o Grafo G e à Direita o Subgrafo de G Induzido Por T , onde $T = \{a, c, d\}$.

Um **emparelhamento** em um grafo G é um conjunto de arestas (que não são laços) que não compartilham extremos. O problema do **emparelhamento de peso máximo** consiste em, dados um grafo G e uma função peso w , que define o peso w_e da aresta $e \in E(G)$, encontrar um emparelhamento M em G tal que $\sum_{e \in M} w_e$ seja o máximo possível. Para grafos

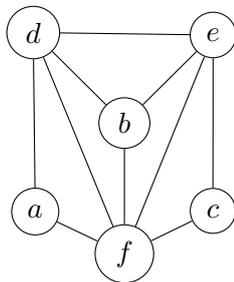
bipartidos, um algoritmo muito conhecido para este problema e também bastante utilizado na prática para resolvê-lo é o **Algoritmo Húngaro** [10] [11]. Algumas implementações deste algoritmo têm complexidade $O(N^3)$, onde N é o tamanho da maior partição do grafo bipartido.

Uma **k -coloração** (de vértices) de um grafo G é uma rotulação $f : V(G) \rightarrow \mathcal{C}$, onde $|\mathcal{C}| = k$. Os rótulos são usualmente chamados **cores**; os vértices de uma cor formam uma **classe de cor**. Uma k -coloração é **própria** se vértices adjacentes têm rótulos diferentes. Um grafo é k -colorível se este tem uma k -coloração própria. O **número cromático** $\chi(G)$ é o menor k tal que G é k -colorível.

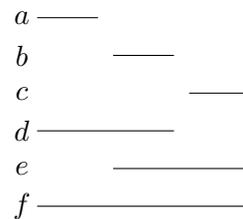
Um grafo G é **perfeito** se $\chi(G) = \omega(G)$ para todo subgrafo induzido H de G . O teorema forte dos grafos perfeitos indica que um grafo G é perfeito se, e somente se, ambos G e seu complementar, \bar{G} , não têm ciclos ímpares induzidos de tamanho pelo menos 5. Um ciclo ímpar induzido em G de tamanho pelo menos 5 é chamado **buraco ímpar** e em \bar{G} é chamado de **anti-buraco ímpar**.

Uma subclasse dos grafos perfeitos são os grafos cordais. Um grafo é **cordal** se todo ciclo de quatro ou mais vértices tem uma corda, que é uma aresta que não é parte do ciclo mas conecta dois vértices deste ciclo.

Uma representação em intervalos de um grafo é uma família de intervalos, um atribuído a cada vértice, de modo que dois vértices são adjacentes se, e somente se, seus intervalos se interceptam. Um grafo que tem uma representação em intervalos é chamado **grafo de intervalos**. A Figura 2.5(a) mostra um grafo de intervalo e na Figura 2.5(b) temos sua representação em intervalos. Podemos mostrar que todo grafo de intervalos é cordal.



(a) Grafo de Intervalos G .



(b) Representação em Intervalos de G .

Figura 2.5: Exemplo de Grafo de Intervalo e Sua Representação em Intervalos.

2.2 Problemas de Otimização

Um **problema de otimização** basicamente se compõe em três partes: um conjunto de instâncias; um conjunto $SOL(I)$ de soluções viáveis para cada instância I ; e uma função que atribui um número $VAL(S)$ a cada solução viável S . O número $VAL(S)$ é o valor de S . Quando o conjunto $SOL(I)$ das soluções viáveis associado a uma instância I é vazio, dizemos que a instância é **inviável**; caso contrário, a instância é **viável**.

Um **problema de maximização** consiste em encontrar uma solução S de maior valor no conjunto $SOL(I)$. De forma análoga, um **problema de minimização** consiste em encontrar uma solução de valor mínimo no conjunto de soluções viáveis. Esta solução S^* a ser encontrada é chamada de solução ótima, cujo valor é ótimo da instância, isto é, $OPT(I) = VAL(S^*)$.

Conseqüentemente, para problemas de maximização,

$$VAL(S^*) \geq VAL(S), \forall S \in SOL(I).$$

2.2.1 PPLs e PLIs

Muitos problemas de otimização podem ser modelados através de **Programação Linear**. Basicamente, a partir de um conjunto de variáveis reais, defini-se uma função objetivo linear neste conjunto de variáveis, a qual se deseja maximizar (ou minimizar), e representa-se o conjunto de soluções viáveis por meio de desigualdades lineares, chamadas de restrições. Um **Problema de Programação Linear** (PPL) convencional pode ser definido da seguinte forma:

$$\max z = \sum_{j=1}^p c_j x_j \quad (2.1)$$

Sujeito a:

$$\sum_{j=1}^p a_{ij} x_j \leq b_i, i = 1, 2, \dots, q \quad (2.2)$$

$$x_j \geq 0, j = 1, 2, \dots, p, \quad (2.3)$$

onde c_j , a_{ij} e b_j são constantes reais conhecidas e x_j são as incógnitas chamadas de variáveis de decisão do problema. A função linear a ser maximizada em (2.1) é denominada função objetivo, função econômica ou função critério. As restrições de não negatividade (2.3) são chamadas de restrições triviais.

O conjunto de restrições que compõem o PPL define o poliedro que caracteriza a região de viabilidade. É possível demonstrar que, se existe solução ótima, então uma delas ocorre em um dos vértices deste poliedro. O método Simplex, introduzido por Dantzig em 1947, explora tal propriedade. Ele percorre vértices cada vez melhores (em relação à função objetivo) até achar o ótimo. Embora de complexidade de pior caso exponencial, esse algoritmo é um dos mais conhecidos e mais comumente utilizados em Programação Linear. Ressalta-se, porém, que há outros algoritmos (de pontos interiores) cuja complexidade é polinomial.

Quando nos problemas de programação linear for necessário que algumas ou todas as variáveis de decisão assumam somente valores inteiros, estamos diante de um problema de **Programação Linear Inteira** (PLI). De modo geral, um PLI pode ser representado pela seguinte formulação:

$$\max z = \sum_{j=1}^{p'} c_j x_j + \sum_{j=1}^{p''} c_{(p'+j)} y_j \quad (2.4)$$

Sujeito a:

$$\sum_{j=1}^{p'} a_{ij} x_j + \sum_{j=1}^{p''} a_{i(p'+j)} y_j \leq b_i, i = 1, 2, \dots, q \quad (2.5)$$

$$x_j \in \mathbb{Z}_+, j = 1, 2, \dots, p' \quad (2.6)$$

$$y_j \geq 0, j = 1, 2, \dots, p'', \quad (2.7)$$

Portanto, a mudança básica entre um PPL e um PLI é o uso de variáveis inteiras no último. De modo mais preciso, quando uma formulação define algumas variáveis reais e outras inteiras, chamamos de problema de **Programação Linear Misto** (PLM).

Comumente, problemas combinatórios NP-Difíceis podem ser modelados como PLIs (ou PLMs) e, neste caso, os métodos utilizados para resolvê-los são mais complexos que o algoritmo Simplex, que, na maioria das vezes, é a principal escolha na resolução dos PPLs.

2.2.2 Relaxação Lagrangeana

Seja (P) o seguinte problema de programação inteira:

$$\max cx \quad (2.8)$$

Sujeito a:

$$Ax \leq b \quad (2.9)$$

$$Dx \leq d \quad (2.10)$$

$$x \in \mathbb{Z}_+^n, \quad (2.11)$$

onde $c^T \in \mathbb{R}^n$, $x \in \mathbb{R}^n$, A uma matriz $m \times n$, $b \in \mathbb{R}^m$, D uma matriz $p \times n$, $d \in \mathbb{R}^p$.

Associamos às restrições (2.9) o vetor linha $u = (u_1, u_2, \dots, u_m)$, tal que $u \geq 0$, conhecido como vetor dos multiplicadores de Lagrange. Então, para $u \geq 0$, escrevemos o seguinte problema:

$$\mathcal{L}(u) = \max cx + u(b - Ax) \quad (2.12)$$

Sujeito a:

$$Dx \leq d \quad (2.13)$$

$$x \in \mathbb{Z}_+^n, \quad (2.14)$$

Este problema denominamos de relaxação lagrangeana de (P) .

Supomos que as restrições (2.9), (2.10) e (2.11) definem um conjunto não vazio. Dessa forma, podemos mostrar que $\mathcal{L}(u) \geq OPT(P)$, onde $OPT(P)$ denota o valor de uma solução ótima de P . Esta propriedade nos diz que $\mathcal{L}(u)$ é uma cota superior de $OPT(P)$. Visando à busca da menor cota possível, consideramos o problema abaixo, o qual chamamos de (D) .

$$\min \mathcal{L}(u) \quad (2.15)$$

Sujeito a:

$$u \geq 0, \quad (2.16)$$

Chamamos (D) de problema dual associado à relaxação lagrangeana de (P) . Se u^* for uma solução ótima de (D) , teremos $OPT(D) = \mathcal{L}(u^*)$.

A seguir, mostramos o método do sub-gradientes, que pode ser utilizado para resolver (D) . Iniciamos por fazer algumas considerações e definições.

É possível provar que $\mathcal{L}(u)$ é uma função convexa e afim por partes. Dizemos que um vetor $\gamma \in \mathbb{R}^n$ é um sub-gradiente de uma função convexa $f : \mathbb{R}^n \rightarrow \mathbb{R}$, no ponto x^0 se $f(x) \geq f(x^0) + \gamma^T(x - x^0)$ para todo $x \in \mathbb{R}^n$. Definimos o sub-diferencial $\partial f(x^0)$ de f em x^0 como o conjunto de todos os sub-gradientes de f em x^0 .

Se f é diferenciável em x^0 então $\partial f(x^0) = \{\nabla f(x^0)\}$, onde $\nabla f(x^0)$ representa o gradiente de f em x^0 . É possível mostrar que $b - A\bar{x}$ é um sub-gradiente de \mathcal{L} em u^0 , onde \bar{x} é uma solução ótima para (2.12)-(2.14) quando $u = u^0$.

Agora, definimos o método dos sub-gradientes. Partimos de um ponto $u^0 \geq 0$ arbitrário e definimos uma sequência de pontos:

$$u^{k+1} = u^k + \theta_k \times \gamma^k, \text{ onde } \gamma^k \in \partial \mathcal{L}(u^k)$$

Depois projetamos u^{k+1} no primeiro octante para obter multiplicadores não negativos. Em outras palavras, a i -ésima componente de u^{k+1} é dada finalmente por

$$u_i^{k+1} = \max\{0, u_i^{k+1}\}$$

Uma escolha do passo θ_k , para $\gamma^k \neq 0$ é:

$$\theta_k = \rho_k \frac{\overline{\mathcal{L}} - \underline{\mathcal{L}}}{\|\gamma^k\|^2}, \text{ para } 0 < \rho \leq 2,$$

onde $\overline{\mathcal{L}}$ é uma cota superior de $OPT(D)$ e $\underline{\mathcal{L}}$ é uma cota inferior de $OPT(D)$. Estas cotas são atualizadas durante as iterações do método, visando diminuir o salto $\overline{\mathcal{L}} - \underline{\mathcal{L}}$. Em geral, sugere-se que ρ_k comece com valor 2 e seja dividido pela metade a cada p iterações sem que $\mathcal{L}(u^k)$ diminua. Um valor recomendado para p é 30.

2.2.3 Algoritmos Aproximativos e Heurísticas

Uma vez que muitos problemas de otimização conhecidos são NP-Difíceis, não há esperanças de obter algoritmos polinomiais para eles (a não ser que $P=NP$). Nessa situação, dentro de um contexto prático, passa a ser viável o uso de algoritmos polinomiais que encontrem soluções próximas das soluções ótimas. Se tal algoritmo polinomial garante uma proximidade de solução para qualquer instância do problema, dizemos que o algoritmo é **aproximativo**; caso contrário, chamamos genericamente o algoritmo de **heurística**. A seguir, definimos formalmente um algoritmo aproximativo.

Considere um problema de otimização em que $VAL(S) \geq 0$ para toda solução viável S de qualquer instância do problema. Seja A um algoritmo que, para toda instância viável I do problema, devolve uma solução viável $A(I)$ de I . Se o problema é de maximização e

$$VAL(A(I)) \geq \alpha \cdot OPT(I)$$

para toda instância I , dizemos que A é uma α -aproximação para o problema. O fator α é um número que pode depender de I . Dizemos que α é a razão de aproximação (*approximation*

factor) do algoritmo. Obviamente α é tal que $0 < \alpha \leq 1$, uma vez que o problema é de maximização.

2.3 Poliedros

Um conjunto $\mathcal{P} \subseteq \mathbb{R}^n$ é um **poliedro** se

$$\mathcal{P} = \{x \in \mathbb{R}^n : Ax \leq b\},$$

para alguma matriz $A \in \mathbb{R}^{m \times n}$ e algum vetor $b \in \mathbb{R}^m$.

Um poliedro \mathcal{P} definido por uma matriz A e por um vetor b é denotado por $\mathcal{P}(A, b)$. Se \mathcal{P} é um poliedro limitado (ou seja, existe um real α tal que todo ponto $x \in \mathcal{P}$ satisfaz $\|x\| \leq \alpha$), então \mathcal{P} é chamado **politopo**.

Se $a \in \mathbb{R}^n \setminus \{0\}$ e $\alpha \in \mathbb{R}$, então o poliedro $\{x \in \mathbb{R}^n : a^T x \leq \alpha\}$ é chamado de **semi-espaço**. Um conjunto de pontos que divide o espaço em dois semi-espaços, ou seja, um conjunto da forma $\{x \in \mathbb{R}^n : a^T x = \alpha\}$, onde a é não nulo, é chamado **hiperplano**.

Observando que cada uma das linhas da matriz A juntamente com a correspondente entrada do vetor b define um semi-espaço, segue imediatamente que $\mathcal{P}(A, b)$ é intersecção de um número finito de semi-espaços. Portanto, o conjunto de restrições de um PPL define um poliedro e, assim, o conjunto dos pontos viáveis dele correspondem aos pontos do poliedro associado.

2.3.1 Faces

Seja $S \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^n$ e $\alpha \in \mathbb{R}$. Uma inequação $a^T x \leq \alpha$ é válida em relação a S se $S \subseteq \{x \in \mathbb{R}^n : a^T x \leq \alpha\}$. Quando o conjunto S está claro pelo contexto, dizemos simplesmente que a inequação é válida.

Seja $\mathcal{P} \subseteq \mathbb{R}^n$ um poliedro. Um conjunto $F \subseteq \mathcal{P}$ é uma **face** de \mathcal{P} , se existe uma inequação $a^T x \leq \alpha$ válida em relação a \mathcal{P} , tal que

$$F = \mathcal{P} \cap \{x : a^T x = \alpha\}.$$

Dizemos que F é uma **face própria** se $F \neq \mathcal{P}$; e F é não-trivial se $\emptyset \neq F \neq \mathcal{P}$. Se $a^T x \leq \alpha$ é válida em relação a \mathcal{P} , então dizemos que $\mathcal{P} \cap \{x : a^T x = \alpha\}$ é a face induzida ou definida por $a^T x \leq \alpha$.

2.3.2 Dimensão de um Conjunto e Facetas

Dados $S = \{v^1, v^2, \dots, v^p\} \subset \mathbb{R}^n$ e $\alpha_1, \alpha_2, \dots, \alpha_p \in \mathbb{R}$, dizemos que o vetor $v = \sum_{i=1}^p \alpha_i v^i$ é uma:

- combinação convexa de S , se $\sum_{i=1}^p \alpha_i = 1$ e $\alpha_i \geq 0$, para todo $i = 1, 2, \dots, p$.
- combinação afim de S , se $\sum_{i=1}^p \alpha_i = 1$;

O **fecho convexo** (resp. afim) de um conjunto não-vazio $S \subseteq \mathbb{R}^n$, denotado por $CONV(S)$ (resp. $AFIM(S)$), é o conjunto de todas as combinações convexas (resp. afins) de seus subconjuntos finitos. Se $S = \emptyset$ então $CONV(S) = AFIM(S) = \emptyset$.

Se $S = \mathcal{P} \cap \mathbb{Z}^n$, onde \mathcal{P} é um poliedro, então $CONV(\mathcal{P})$ é um poliedro. Além disso, maximizar uma função linear sobre S é equivalente a maximizar tal função sobre $CONV(S)$. Desta maneira, um PLI poderia ser resolvido como um PPL, caso soubéssemos determinar o fecho convexo dos seus pontos viáveis.

Um conjunto de vetores $S \subset \mathbb{R}^n$ é dito **afim-independente** (AI) se

$$\forall S' \subseteq S, \text{ finito: } \sum_{v \in S'} \alpha_v v = 0, \sum_{v \in S'} \alpha_v = 0 \Rightarrow \alpha = 0.$$

Caso contrário é dito **afim-dependente** (AD).

A **dimensão** de um conjunto F , denotada por $DIM(F)$ é d se F contém exatamente $d + 1$ vetores afim-independentes. Em outros termos, $DIM(F) = DIM(AFIM(F))$.

Seja $\mathcal{P} \subseteq \mathbb{R}^n$ um poliedro. Se $DIM(\mathcal{P}) = n$, dizemos que \mathcal{P} tem **dimensão plena**. Podemos mostrar que as seguintes assertivas são equivalentes:

- \mathcal{P} tem dimensão plena;
- \mathcal{P} não está contido em nenhum hiperplano;
- $\mathcal{P} \subseteq \{x \in \mathbb{R}^n : a^T x = \alpha\} \Rightarrow a = 0, \alpha = 0$.

Uma face não-trivial F de \mathcal{P} é uma **faceta** se F não está contida em nenhuma outra face própria de \mathcal{P} . Equivalentemente, F é uma faceta de \mathcal{P} se, e somente se, $DIM(F) = DIM(\mathcal{P}) - 1$. Consequentemente, temos que, para ser faceta, F contém $DIM(\mathcal{P})$ vetores afim-independentes.

As desigualdades indutoras de facetas são exatamente aquelas necessárias e suficientes para definir o poliedro. Daí a importância de se estudar desigualdades indutoras de facetas para o poliedro dado pelo fecho convexo de conjunto viáveis de PLIs. As caracterizações acima fornecem as duas formas principais de se determinar facetas de um poliedro \mathcal{P} .

Particularmente, supondo que $\mathcal{P} \subseteq \mathbb{R}^n$ tem dimensão plena e $\pi^T x \leq \pi_0$ é válida para \mathcal{P} , elas se traduzem como segue:

- (i) $F = \{x \in \mathcal{P} : \pi^T x = \pi_0\}$, $\pi \neq 0$, é faceta de \mathcal{P} ;
- (ii) F possui n pontos afim-independentes;
- (iii) $F \subseteq \{x \in \mathcal{P} : \lambda^T x = \lambda_0\} \Leftrightarrow (\lambda^T, \lambda_0) = \alpha(\pi^T, \pi_0)$.

2.4 O Problema da Recoloração Convexa - RCC

Dada a semelhança entre ACV e o **Problema da Recoloração Convexa de Caminho**, RCC, apresentamos nesta seção alguns resultados e propriedades conhecidas para este problema. Iniciamos por definir formalmente o Problema RCC.

Definição 2.1. *Uma coloração é convexa se, para qualquer cor, todo subgrafo induzido por esta cor é conexo.*

Problema 2.2. *Dado um par (P, C) , onde P é um caminho e $C : V(P) \rightarrow \mathcal{C}$ uma coloração arbitrária de P , o Problema da Recoloração Convexa de Caminhos, RCC, consiste em encontrar uma recoloração convexa C^* de P que mude o menor número de cores, ou seja, $\text{custo}(C^*) = |\{v \in V(P) : C(v) \neq C^*(v)\}|$ seja o menor possível.*

Na região superior da Figura 2.6, temos uma instância do problema RCC. Tal instância consiste em um caminho com 6 vértices e uma coloração arbitrária dos vértices. A região inferior desta figura apresenta uma recoloração convexa ótima para esta instância que mantém a cor de 4 dos 6 vértices.

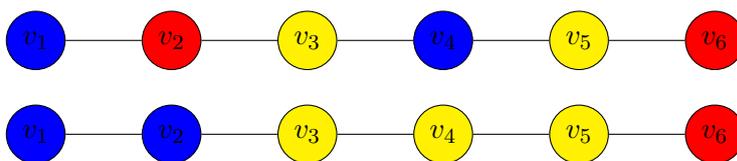


Figura 2.6: Exemplo de Instância do Problema RCC (Grafo Superior) e Sua Recoloração Convexa Ótima (Grafo Inferior)

Moran e Snir demonstraram que o problema RCC é NP-Difícil [12] e apresentaram um algoritmo 2-aproximativo para o mesmo [13]. Mostramos, a seguir, a formulação apresentada em [3] para o problema RCC. Para cada vértice $i \in V(P)$ e cor $c \in \mathcal{C}$, consideramos uma variável de decisão binária, x_{ic} , com o seguinte significado:

$$x_{ic} = \begin{cases} 1, & \text{se o vértice } i \text{ recebe a cor } c, \\ 0, & \text{C.C.} \end{cases}$$

Para representar a instância do problema, além do caminho P , consideramos as constantes w_{ic} para cada $i \in V(P)$ e $c \in \mathcal{C}$, que seguem a definição abaixo:

$$w_{ic} = \begin{cases} 1, & \text{se o vértice } i \text{ tem a cor } c, \\ 0, & \text{C.C.} \end{cases}$$

Observe que minimizar o custo de recoloração corresponde a minimizar a quantidade de cores que se modificam e, conseqüentemente, equivalente a maximizar o número de cores mantidas. A formulação apresentada abaixo explora esta propriedade. Por simplicidade, considere $V(P) = \{1, 2, \dots, N\}$ e $E(P) = \{(i, i + 1) : i = 1, \dots, N - 1\}$. A restrição (2.17) indica que todo vértice i de $V(P)$ deve receber uma cor pela nova coloração, e a restrição (2.18) garante que a coloração obtida será convexa. Note que a função objetivo maximiza o número de cores mantidas.

$$\max \sum_{i \in V(P)} \sum_{c \in \mathcal{C}} w_{ic} x_{ic}$$

Sujeito a:

$$\sum_{c \in \mathcal{C}} x_{ic} = 1, \forall i \in V(P) \quad (2.17)$$

$$x_{pc} - x_{rc} + x_{qc} \leq 1, \forall p, q \text{ e } r \text{ onde } 1 \leq p < r < q \leq N, \forall c \in \mathcal{C} \quad (2.18)$$

$$x_{ic} \in \{0, 1\}, \forall i \in V(P) \text{ e } \forall c \in \mathcal{C} \quad (2.19)$$

Um caso especial muito conhecido do problema RCC ocorre quando cada cor aparece no máximo 2 vezes na coloração inicial C . Este caso especial é conhecido como **2-RCC**. Em 2009, Kanj e Kratsch mostraram que este caso particular permanece NP-Difícil [14]. Lima apresentou em [3] um algoritmo 3/2-aproximativo para o problema 2-RCC.

Capítulo 3

O Problema ACV

Neste capítulo, apresentamos a definição formal do problema ACV, demonstramos que este problema é NP-Difícil a partir de uma redução do problema RCC e mostramos um algoritmo aproximativo para um caso particular (k -ACV).

3.1 Definições

Definição 3.1 (Atribuição Conexa). *Dados um conjunto $I = \{1, 2, \dots, M\}$ de símbolos e um conjunto $J = \{1, 2, \dots, N\}$ de posições, uma atribuição conexa é uma função $A : J \rightarrow I$, tal que para todo $i \in I$, existe um intervalo $[a_i, b_i] \subseteq J$, onde $A(k) = i$ se, e somente se, $k \in [a_i, b_i]$. Além disso, dizemos que $i \in I$ é atribuído por A se, e somente se, existe $j \in J$ tal que $A(j) = i$.*

Problema 3.2. *O Problema da Atribuição Conexa em Vetores, **ACV**, consiste em, dado um vetor v indexado por $J = \{1, 2, \dots, N\}$, um conjunto $I = \{1, 2, \dots, M\}$ e uma função $\rho : I \times J \Rightarrow \mathbb{Z}_+$, que define o ganho ao inserir $i \in I$ na posição $j \in J$ do vetor v , encontrar uma atribuição conexa $A : J \rightarrow I$ dos símbolos de I em v que maximize $\text{val}(A)$, onde:*
$$\text{val}(A) = \sum_{j=1}^N \rho_{A(j)j}.$$

Problema 3.3. *O Problema de Decisão da Atribuição Conexa em Vetores, **ACV-D**, consiste em, dado um vetor v indexado por $J = \{1, 2, \dots, N\}$, um conjunto $I = \{1, 2, \dots, M\}$, uma função $\rho : I \times J \Rightarrow \mathbb{Z}_+$ que define o ganho ao inserir i na posição j do vetor v e um valor inteiro k , decidir se existe uma atribuição conexa $A : J \rightarrow I$ dos símbolos de I em v tal que $\text{val}(A) \geq k$.*

Pela definição do problema de otimização ACV, observe que a tripla (N, M, ρ) define perfeitamente uma instância deste problema. Da mesma forma, a quádrupla (N, M, ρ, k) é suficiente para definir uma instância de ACV-D.

3.2 Complexidade

A seguir, mostramos que ACV-D é NP-Completo a partir de uma redução da versão de decisão do problema RCC-D.

Problema 3.4. *Seja k' um valor inteiro e (P, C) um caminho P com coloração arbitrária C . O Problema de Decisão da Recoloração Convexa de Caminhos, RCC-D, consiste em decidir se há uma recoloração convexa C' de (P, C) tal que $\text{custo}(C') \leq k'$, onde $\text{custo}(C') = |\{v \in V(P) : C(v) \neq C'(v)\}|$.*

Definição 3.5. *Dados um caminho colorido (P, C) e uma recoloração C' , definimos $\text{mantem}(C') = |V(P)| - \text{custo}(C') = |\{v \in V(P) : C(v) = C'(v)\}|$.*

Pela definição 3.5 podemos enunciar de forma equivalente o problema de decisão RCC-D da seguinte maneira:

Problema 3.6 (Redefinição do Problema RCC-D). *Seja k' um valor inteiro e (P, C) um caminho P com coloração arbitrária C . O Problema de Decisão da Recoloração Convexa de Caminhos, RCC-D, consiste em encontrar uma recoloração convexa C' de (P, C) tal que $\text{mantem}(C') \geq |V(P)| - k'$.*

3.2.1 Redução: RCC-D \rightarrow ACV-D

A redução se resume em mostrar algoritmos polinomiais f e g que convertem uma instância I de RCC-D numa instância de ACV-D e uma solução A de ACV-D numa solução de RCC-D, respectivamente. A seguir mostramos a redução e provamos que de fato ela é válida.

Seja $I = (P, C, k')$ uma instância de RCC-D, onde $P = \langle v_1, v_2, \dots, v_n \rangle$. Então, a instância $I_{ACV-D} = (N, M, \rho, k) = f(I)$ é tal que $N = |V(P)|$, $M = |\text{img}(C)|$, $\rho_{ij} = \begin{cases} 1, & \text{se } i = C(v_j) \\ 0, & \text{caso contrário} \end{cases}$ e $k = |V(P)| - k'$.

Sendo A uma solução da instância $f(I)$ do problema ACV-D, o algoritmo $g(A)$ obtém a recoloração C' de I_{RCC-D} fazendo $C'(v_j) = A(j)$.

Lema 3.7. *Sendo I_{RCC-D} uma instância de RCC-D com solução C' , então existe atribuição viável A para $f(I_{RCC-D})$, tal que $\text{mantem}(C') = \text{val}(A)$.*

Demonstração. Note que $A(j) = C'(v_j)$ é uma atribuição viável para $f(I_{RCC-D})$, pois C' é convexa e, por consequência, A deverá satisfazer as restrições de conexidade de ACV-D. Daí,

$val(A) = \sum_{j=1}^N \rho_{C'(v_j)j}$. Se $C'(v_j) = C(v_j)$, então $\rho_{C'(v_j)j} = \rho_{C(v_j)j} = 1$. Se $C'(v_j) \neq C(v_j)$, então $\rho_{C'(v_j)j} \neq \rho_{C(v_j)j} = 1 \Rightarrow \rho_{C'(v_j)j} = 0$. Portanto, $val(A) = |\{v_j \in V(P) : C(v_j) = C'(v_j)\}| = mantem(C')$. \square

Lema 3.8. *Seja I_{RCC-D} uma instância de RCC-D e $f(I_{RCC-D})$ uma instância de ACV-D com solução A , então $mantem(g(A)) = val(A)$.*

Demonstração. Observe que, sendo $A(j) = i$, na solução $g(A)$ temos $C'(v_j) = A(j) = i$. Se $\rho_{ij} = 1 \Rightarrow i = C(v_j) \Rightarrow C'(v_j) = C(v_j)$. Se $\rho_{ij} = 0 \Rightarrow i \neq C(v_j) \Rightarrow C'(v_j) \neq C(v_j)$. Portanto, cada unidade em $val(A)$ representa um cor mantida na solução $g(A)$ e, assim, $mantem(g(A)) = val(A)$. \square

Lema 3.9. *Sejam (P, C, k') e $(N, M, \rho, k) = f(P, C, k')$ instâncias de RCC-D e ACV-D, respectivamente. Então, existe solução C' de (P, C, k') se, e somente se, existe solução A de (N, M, ρ, k) .*

Demonstração. Se C' é solução de (P, C, k') , então, pelo Lema 3.7, existe atribuição viável A de (N, M, ρ, k) tal que $mantem(C') = val(A)$. Sendo $k = |V(P)| - k'$ e sabendo que $mantem(C') \geq |V(P)| - k'$, temos que $val(A) \geq k$; portanto A é solução para (N, M, ρ, k) . Agora, suponha que A seja solução de (N, M, ρ, k) . Pelo Lema 3.8, sabemos que $C' = g(A)$ é uma recoloração convexa para (P, C, k') tal que $mantem(C') = val(A)$. De forma análoga ao caso anterior, temos que $mantem(C') \geq |V(P)| - k'$ e, assim, C' é solução para (P, C, k') . \square

3.2.2 NP-Completo

O teorema seguinte demonstra a NP-Completo de ACV-D.

Teorema 3.10. *ACV-D é NP-Completo mesmo que ρ seja uma matriz binária com no máximo duas entradas não nulas por linha e uma única entrada não nula por coluna.*

Demonstração. Seja A uma atribuição para uma instância I_{ACV-D} do problema ACV-D. Observe que podemos verificar se A é conexa e computar $val(A)$ em $O(M + N)$, que é polinomial. Podemos verificar a conexidade em $O(M + N)$ percorrendo o vetor v uma única vez e utilizando um vetor auxiliar aux com uma posição para cada símbolo inicializado com 0. Em uma iteração j , se $aux[A(j)] = 0$ fazemos $aux[A(j)] = 1$, caso $aux[A(j)] = 1$ e $A(j-1) \neq A(j)$ concluímos que A não é conexa. Portanto, temos que o problema ACV-D pertence à classe NP. Pelo Lema 3.9, é possível decidir o Problema RCC-D, que é NP-Completo, a partir de ACV-D. Assim, segue que ACV-D deve também ser NP-Completo.

Em particular, perceba que ao considerar o problema NP-Completo 2-RCC, pela redução temos que a função ganho ρ se resume a uma matriz binária com no máximo 2 entradas não nulas por linha e uma única entrada não nula por coluna. Assim, o Problema ACV-D permanece NP-Completo mesmo com este tipo de matriz de ganho. \square

Corolário 3.11. *ACV é NP-Difícil mesmo que ρ seja uma matriz binária com no máximo duas entradas não nulas por linha e uma única entrada não nula por coluna.*

Pelo que foi estabelecido no Corolário 3.11, observe que o Problema ACV é NP-Difícil mesmo num caso aparentemente simples, que consiste de uma matriz binária com no máximo duas entradas não nulas por linha e com uma única entrada não nula por coluna. Dessa maneira, no caso de uma matriz arbitrária ρ , o problema é tão ou mais difícil que este caso particular.

3.3 Algoritmo Aproximativo Para o Problema k -ACV

Consideraremos agora um caso particular do problema ACV, para o qual mostramos um algoritmo aproximativo.

Problema 3.12 (Problema k -ACV). *O Problema k -ACV consiste do Problema ACV onde todo símbolo $i \in I$ possui ganho diferente de zero em no máximo em k posições indexadas por J .*

Como foi mostrado anteriormente, o problema ACV permanece NP-Difícil mesmo que a função de ganho ρ seja uma matriz binária com duas entradas não nulas por linha e uma entrada não nula por coluna. Desta maneira, a versão particular definida por k -ACV generaliza este caso NP-Difícil.

Apresentamos a seguir um algoritmo com fator de aproximação $1/k$ para k -ACV. Por simplicidade, consideramos que a função de ganho ρ é definida por uma matriz sem linhas nulas, isto é, todo símbolo possui ganho ao ser alocado em alguma posição j ; do contrário este símbolo nunca será alocado. Também consideramos que não há colunas nulas em ρ , ou seja, toda posição tem algum símbolo que resulte em ganho ao ser alocado nela; do contrário é indiferente a alocação de qualquer símbolo nesta posição.

Inicialmente, na linha 2 do Algoritmo 1, calcula-se uma atribuição A , não necessariamente conexa, que atribui o símbolo de maior ganho em cada posição. Com base nesta atribuição A , o algoritmo constrói uma atribuição conexa \hat{A} . No laço da linha 3, para cada

símbolo i que foi atribuído por A , o algoritmo considera as k possíveis posições onde A fez a alocação deste símbolo e o atribui na posição de maior ganho (linha 6). Ao final, na linha 8, o algoritmo completa a atribuição mantendo a conexidade e retorna tal atribuição na linha 9.

O passo da linha 8 pode ser implementado de várias formas. Seja $[a, b] \subseteq J$ um subintervalo não alocado por A (consideramos $[a, b]$ maximal, ou seja, não existe outro que o contém) e sejam $i_e = A(a - 1)$ e $i_d = A(b + 1)$ os símbolos alocados imediatamente à esquerda e à direita desse subintervalo, respectivamente. Caso $a = 1$ (resp. $b = N$) dizemos que i_e (resp. i_d) não está definido. De todo modo, pelo menos um dos dois, que denotaremos genericamente por i , está bem definido. Então, uma das formas mais simples de completar a atribuição consiste em alocar i em $[a, b]$. Podemos também pensar em outras estratégias mais engenhosas para completar a alocação. Supondo ambos i_d e i_e bem definidos, podemos determinar uma fronteira $f \in [a - 1, b]$ que separa a alocação de i_e em $[a, f]$ e de i_d em $[f + 1, b]$. A fronteira que maximiza o ganho no intervalo pode ser determinada em $O(b - a)$. Poderíamos também pensar em estratégias que consideram a alocação de um terceiro símbolo, ainda não atribuído por A , no preenchimento de $[a, b]$.

Algoritmo 1: k -ACV Apx1

Entrada: M, N e a função ganho ρ

Saída: Uma atribuição conexa

```

1 início
2   Calcule uma atribuição  $A$ , não necessariamente conexa, tal que  $A(j) \leftarrow$  qualquer
   símbolo de  $\arg \max_{i \in I} \{\rho_{ij}\}, \forall j \in J$ ;
3   para cada  $i$  atribuído por  $A$  faça
4      $J' = \{j \in J : A(j) = i\}$ ; //  $|J'| \leq k$ 
5     Seja  $j^*$  a posição onde  $i$  tem o maior ganho dentre todas de  $J'$ , isto é,
      $j^* \in \arg \max_{j \in J'} \{\rho_{ij}\}$ ;
6      $\hat{A}(j^*) \leftarrow i$ ;
7   fim
8   Complete a atribuição  $\hat{A}$  mantendo a conexidade;
9   retorna  $\hat{A}$ ;
10 fim

```

Mostramos abaixo o fator de aproximação do Algoritmo 1.

Lema 3.13. $val(\hat{A}) \geq \frac{1}{k} val(A)$.

Demonstração. Denote por $n_i = \{j \in J : A(j) = i\}$ as posições onde A aloca i , $S_A = \{i \in I : n_i \neq \emptyset\}$ os símbolos atribuídos por A e $j_i \in \arg \max_{j' \in n_i} \{\rho_{ij'}\}$ uma posição de maior ganho para $i \in S_A$ dentre todas aquelas em que i foi alocado por A . Nessa situação, observe

que

$$val(A) = \sum_{i \in S_A} \sum_{j \in n_i} \rho_{ij} \leq \sum_{i \in S_A} |n_i| \rho_{ij_i},$$

pois, $\forall i \in S_A$ e $\forall j' \in n_i$, temos: $\rho_{ij'} \leq \rho_{ij_i}$. Como $|n_i| \leq k$, encontramos:

$$val(A) \leq k \sum_{i \in S_A} \rho_{ij_i}$$

Por outro lado, note que $val(\hat{A}) \geq \sum_{i \in S_A} \rho_{ij_i}$, pois todo símbolo atribuído por A é alocado por \hat{A} na posição de maior ganho, dentre aquelas onde A atribui este símbolo. Daí,

$$val(A) \leq k \cdot val(\hat{A}) \Rightarrow val(\hat{A}) \geq \frac{1}{k} val(A).$$

□

Teorema 3.14. *O Algoritmo 1 é uma $1/k$ -Aproximação para o Problema k -ACV. Além disso, a razão $1/k$ é justa.*

Demonstração. Considere uma instância I do Problema k -ACV e seja A^* uma solução ótima de I . Observe que, como a atribuição inicial A do Algoritmo 1 aloca um símbolo de maior ganho em cada posição, temos que $val(A) \geq val(A^*)$. Pelo Lema 3.13, $val(\hat{A}) \geq \frac{1}{k} val(A)$, então $val(\hat{A}) \geq \frac{1}{k} val(A^*) = \frac{1}{k} OPT$.

Para mostrar que a razão é justa, tome a família de instâncias definidas pelas funções ganho ρ da Figura 3.1. Neste exemplo, temos $I = \{1, 2, \dots, k\}$ e $J = \{1, 2, \dots, k^2\}$. A Figura 3.2 mostra a atribuição inicial A que seria obtida na linha inicial do algoritmo. Observe que tal atribuição é a solução ótima A^* . Então $val(A) = val(A^*) = k \sum_{i=1}^{i=k} \gamma_i$. Imediatamente após a saída do laço das linhas 3-7 do algoritmo, a atribuição \hat{A} poderia ter sido calculada como mostrado pela atribuição na parte superior da Figura 3.3. Finalmente, a linha 8 do algoritmo dando preferência a i_d , levaria à atribuição mostrada na parte inferior desta mesma figura. Nesta situação, note que $val(\hat{A}) = (k-1)\gamma_k + \sum_{i=1}^{i=k} \gamma_i$. Daí, o fator de aproximação do algoritmo, para este conjunto de instâncias, pode ser $\frac{(k-1)\gamma_k + \sum_{i=1}^{i=k} \gamma_i}{k \sum_{i=1}^{i=k} \gamma_i}$, que converge para $1/k$ quando $\sum_{i=1}^{i=k} \gamma_i \gg (k-1)\gamma_k$. □

$i \setminus j$	1	...	k	$k+1$...	$2k$...	$k(k-1)+1$...	k^2
1	γ_1	...	γ_1	0	...	0	...	0	...	0
2	0	...	0	γ_2	...	γ_2	...	0	...	0
...
k	0	...	0	0	...	0	...	γ_k	...	γ_k

Figura 3.1: Função ganho ρ da família de instâncias tratada no Teorema 3.14.

1	...	k	$k+1$...	$2k$...	$k(k-1)+1$...	k^2
1	...	1	2	...	2	...	k	...	k

Figura 3.2: Atribuição inicial A do Algoritmo 1.

1	2	...	k	$k+1$	$k+2$...	$2k$...	$k(k-1)+1$	$k(k-1)+2$...	k^2
1	-	...	-	2	-	...	-	...	k	-	...	-

1	2	...	k	$k+1$	$k+2$...	$2k$...	$k(k-1)+1$	$k(k-1)+2$...	k^2
1	2	...	2	2	3	...	3	...	k	k	...	k

Figura 3.3: Na parte superior, atribuição \hat{A} após o laço. Na parte inferior, atribuição \hat{A} finalizada.

Capítulo 4

Formulações Matemáticas

Neste capítulo, apresentamos três formulações de programação inteira para o problema ACV. A primeira é inspirada na formulação do problema RCC [15] [16]; a segunda toma por base a formulação apresentada em [2], que utiliza uma matriz de padrões de atribuição para definir a alocação dos símbolos. A última se assemelha à segunda no que diz respeito ao uso de uma matriz de padrões, porém acrescenta um conjunto de variáveis com o propósito de indicar a utilização dos padrões. Ao final do capítulo, demonstramos o relacionamento que há entre as formulações.

4.1 Formulação Z

A formulação mostrada abaixo se baseia em [3]. Chamamos esta de Formulação Z. Para cada $i \in I$ e $j \in J$, definimos uma variável de decisão binária z_{ij} , que tem o seguinte significado:

$$z_{ij} = \begin{cases} 1, & \text{se } i \text{ é alocado na posição } j \text{ de } v, \\ 0, & \text{caso contrário.} \end{cases}$$

A Formulação Z consiste de apenas duas restrições. A primeira restrição, (4.1), estabelece que cada posição de v pode receber no máximo um símbolo. A segunda, (4.2), especifica que símbolos de mesmo valor em v devem estar em uma região contígua neste vetor, isto é, garante a conexidade da atribuição. De fato, ela assegura que, se um símbolo i é alocado a duas posições p e q , também será alocado a toda porção intermediária $r \in [p, q]$.

$$\max \sum_{i=1}^M \sum_{j=1}^N \rho_{ij} z_{ij}$$

Sujeito a:

$$\sum_{i=1}^M z_{ij} \leq 1, \forall j \in J \quad (4.1)$$

$$z_{ip} - z_{ir} + z_{iq} \leq 1, \forall p, q \text{ e } r \text{ onde } 1 \leq p < r < q \leq N, \forall i \in I \quad (4.2)$$

$$z_{ij} \in \{0, 1\}, \forall i \in I \text{ e } \forall j \in J \quad (4.3)$$

4.2 Formulação X

A formulação seguinte toma por base [2]. Esta formulação utiliza uma matriz de atribuições T , cujo propósito é garantir a condição de adjacência de símbolos de mesmo valor (garantir a conexidade da atribuição) em v . Cada coluna p da matriz T define um padrão de alocação (ou padrão de atribuição) e cada linha se refere a uma posição j do vetor v . A entrada t_{jp} indica se a posição j é utilizada pelo padrão p , isto é, se p considera j na alocação.

Um padrão p da matriz de padrões T define um intervalo não vazio $[a, b]$ de alocação no vetor v . Sendo p_1 e p_2 dois padrões distintos cujos intervalos são $[a_1, b_1]$ e $[a_2, b_2]$, respectivamente, então p_1 deve ser uma coluna antecedente a p_2 na matriz T se uma das seguintes condições é verdadeira: $(b_1 - a_1) < (b_2 - a_2)$; ou, $a_1 < a_2$ e $b_1 - a_1 = b_2 - a_2$. Dessa maneira, para construir T , consideramos todos os padrões de alocação possíveis e aplicamos a ordenação considerada. Para um instância de ACV onde $N = 4$, temos a seguinte matriz:

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

Observe que a sequência de padrões de T inicia com os N padrões que utilizam uma única posição, seguindo com os $N - 1$ padrões de 2 posições, $N - 2$ padrões de 3 posições, chegando ao único padrão que aloca todas as posições.

Sendo $N \times P$ a ordem da matrix T , observe que P é dado por:

$$P = \sum_{k=1}^N (N - k + 1) = \frac{1}{2}N^2 + \frac{1}{2}N = \binom{N+1}{2}$$

O termo do somatório refere-se à existência de $N - k + 1$ padrões de tamanho k . Consideramos $Q = \{1, 2, \dots, P\}$ o conjunto de índices dos padrões.

Para cada $i \in I$ e $p \in Q$, definimos uma variável de decisão binária com o seguinte significado:

$$x_{ip} = \begin{cases} 1, & \text{se } i \text{ segue o padrão de atribuição } p, \\ 0, & \text{caso contrário.} \end{cases}$$

Dessa maneira, obtemos a formulação abaixo. A restrição (4.4) estabelece que todo símbolo $i \in I$ deve seguir no máximo um padrão de atribuição. Se i não segue nenhum padrão de atribuição, significa que i não é alocado. A restrição (4.5) garante que haverá no máximo um símbolo alocado em cada posição do vetor v . A função objetivo calcula o valor da atribuição, a ser maximizada.

$$\max \sum_{i=1}^M \sum_{j=1}^N \rho_{ij} \sum_{p=1}^P t_{jp} x_{ip}$$

Sujeito a:

$$\sum_{p=1}^P x_{ip} \leq 1, \forall i \in I \quad (4.4)$$

$$\sum_{i=1}^M \sum_{p=1}^P t_{jp} x_{ip} \leq 1, \forall j \in J \quad (4.5)$$

$$x_{ip} \in \{0, 1\}, \forall i \in I \text{ e } \forall p \in Q \quad (4.6)$$

4.3 Formulação XY

A Formulação XY utiliza um conjunto de variáveis binárias, x_{ip} e y_p , e a matriz de padrões de atribuições T para modelar o problema ACV. Para cada símbolo i e para cada padrão de alocação p , definimos uma variável binária x_{ip} como antes, ou seja,

$$x_{ip} = \begin{cases} 1, & \text{se } i \text{ segue o padrão de atribuição } p, \\ 0, & \text{caso contrário.} \end{cases}$$

Para cada padrão p , definimos uma variável binária y_p :

$$y_p = \begin{cases} 1, & \text{se utiliza o padrão } p, \\ 0, & \text{caso contrário.} \end{cases}$$

A restrição (4.7) garante que todo símbolo i pode utilizar, no máximo, um padrão p . A restrição (4.8) permite a utilização de um padrão p para alocação de algum símbolo i somente se este padrão for escolhido, ou seja, $y_p = 1$. Por fim, a restrição (4.9) garante que não serão utilizados padrões conflitantes, isto é, padrões com interseção. Dizemos que dois padrões são conflitantes se ambos ocupam alguma posição em comum. Formalmente, p e p' são conflitantes se $\sum_{j=1}^N t_{jp} t_{jp'} \geq 1$. A função objetivo desta formulação calcula o valor do ganho da atribuição que deve ser maximizada.

$$\max \sum_{i=1}^M \sum_{j=1}^N \rho_{ij} \sum_{p=1}^P t_{jp} x_{ip}$$

Sujeito a:

$$\sum_{p=1}^P x_{ip} \leq 1, \forall i \in I \quad (4.7)$$

$$\sum_{i=1}^M x_{ip} \leq y_p, \forall p \in Q \quad (4.8)$$

$$y_p + y_{p'} \leq 1, \forall p, p' \in Q : p \text{ é conflitante com } p' \quad (4.9)$$

$$x_{ip} \in \{0, 1\}, \forall i \in I \text{ e } \forall p \in Q \quad (4.10)$$

$$y_p \in \{0, 1\}, \forall p \in Q \quad (4.11)$$

Observe que o conjunto de desigualdades (4.7) e (4.9) são independentes. Dessa maneira, a aplicação de alguma técnica de decomposição é imediata, por exemplo, Relaxação Lagrangeana. De fato, esta ideia inspirou o desenvolvimento da Heurística H_4 proposta no Capítulo 6.

4.4 Relacionamento Entre as Formulações

Nesta seção, apresentamos algumas relações entre as 3 formulações apresentadas anteriormente. A seguir mostramos que a Formulação X corresponde ao problema mestre obtido a partir da aplicação da Decomposição de Dantzig-Wolfe à Formulação Z. Ao final, fazemos um fortalecimento da Formulação XY e mostramos que este resulta na própria Formulação X.

4.4.1 Relação Entre as Formulações Z e X

Vamos demonstrar que a Formulação X consiste do problema mestre obtido a partir da aplicação da Decomposição de Dantzig-Wolfe à Formulação Z. Para isso, vamos considerar o conjunto $Z_i = \{z_i \in \{0, 1\}^N : z_{ip} - z_{ir} + z_{iq} \leq 1, 1 \leq p < r < q \leq N\}$. Nessa situação, a Formulação Z, equivale a:

$$\max \sum_{i=1}^M \sum_{j=1}^N \rho_{ij} z_{ij}$$

Sujeito a:

$$\sum_{i=1}^M z_{ij} \leq 1, \forall j \in J$$

$$z_i \in Z_i, \forall i \in I$$

Observe que temos $Z_i = Z_{i'}$ para todo i e $i' \in \{1, 2, \dots, M\}$, pois os coeficientes da restrições que definem Z_i e $Z_{i'}$ são invariantes com relação aos índices i e i' e, desta maneira, os pontos destes conjuntos são os mesmos. Além disso, os pontos não nulos de Z_i são exatamente as colunas da matriz T , ou seja, $Z_i = \{T_1, T_2, \dots, T_P\} \cup \{0\}$, onde T_j é a j -ésima coluna de T . Observe que $z_i \in Z_i$ se, e somente se, existe $\lambda_i \in \{0, 1\}^P$ tal que $z_i = \sum_{p=1}^P \lambda_{ip} T_p$, $\sum_{p=1}^P \lambda_{ip} \leq 1$. Considerando cada entrada $j \in J$ de z_i , temos que: $z_i \in Z_i \Leftrightarrow z_{ij} = \sum_{p=1}^P \lambda_{ip} t_{jp}, \forall j \in J, \lambda_i \in \{0, 1\}^P$. Consequentemente podemos reescrever a formulação como:

$$\max \sum_{i=1}^M \sum_{j=1}^N \rho_{ij} \sum_{p=1}^P t_{jp} \lambda_{ip}$$

Sujeito a:

$$\begin{aligned} \sum_{i=1}^M \sum_{p=1}^P t_{jp} \lambda_{ip} &\leq 1, \forall j \in J \\ \sum_{p=1}^P \lambda_{ip} &\leq 1, \forall i \in I \\ \lambda_{ip} &\in \{0, 1\}, \forall p \in \{1, 2, \dots, P\} \end{aligned}$$

Portanto, a Formulação X consiste do problema mestre obtido a partir da aplicação da Decomposição de Dantzig-Wolfe à Formulação Z pela enumeração dos pontos definidos pelas restrições (4.2).

4.4.2 Relação Entre as Formulações XY e X

Vamos mostrar uma maneira de fortalecer a Formulação XY que a torna equivalente a X. Inicialmente, observe que podemos substituir a restrição (4.8) por:

$$\sum_{i=1}^M x_{ip} = y_p, \forall p \in Q \quad (4.12)$$

A modificação trata-se apenas da troca da desigualdade de menor ou igual por uma igualdade. Tal mudança não modifica a solução ótima. De fato, considere (x, y) um ponto viável tal que existe $p \in Q$ onde $\sum_{i=1}^M x_{ip} < y_p$, ou seja, $\sum_{i=1}^M x_{ip} = 0$ e $y_p = 1$. Tomando $x' = x$ e y' obtido a partir de y trocando y'_p para zero, encontramos um ponto (x', y') equivalente a (x, y) , isto é, ambos os pontos definem a mesma atribuição.

Agora, lembre que dois padrões que se interceptam possuem uma posição em comum. Considere, então, uma posição $j \in J$ e todos os padrões que a contém, isto é, $\{p \in Q : t_{jp} =$

1}. Podemos fortalecer as restrições (4.9), substituindo por

$$\sum_{p \in Q} t_{jp} y_p \leq 1, \forall j \in J \quad (4.13)$$

Chamamos a formulação obtida de XY substituindo (4.8)-(4.9) por (4.12)-(4.13) de Formulação XY Fortalecida. A anunciada relação entre esta e a Formulação X é consequência da seguinte propriedade, cuja demonstração é imediata.

Lema 4.1. *Seja $(x, y) \in \mathbb{R}^{M \times P} \times \mathbb{R}^P$ satisfazendo (4.12). Então x satisfaz (4.5) se, e somente se, y satisfaz (4.13).*

Como consequência direta das demonstrações apresentadas até aqui, segue o corolário abaixo.

Corolário 4.2. *Denote por $RL(F)$ o valor da relaxação linear de uma Formulação F . Então, $RL(X) = RL(XY Fortalecida) \leq RL(Z)$.*

Capítulo 5

Estudo Poliédrico do Problema ACV

Neste capítulo, utilizamos a Formulação X para definir o poliedro \mathcal{P} associado ao problema ACV. Mostramos, inicialmente, a dimensão de \mathcal{P} . Em seguida, apresentamos uma classe de desigualdades válidas com lado direito igual a 1 e provamos que esta juntamente com as restrições de não-negatividade compreende todas as desigualdades indutoras de facetas com coeficientes binários. Generalizamos estas desigualdades obtendo uma classe mais ampla indutora de faceta do poliedro \mathcal{P} . Ao final, caracterizamos completamente o poliedro \mathcal{P} para os casos onde $M = 2$ ou $N = 2$.

5.1 O Poliedro \mathcal{P} Associado ao Problema ACV

Iniciamos por definir algumas notações. Seja p um padrão de atribuição, denotamos por: $J(p)$ o conjunto das posições ativas em p ; $P_{\cap}(p)$ o conjunto dos padrões que interceptam p ; e $P_{\supset}(p)$ o conjunto dos padrões que contém p . Abaixo, estes conjuntos são expressos formalmente.

$$J(p) = \{j \in J : t_{jp} = 1\} \quad (5.1)$$

$$P_{\cap}(p) = \{p' : J(p) \cap J(p') \neq \emptyset\} \quad (5.2)$$

$$P_{\supset}(p) = \{p' : J(p') \supseteq J(p)\} \quad (5.3)$$

Caso $|J(p)| = 1$, dizemos que p é um padrão unitário. Neste caso, se p ocupa apenas a posição j , por simplicidade podemos usar também j para identificar o padrão.

A seguir, reescrevemos a Formulação X utilizando as notações definidas. O poliedro associado ao problema ACV será definido com base nesta formulação.

$$\max \sum_{i \in I} \sum_{j \in J} \sum_{p: j \in J(p)} \rho_{ij} x_{ip}$$

Sujeito a:

$$\begin{aligned} \sum_{p \in Q} x_{ip} &\leq 1, \forall i \in I \\ \sum_{i \in I} \sum_{p: j \in J(p)} x_{ip} &\leq 1, \forall j \in J \\ x_{ip} &\in \{0, 1\}, \forall i \in I \text{ e } \forall p \in Q \end{aligned}$$

A Definição 5.1 expressa o poliedro \mathcal{P} associado ao Problema ACV, dado pela envoltória convexa dos pontos que satisfazem as restrições da Formulação X. A Definição 5.2 será frequentemente utilizada nas considerações de pontos de \mathcal{P} tratados nas demonstrações que seguem.

Definição 5.1. $\mathcal{P} = \text{CONV}(\{x \in \{0, 1\}^{M \times P} : \sum_{p \in Q} x_{ip} \leq 1, \forall i \in I; \text{ e } \sum_{i \in I} \sum_{p: j \in J(p)} x_{ip} \leq 1, \forall j \in J\})$

Definição 5.2. Dados $i \in I$ e $p \in Q$, denotamos por $x = e^{i \cdot p}$ o vetor $x \in \{0, 1\}^{M \times P}$ de \mathcal{P} tal que $x_{i', p'} = 1$ se, e somente se, $i' = i$ e $p' = p$.

Vamos analisar o poliedro \mathcal{P} somente para os casos onde $M \geq 2$. Observe que para $M = 1$ o problema é fácil. Basta perceber que, sendo $\rho_{ij} \geq 0, \forall i \in I$ e $\forall j \in J$, uma solução ótima consiste em alocar o único símbolo disponível em todas as posições.

Os primeiros resultados que mostramos tratam da dimensão do poliedro \mathcal{P} , das facetas triviais e de duas propriedades gerais sobre desigualdades indutoras de faceta deste poliedro. As Proposições 5.3 e 5.4 mostram que o poliedro \mathcal{P} tem dimensão plena e que as restrições de não-negatividade definem facetas. Mais ainda, no Lema 5.5 provamos que as únicas desigualdades relevantes na descrição de \mathcal{P} que tem algum coeficiente negativo são, de fato, aquelas definidas por $-x_{ip} \leq 0$; todas as demais devem ter coeficientes não negativos. Além disso, mostramos que, em qualquer outra desigualdade forte, os coeficientes associados ao padrão P são todos iguais ao lado direito.

Proposição 5.3. \mathcal{P} tem dimensão plena.

Demonstração. Seja $i' \in I$, $p' \in \{1, 2, \dots, P\}$ e $x = e^{i', p'}$. Observe que x pertence a \mathcal{P} e que existem $M \times P$ vetores deste tipo. Além disso, perceba que o vetor nulo também está presente em \mathcal{P} . Portanto, temos $M \times P + 1$ pontos afim-independentes em \mathcal{P} . Consequentemente, \mathcal{P} tem dimensão plena. \square

Proposição 5.4. *Para todo $i \in I$ e todo $p \in Q$, $-x_{ip} \leq 0$ define faceta de \mathcal{P} .*

Demonstração. Basta notar que o vetor nulo e os vetores $e^{i', p'}$ para $i' \neq i$ e $p' \neq p$, são $DIM(\mathcal{P})$ vetores afim-independentes que satisfazem $x_{ip} = 0$. \square

Lema 5.5. *Seja $\pi x \leq \pi_0$ uma desigualdade indutora de faceta de \mathcal{P} diferente de $-x_{ip} \leq 0$, para todo $i \in I$ e para todo $p \in Q$; Então $\pi \geq 0$.*

Demonstração. Tome uma faceta induzida por $\pi x \leq \pi_0$, diferente de $-x_{ip} \leq 0, \forall i \in I, \forall p \in Q$. Sejam i e p arbitrários. Observe que a faceta deve conter um ponto \bar{x} tal que $\bar{x}_{ip} = 1$. Se todos os pontos da faceta tivessem $x_{ip} = 0$, ela estaria no hiperplano $x_{ip} = 0$ e, assim, esta seria $-x_{ip} \leq 0$. Temos que $\pi \bar{x} = \pi_0$. Por outro lado, podemos definir outra solução viável \hat{x} , obtida a partir de \bar{x} , trocando apenas a componente ip para 0. Como essa solução é viável, satisfaz $\pi \hat{x} \leq \pi_0 = \pi \bar{x}$. Então, $\pi(\hat{x} - \bar{x}) \leq 0 \Rightarrow -\pi_{ip} \leq 0$, ou seja, $\pi_{ip} \geq 0$. Como i, p são arbitrários, temos $\pi \geq 0$. \square

Lema 5.6. *Seja $\pi x \leq \pi_0$, $\pi_0 \neq 0$, uma desigualdade indutora de faceta para \mathcal{P} . Então, sendo P o padrão que aloca todas as posições, $\pi_{iP} = \pi_0, \forall i \in I$.*

Demonstração. Seja $i \in I$. Como $e^{iP} \in \mathcal{P}$, temos que $\pi_{iP} \leq \pi_0$. Suponha, por absurdo, que exista $i' \in I$ tal que $\pi_{i'P} < \pi_0$. Defina π' a partir de π trocando $\pi'_{i'P}$ para π_0 . Temos que $\pi' \neq 0$. Vamos mostrar que $\pi' x \leq \pi_0$ é válida para \mathcal{P} . Seja $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x_{i'P} = 0$, então $\pi' x = \pi x \leq \pi_0$. Caso contrário, $x_{i'P} = 1$ e, consequentemente, todas as outras componentes de x são nulas. Daí, $\pi' x = \pi_0$. Portanto, $\pi' x \leq \pi_0$ é válida e domina $\pi x \leq \pi_0$: absurdo. \square

5.2 Todas as Desigualdades Fortes com Coeficientes Binários

Nesta seção, apresentamos uma classe de desigualdades válidas com lado direito igual a 1 e mostramos que esta define qualquer faceta com coeficientes binários para \mathcal{P} .

Seja $p \in \{1, 2, \dots, P\}$ um padrão e $i \in I$. Então a desigualdade (5.4) é válida para \mathcal{P} .

$$\sum_{i' \in I \setminus \{i\}} \sum_{p' \in P_{\supset}(p)} x_{i'p'} + \sum_{p' \in P_{\cap}(p)} x_{ip'} \leq 1 \quad (5.4)$$

Observe que o termo $\sum_{p' \in P_{\cap}(p)} x_{ip'}$ é limitado em uma unidade, pois envolve um único símbolo $i \in I$, e todo símbolo deve ser alocado a no máximo um padrão. Considerando primeiro $\sum_{p' \in P_{\cap}(p)} x_{ip'} = 1$, note que $\sum_{i' \in I \setminus \{i\}} \sum_{p' \in P_{\supset}(p)} x_{i'p'}$ deve ser zero, já que este termo considera apenas padrões p' que contém p . Como qualquer um destes padrões possui interseção não nula com qualquer outro de $P_{\cap}(p)$, nenhum poderá ser utilizado, obrigando a soma ser nula. Admita agora que $\sum_{p' \in P_{\cap}(p)} x_{ip'} = 0$. Perceba que $\sum_{i' \in I \setminus \{i\}} \sum_{p' \in P_{\supset}(p)} x_{i'p'}$ é limitado em 1, pois considera padrões em $P_{\supset}(p)$ e assim, ao atribuir um deles a algum símbolo, nenhum outro deste conjunto poderá ser utilizado por ter interseção não nula com o primeiro. De todo modo, temos que a soma dos dois termos não ultrapassa uma unidade. Portanto, a desigualdade (5.4) é válida para \mathcal{P} .

Teorema 5.7. *A desigualdade (5.4) induz uma faceta para \mathcal{P} , se $M \geq 2$.*

Demonstração. Dados $i \in I$ e $p \in Q$, seja $F_{ip} = \{x \in \mathcal{P} : \sum_{i' \in I \setminus \{i\}} \sum_{p' \in P_{\supset}(p)} x_{i'p'} + \sum_{p' \in P_{\cap}(p)} x_{ip'} = 1\} = \{x \in \mathcal{P} : \pi^T x = \pi_0\}$ a face definida por (5.4). Supondo $F_{ip} \subseteq \{x \in \mathcal{P} : \lambda^T(x) = \lambda_0\}$, vamos mostrar que $(\lambda^T, \lambda_0) = \alpha(\pi^T, \pi_0)$, isto é, as desigualdades representadas por (λ^T, λ_0) e por (π^T, π_0) são equivalentes e, por consequência, (5.4) induz faceta em \mathcal{P} .

Caso (1): Seja $p' \notin P_{\cap}(p)$. Tome $i' \in I \setminus \{i\}$, \bar{x} e $\hat{x} \in F_{ip}$ tais que $\bar{x} = e^{i',p}$ e $\hat{x} = e^{i',p} + e^{i,p'}$. Note que $\lambda(\hat{x}) = \lambda_0$ e $\lambda(\bar{x}) = \lambda_0 \Rightarrow \lambda(\hat{x} - \bar{x}) = 0 \Rightarrow \lambda_{ip'} = 0$. Com isso, temos que todo coeficiente $\lambda_{ip'}$ associado a i e a um padrão p' sem interseção com p tem valor igual a zero.

Caso (2): Seja $p' \in P_{\cap}(p)$. Tome $\hat{x} \in F_{ip}$ tal que $\hat{x} = e^{i,p'}$. Temos que $\lambda_{ip'} = \lambda_0$. Dessa maneira, todo coeficiente $\lambda_{ip'}$ associado a i e a um padrão p' que tem interseção com p tem valor igual λ_0 .

Observe que os 2 casos anteriores determinam todos os coeficiente λ associados a i . Os próximos 3 casos determinam os demais coeficientes para os símbolos diferentes de i .

Caso (3): Sejam $p' \notin P_{\supset}(p)$, $i' \in I \setminus \{i\}$. Então existe padrão \bar{p} tal que $J(\bar{p}) \subseteq J(p) \setminus J(p')$. Note que $\bar{p} \in P_{\cap}(p)$. Tome $\bar{x} = e^{i,\bar{p}}$ e $\hat{x} = e^{i,\bar{p}} + e^{i',p'}$, ambos em F_{ip} . Então, temos que $\lambda_{i'p'} = 0$. Dessa maneira, todo coeficiente $\lambda_{i'p'}$ associado a um símbolo i' diferente de i e a um padrão p' que não contém p tem valor igual a zero.

Caso (4): Sejam $p' \in P_{\supset}(p)$, $i' \in I \setminus \{i\}$. Tome \bar{x} e $\hat{x} \in F_{ip}$ tais que $\bar{x} = e^{i,p}$ e $\hat{x} = e^{i',p'}$. Então, $\lambda(\hat{x} - \bar{x}) = 0 \Rightarrow \lambda_{ip} = \lambda_{i'p'} \Rightarrow \lambda_{i'p'} = \lambda_0$. Dessa forma, todo coeficiente $\lambda_{i'p'}$ associado a um símbolo i' diferente de i e a um padrão p' que contém p tem valor igual a λ_0 .

Portanto, os coeficiente nulos em π também são nulos em λ e os não nulos em π são iguais a λ_0 em λ . Logo, as desigualdades são equivalentes e, assim, (5.4) induz uma faceta

de \mathcal{P} . □

Vale observar que (5.4) domina e generaliza, respectivamente, as desigualdades (4.4) e (4.5) da formulação. Primeiro perceba que, tomando algum $i \in I$ e $p = P$ (o padrão que aloca todas as posições), (5.4) torna-se $\sum_{i' \in I \setminus \{i\}} x_{i'P} + \sum_{p'=1}^P x_{i'p'} \leq 1$. Portanto, (5.4) domina as restrições (4.4) da formulação. Por outro lado, tomando qualquer padrão p tal que $|J(p)| = 1$, temos $J(p) = \{k\}$, daí $P_{\supset}(p) = P_{\cap}(p)$ é igual ao conjunto dos padrões que têm a posição k . Desta maneira, a desigualdade (5.4) relativa a p e a qualquer $i \in I$ é exatamente a restrição (4.5) da formulação, isto é, a desigualdade que garante que a posição k do vetor irá alocar no máximo um símbolo. Portanto, (5.4) generaliza as restrições (4.5) da formulação.

Agora, vamos mostrar que a desigualdade (5.4) compreende todas as desigualdades não-triviais (diferentes de $-x_{ip} \leq 0$) com coeficientes binários que definem facetas de \mathcal{P} . Para isso, considere $ax \leq 1$, com $a \in \mathbb{Z}^{M \times P}$, uma desigualdade indutora de faceta de \mathcal{P} . Note que, na verdade, a tem apenas coeficientes 0 ou 1. Denote por

$$P_a(i) = \{p \in Q : a_{ip} = 1\}$$

os padrões associados a i em a . Em outras palavras, $p \in P_a(i)$ se, e somente se, $a_{ip} = 1$.

Lema 5.8. *Seja P o último padrão, isto é, o padrão que aloca todas as posições. Então, $P \in P_a(i), \forall i \in I$.*

Demonstração. Suponha, por absurdo, que exista $i \in I$ tal que $P \notin P_a(i)$. Defina a' a partir de a trocando a'_{iP} para 1. Vamos mostrar que $a'x \leq 1$ é válida para \mathcal{P} . Seja $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x_{iP} = 0$, então $a'x = ax \leq 1$. Caso contrário, $x_{iP} = 1$. Observe que, nessa situação, $x_{i'p} = 0$ para toda combinação de i' e p com $i' \neq i$ ou $p \neq P$. Portanto, $a'x = 1$. Assim, $a'x \leq 1$ é válida e domina $ax \leq 1$: absurdo, pois esta última define faceta para \mathcal{P} . □

Lema 5.9. *Sejam $p' \in P_a(i')$ e $p'' \in P_a(i'')$, onde i' e i'' são símbolos distintos de I . Então, $p' \in P_{\cap}(p'')$.*

Demonstração. Suponha, por absurdo, que existam $p' \in P_a(i')$ e $p'' \in P_a(i'')$ tais que $p' \notin P_{\cap}(p'')$, para algum i' e i'' distintos de I . Tome $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$ tal que $x = e^{i'p'} + e^{i''p''}$. Daí, $ax = 2$: absurdo, pois $ax \leq 1$ é válida para \mathcal{P} . □

Lema 5.10. *Sejam $i', i'' \in I$ tais que $\bigcap_{p \in P_a(i')} J(p) \neq \emptyset$ e $\bigcap_{p \in P_a(i'')} J(p) \neq \emptyset$. Então $P_a(i') = P_a(i'')$.*

Demonstração. Como i' e i'' são indistintos, basta mostrar que $P_a(i') \subseteq P_a(i'')$ para $i' \neq i''$. Suponha, por absurdo, que $P_a(i') \not\subseteq P_a(i'')$. Então, deve existir $p \in P_a(i')$ tal que $p \notin P_a(i'')$. Defina a' a partir de a fazendo $a'_{i''p} = 1$. Vamos mostrar que $a'x \leq 1$ é válida para \mathcal{P} . Seja $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x_{i''p} = 0$, então $a'x = ax \leq 1$. Suponha, então, $x_{i''p} = 1$: conseqüentemente, $x_{i'p} = 0$. Tome x' a partir de x fazendo $x'_{i''p} = 0$ e $x'_{i'p} = 1$. Se $x_{i'p'} = 1$, para algum $p' \neq p$, faça também $x'_{i'p'} = 0$ e observe que $p' \notin P_a(i')$, ou seja, $a_{i'p'} = 0$. Do contrário, pela hipótese, $J(p) \cap J(p') \neq \emptyset$, contrariando o fato de x ser viável com $x_{i''p} = x_{i'p'} = 1$. Note que $x' \in \mathcal{P}$ e, dessa maneira, temos $a'x = ax' \leq 1$. Portanto, $a'x \leq 1$ é válida e domina $ax \leq 1$, absurdo! Assim, $P_a(i') \subseteq P_a(i'')$. \square

Lema 5.11. *Seja $i' \in I$ tal que $\bigcap_{p \in P_a(i')} J(p) = \emptyset$. Então, $\bigcap_{p \in P_a(i'')} J(p) \neq \emptyset, \forall i'' \in I \setminus \{i'\}$.*

Demonstração. Suponha, por absurdo, que exista $i'' \in I \setminus \{i'\}$ tal que $\bigcap_{p \in P_a(i'')} J(p) = \emptyset$. Dessa forma, devem existir p''_1 e p''_2 de $P_a(i'')$ onde $p''_1 \notin P_\cap(p''_2)$. Do mesmo modo, devem existir p'_1 e p'_2 de $P_a(i')$ tais que $p'_1 \notin P_\cap(p'_2)$. Considere $J(p''_1) = \{j_{min}^1, j_{min}^1 + 1, \dots, j_{max}^1\}$ e $J(p''_2) = \{j_{min}^2, j_{min}^2 + 1, \dots, j_{max}^2\}$. Suponha, sem perda de generalidade, que $j_{min}^1 < j_{min}^2$ ou $j_{min}^1 = j_{min}^2$ e $j_{max}^1 < j_{max}^2$. Pelo Lema 5.9, temos que p'_1 tem interseção não vazia com p''_1 e com p''_2 . Sabendo que qualquer padrão define um subintervalo do vetor e $J(p''_1) \cap J(p''_2) = \emptyset$, podemos concluir que $\{j_{max}^1, j_{min}^2\} \subseteq J(p'_1)$. Pelos mesmos argumentos, temos $\{j_{max}^1, j_{min}^2\} \subseteq J(p'_2)$. Daí, $\{j_{max}^1, j_{min}^2\} \subseteq J(p'_1) \cap J(p'_2)$: absurdo, pois, por hipótese, p'_1 e p'_2 tem interseção vazia. \square

Lema 5.12. *Sejam $i \in I$ e $p \in P_a(i)$. Então $p' \in P_a(i), \forall p' \in P_\supset(p)$.*

Demonstração. Suponha, por absurdo, que exista $p' \in P_\supset(p)$ tal que $p' \notin P_a(i)$. Defina a' a partir de a trocando $a'_{ip'}$ para 1. Vamos mostrar que $a'x \leq 1$ é válida para \mathcal{P} . Seja $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x_{ip'} = 0$, então $a'x = ax \leq 1$. Caso contrário, $x_{ip'} = 1$ e, conseqüentemente, $x_{ip} = 0$. Tome x' a partir de x fazendo $x'_{ip'} = 0$ e $x'_{ip} = 1$. Observe que $x' \in \mathcal{P}$ e que $a'x = ax' \leq 1$. Portanto, $a'x \leq 1$ é válida e domina $ax \leq 1$, absurdo! \square

Lema 5.13. *Existem $i^* \in I$ e $p^* \in Q$ tais que*

$$(i) P_a(i^*) = P_\cap(p^*)$$

$$(ii) P_a(i) = P_\supset(p^*), \forall i \neq i^*$$

Demonstração. Pelo Lema 5.11, existe $i^* \in I$ tal que $\bigcap_{p \in P_a(i^*)} J(p) \neq \emptyset, \forall i \neq i^*$. Pelo Lema 5.10, $P_a(i) = P_a(i'), \forall i, i' \neq i^*$. Logo, existe $J' \neq \emptyset$ tal que $\bigcap_{p \in P_a(i)} J(p) = J', \forall i \neq i^*$. Mais

ainda, J' é um subconjunto de inteiros consecutivos de $\{1, \dots, N\}$, pois cada $J(p)$ também o é. Portanto, existe $p^* \in Q$ tais que

$$\bigcap_{p \in P_a(i)} J(p) = J(p^*), \forall i \neq i^* \quad (5.5)$$

Vamos mostrar que i^* e p^* acima definidos satisfazem (i) e (ii).

Primeiro, suponha, por absurdo, que $P_a(i^*) \subseteq P_\cap(p^*)$, ou seja, existe $p \in P_a(i^*) \setminus P_\cap(p^*)$. Seja $i \neq i^*$. Por 5.5, deve existir $p' \in P_a(i)$ tal que $p' \notin P_\cap(p)$. Do contrário, se todo padrão em $P_a(i)$ interceptasse p , p também interceptaria p^* . Logo, $x = e^{ip'} + e^{i^*p} \in \mathcal{P}$ e $ax = 2$: absurdo, pois $ax \leq 1$ é válida para \mathcal{P} . Portanto, $P_a(i^*) \subseteq P_\cap(p^*)$.

Para mostrar a inclusão inversa, pelo Lema 5.12, é suficiente provar que os padrões unitários em $P_\cap(p^*)$ pertencem a $P_a(i^*)$. Seja p um tal padrão e suponha, por absurdo, que $p \notin P_a(i^*)$. Defina a' a partir de a trocando a_{i^*p} para 1. Vamos mostrar que $a'x \leq 1$ é válida para \mathcal{P} . Seja $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x_{i^*p} = 0$, então $a'x = ax \leq 1$. Caso contrário, $x_{i^*p} = 1$ e, conseqüentemente, $x_{ip'} = 0, \forall i \neq i^*$ e $\forall p' \notin P_a(i)$. Isto porque $p' \in P_\supset(p^*)$, por 5.5, e portanto p' contém p , que é um padrão unitário definido em uma posição de p^* . Logo, $ax = \sum_{\hat{p} \in P_a(i^*)} x_{i^*\hat{p}} + \sum_{i \neq i^*} \sum_{p' \in P_a(i)} x_{ip'} = 0$ e $a'x = 1$. Concluimos disto que $a'x \leq 1$ é válida e domina $ax \leq 1$: um absurdo. Pelo Lema 5.12, chegamos a $P_\cap(p^*) \subseteq P_a(i^*)$.

Para mostrar o item (ii), tome $i \neq i^*$. Por 5.5, $J(p^*) \subseteq J(p), \forall p \in P_a(i)$, ou seja, $p \in P_a(i)$ implica $p \in P_\supset(p^*)$. Tome agora $p \in P_\supset(p^*)$. Para obter $p \in P_a(i)$, pelo Lema 5.12, é suficiente mostrar que $p^* \in P_a(i)$. Suponha, por absurdo, que tal não ocorre e defina a' a partir de a , fazendo $a'_{ip^*} = 1$. Vamos mostrar que $a'x \leq 1$ é válida para \mathcal{P} . Seja $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x_{ip^*} = 0$, $a'x = ax \leq 1$. Caso contrário, $x_{ip^*} = 1$ e, conseqüentemente, $x_{ip'} = 0, \forall p' \in P_a(i^*)$. De fato, por 5.5, todo p' em $P_a(i)$ intercepta p^* . Tome x' obtido a partir de x , fazendo $x'_{ip^*} = 0, x'_{i^*p} = 1$ e $x'_{i^*p'} = 0, \forall p' \neq p$. Note que $x' \in \mathcal{P}$ e $a'x = ax' \leq 1$. Logo $a'x \leq 1$ é válida para \mathcal{P} e domina $ax \leq 1$: absurdo. \square

Teorema 5.14. *As desigualdades (5.4) compreendem todas as desigualdades indutoras de facetas de \mathcal{P} da forma $ax \leq 1$, com $a \in \mathbb{Z}^{M \times P}$.*

Demonstração. Seja $ax \leq 1$ uma faceta de \mathcal{P} . Pelo Lema 5.13, existem $i^* \in I$ e $p^* \in Q$ tais que $P_a(i^*) = P_\cap(p^*)$ e $P_a(i) = P_\supset(p^*), \forall i \neq i^*$. Daí, $ax = \sum_{i \in I} \sum_{p \in P_a(i)} x_{ip} = \sum_{i \in I \setminus \{i^*\}} \sum_{p \in P_\supset(p^*)} x_{ip} + \sum_{p \in P_\cap(p^*)} x_{i^*p} \leq 1$, que é a própria desigualdade (5.4) tomando $i = i^*$ e $p = p^*$. \square

Observação 5.15. *Seja $(a, a_0) \in \mathbb{Z}^{M \times P} \times \mathbb{Z}$ onde $ax \leq a_0$ define faceta de \mathcal{P} . As condições abaixo são equivalentes e decorrem diretamente do Lema 5.6.*

- i) $a_0 = 1$;
- ii) $a_{iP} = 1$, para algum $i \in I$;
- iii) $a \in \{0, 1\}^{M \times P}$, $a \neq 0$.

Portanto, se $ax \leq a_0$ define faceta de \mathcal{P} e uma destas três condições ocorre, concluímos que tal desigualdade é uma das desigualdades (5.4).

5.3 Uma Generalização da Faceta de Coeficientes Binários

Nesta seção, vamos mostrar uma classe de desigualdades válidas indutora de facetas de \mathcal{P} que generaliza as desigualdades (5.4) apresentadas anteriormente.

Sejam $I' \subseteq I$ um subconjunto não vazio de símbolos e p um padrão. Provamos, a seguir, que a desigualdade (5.6) é válida para \mathcal{P} se $|I'| \leq |J(p)|$, que generaliza (5.4) e que define faceta quando $|I'| = 1$ e $M \geq 2$ ou quando $2 \leq |I'| \leq |J(p)| - 1$ e $M \geq 3$.

$$\sum_{i' \in I'} \sum_{p' \in P_{\cap}(p)} \max(|I'| - (|J(p)| - |J(p') \cap J(p)|), 1) \cdot x_{i', p'} +$$

$$\sum_{i' \in I \setminus I'} \sum_{p' \in P_{\cap}(p)} \max(|I'| - (|J(p)| - |J(p') \cap J(p)|), 0) \cdot x_{i', p'} \leq |I'| \quad (5.6)$$

Nas demonstrações que seguem, considere $A(x) = \sum_{i' \in I'} \sum_{p' \in P_{\cap}(p)} \max(|I'| - (|J(p)| - |J(p') \cap J(p)|), 1) \cdot x_{i', p'}$, $B(x) = \sum_{i' \in I \setminus I'} \sum_{p' \in P_{\cap}(p)} \max(|I'| - (|J(p)| - |J(p') \cap J(p)|), 0) \cdot x_{i', p'}$ e $f(x) = A(x) + B(x)$. Então a desigualdade (5.6) é definida por $f(x) \leq |I'|$.

Inicialmente, observe que os coeficientes nas parcelas A ou B de f são invariantes com relação aos símbolos e dependem essencialmente do tamanho do padrão utilizado, sendo irrelevantes as posições consideradas por ele.

Chamamos de tamanho de interseção dos padrões p' e p a cardinalidade do conjunto $J(p') \cap J(p)$ e denotamos por $\cap_{p'p}$ este valor. Consideramos ainda $|I'| = |J(p)| - k$, com $0 \leq k \leq |J(p)| - 1$. Em A , observamos coeficientes iguais a 1 para variáveis associadas a padrões p' que têm tamanho de interseção com p entre 1 e k e coeficientes iguais a t para padrões p' cujo tamanho de interseção com p é igual $k + t$ com $1 \leq t \leq |J(p)| - k$. Similarmente, em B , percebemos coeficientes nulos para variáveis associadas a padrões p' que têm tamanho de interseção entre 1 e k e coeficientes iguais a t para padrões cujo tamanho

de interseção é $k + t$ com $1 \leq t \leq |J(p)| - k$. Isto é, os coeficientes que aparecem em A tem a mesma expressão daqueles que aparecem em B , com exceção dos nulos, que passam a ser iguais a 1 em A .

Lema 5.16. *Sejam $I' \subseteq I$ e p um padrão, tais que $|I'| = |J(p)| - k$ com $0 \leq k \leq |J(p)| - 1$, e $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$ com $f(x) \neq 0$. Então existe $x' \in \mathcal{P}$ que aloca apenas símbolos de I' de modo que $f(x) = f(x')$.*

Demonstração. Se x não aloca símbolos de $I \setminus I'$, então faça $x' = x$. Do contrário, seja \hat{x} uma solução obtida através de x , desalocando qualquer símbolo $i \in I$ alocado em padrões que não possuem interseção com p . Observe que $f(x) = f(\hat{x}) \neq 0$. Seja $S \neq \emptyset$ o conjunto de símbolos de $I \setminus I'$ que são alocados por \hat{x} e denote por $p_i \in P_\cap(p)$ o padrão utilizado por $i \in S$. Seja $i' \in \arg \max_{i \in S} |\cap_{pp_i}|$ o símbolo, dentre todos de S , que está alocado no padrão que tem o maior tamanho de interseção com p . Observe que, se $|\cap_{pp_{i'}}| \leq k$, então todos os coeficientes de \hat{x}_{ip_i} , $\forall i \in S$, são nulos. Dessa forma, obtemos x' a partir de \hat{x} desalocando todos os símbolos de S . Se $|\cap_{pp_{i'}}| = k + t$, com $t > 0$, note que, sendo l o número de símbolos de I' alocados pela solução \hat{x} em um dos padrões de $P_\cap(p)$, então $l \leq |J(p)| - (k + t)$. Isto é, $l \leq (|J(p)| - k) - t = |I'| - t$. Com isso, existe pelo menos um símbolo i'' de I' que não foi alocado em \hat{x} . Neste caso, obtemos x' a partir de \hat{x} , desalocando i' e alocando i'' no padrão $p_{i'}$. Se x' ainda aloca símbolo de $I \setminus I'$, aplique a mesma estratégia novamente até não restar símbolos deste conjunto alocado. \square

Lema 5.17. *Sejam $p \in Q$, $I' \subseteq I$ tal que $|I'| = |J(p)| - k$, para algum k onde $0 \leq k \leq |J(p)| - 1$, e $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$ uma solução com $f(x) \neq 0$ que aloca exclusivamente símbolos de I' . Então existe solução $x' \in \mathcal{P}$ que aloca um único símbolo de I' tal que $f(x) = f(x')$.*

Demonstração. Considere \hat{x} a solução obtida a partir de x desalocando qualquer símbolo $i \in I'$ que utiliza padrões que não têm interseção com p . Note que $f(x) = f(\hat{x}) \neq 0$. Considere os símbolos de I' alocados por \hat{x} (em padrões que interceptam p). Sejam i'_1, i'_2, \dots, i'_c os símbolos alocados em padrões cujo tamanho de interseção com p são l_1, l_2, \dots, l_c onde $1 \leq l_e \leq k, \forall e \in \{1, \dots, c\}$, e i_1, i_2, \dots, i_d os símbolos alocados em padrões cujo tamanho de interseção com p são $k + t_1, k + t_2, \dots, k + t_d$ com $t_e > 0, \forall e \in \{1, \dots, d\}$. Note que: (i) $c + d \cdot k + (t_1 + t_2 + \dots + t_d) \leq (l_1 + l_2 + \dots + l_c) + d \cdot k + (t_1 + t_2 + \dots + t_d) \leq |J(p)|$; (ii) $c + d \leq |I'| = |J(p)| - k$; e (iii) $f(x) = c + t_1 + t_2 + \dots + t_d$. Obtemos x' alocando um único símbolo $i \in I'$ em um padrão p' onde $|\cap_{pp'}| = k + f(x) = k + c + t_1 + t_2 + \dots + t_d$. Mostramos que tal padrão existe. Caso $d > 0$, por (i), temos: $c + k + (t_1 + t_2 + \dots + t_d) \leq |J(p)|$. Por

outro lado, se $d = 0$, por (ii), encontramos: $c + k \leq |J(p)|$. Em ambos os casos, a condição $k + c + t_1 + t_2 + \dots + t_d \leq |J(p)|$ se verifica, garantindo a existência do padrão desejado. Como $\cap_{pp'} = k + f(x)$, temos que $f(x') = f(\hat{x}) = f(x)$. \square

Proposição 5.18. *A desigualdade (5.6) é válida para \mathcal{P} , quando $1 \leq |I'| \leq J(p)$, $\forall I' \subseteq I$ e $\forall p \in Q$.*

Demonstração. Considere o problema: (P) $\max_{x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}} f(x)$, cujo valor ótimo é não nulo. Pelos lemas 5.16 e 5.17, concluímos que existe uma solução ótima x^* para (P) que aloca um único símbolo $i' \in I'$. Além disso, por (5.6), temos que i' é alocado em um padrão p' que contém p , levando a $f(x^*) = |I'|$. Concluímos que $f(x) \leq |I'|$, $\forall x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. Portanto, a desigualdade (5.6) é válida para \mathcal{P} , quando $1 \leq |I'| \leq J(p)$. \square

O próximo resultado estabelece que a desigualdade (5.6) é uma generalização de (5.4).

Proposição 5.19. *A desigualdade (5.6) generaliza a desigualdade (5.4).*

Demonstração. Seja $I' = \{i\}$. Perceba que $\max(|I'| - |J(p)| + \cap_{pp'}, 1)$ é o coeficiente da variável $x_{ip'}$, onde p' tem interseção não nula com p . Como $\cap_{pp'} \leq |J(p)|$, então $|I'| - |J(p)| + \cap_{pp'} \leq |I'| = 1$, implicando $\max(|I'| - |J(p)| + \cap_{pp'}, 1) = 1$. Assim, todos os coeficientes não nulos de A são iguais a 1. Além disso, devem estar associados ao símbolo i e a padrões que tem interseção não nula com p . De maneira similar, em B , $\max(|I'| - |J(p)| + \cap_{pp'}, 0)$ é o coeficiente da variável $x_{i'p'}$, onde $i' \in I \setminus \{i\}$ e p' tem interseção não nula com p . Se $\cap_{pp'} < |J(p)|$ então $|I'| - |J(p)| + \cap_{pp'} < |I'| = 1$. Em outras palavras, $\max(|I'| - |J(p)| + \cap_{pp'}, 0) = 0$ para p' que não contém p . Se p' contém p , então $\max(|I'| - |J(p)| + \cap_{pp'}, 0) = \max(|I'|, 0) = 1$. Desta maneira, se p' não contém p o coeficiente em B é 0, caso contrário será 1. Portanto, para $|I'| = |\{i\}| = 1$, a desigualdade (5.6) será $\sum_{p' \in P \cap (p)} x_{ip'} + \sum_{i' \in I \setminus \{i\}} \sum_{p' \in P \supset (p)} x_{i'p'} \leq 1$ que é a própria desigualdade (5.4). \square

O teorema seguinte mostra as condições suficientes para que (5.6) induza uma faceta de \mathcal{P} .

Teorema 5.20. *A desigualdade (5.6) induz uma faceta para \mathcal{P} quando $|I'| = 1$ e $M \geq 2$ ou quando $2 \leq |I'| \leq |J(p)| - 1$ e $M \geq 3$.*

Demonstração. Pelo Teorema 5.19 a desigualdade (5.6) generaliza (5.4). Portanto, pelo Teorema 5.7, esta deverá induzir faceta quando $|I'| = 1$ e $M \geq 2$.

Vamos analisar, agora, o caso onde $2 \leq |I'| \leq |J(p)| - 1$ e $M \geq 3$. Sejam $|I| = |J(p)| - k$, onde $1 \leq k \leq |J(p)| - 1$, $p \in Q$ e $F = \{x \in \mathcal{P} : f(x) = |I'|\}$ a face induzida pela desigualdade (5.6). Suponha que $F \subseteq F' = \{x \in \mathcal{P} : \lambda(x) = \lambda_0\}$.

Inicialmente, mostramos que os coeficientes nulos na desigualdade (5.6) também são nulos em $\lambda(x) = \lambda_0$. Primeiro, consideramos os coeficientes nulos $x_{i'p'}$ com $i' \in I$ e $p' \in Q$ tal que $J(p') \cap J(p) = \emptyset$. Como $|I'| \geq 2$, existe $i'' \in I' \setminus \{i'\}$. Defina $x^1, x^2 \in \mathcal{P}$ tais que $x^1 = e^{i''p}$ e $x^2 = x^1 + e^{i'p'}$. Observe que ambos os pontos pertencem a F . Desta maneira, temos: $\lambda(x^2) - \lambda(x^1) = 0 \Rightarrow \lambda_{i',p'} = 0$. Como foi visto, $f(x) = A(x) + B(x) \leq |I'|$ representa (5.6). Agora, consideramos os demais coeficientes nulos, que se tratam daqueles que aparecem na parcela $B(x)$ de $f(x)$. Tais coeficientes consistem daqueles associados às variáveis $x_{i'p'}$, onde $i' \in I \setminus \{I'\}$, $p' \in P \cap (p)$ com $\cap_{pp'} \leq k$. Seja $x^3 \in \mathcal{P}$ a solução que consiste em alocar cada símbolo de I' numa única posição de p que não pertence a p' . Observe que esta atribuição é possível, pois $|J(p)| - \cap_{pp'} \geq |J(p)| - k = |I'|$, isto é, o número de posições de p que não interceptam p' é pelo menos o número de símbolos de I' . Assim, todos os símbolos deste conjunto podem ser alocados. Como descrito, temos que $x^3 \in F$. Tomando $x^4 = x^3 + e^{i'p'} \in F$, concluímos que $\lambda_{i',p'} = 0$.

Fazemos agora a análise dos coeficientes não nulos, de acordo com o tamanho da interseção dos padrões com p , começando com aqueles onde tal tamanho é 1. Em outras palavras, mostramos agora que os coeficientes em λ associados a $x_{i'p'}$ são iguais para todo $i' \in I'$ e $p' \in P \cap (p)$ com $\cap_{pp'} = 1$. Consideramos dois casos:

- Sejam $i' \in I'$ e $p', p'' \in Q$ tais que $\cap_{pp'} = \cap_{pp''} = 1$, onde $J(p) \cap J(p') = \{j'\}$ e $J(p) \cap J(p'') = \{j''\}$. Vamos mostrar que $\lambda_{i',p'} = \lambda_{i',p''}$. Tome x^5 uma solução de \mathcal{P} que consiste em alocar i' em p' e alocar cada símbolo de $I' \setminus \{i'\}$ em um padrão unitário distinto definido por cada posição em $J(p) \setminus \{j', j''\}$. Seja $x^6 = x^5 - e^{\{i',p'\}} + e^{\{i',p''\}} \in \mathcal{P}$. Note que $f(x^5) = f(x^6) = |I'|$ e, portanto, $x^5, x^6 \in F$. Daí, $\lambda(x^5) - \lambda(x^6) = 0 \Rightarrow \lambda_{i',p'} = \lambda_{i',p''}$.
- Sejam $i', i'' \in I'$ símbolos distintos e $p' \in P \cap (p)$ com $\cap_{pp'} = 1$. Vamos mostrar que $\lambda_{i',p'} = \lambda_{i'',p'}$. Se p é um padrão unitário, então $e^{i',p'}$ e $e^{i'',p'}$ estão em F mostrando que $\lambda_{i',p'} = \lambda_{i'',p'}$. Caso contrário, consideramos 2 subcasos. Primeiramente, tratamos o caso onde $|I'| = 2$. Como $M \geq 3$, existe $i''' \in I \setminus I'$. Seja $j_{min} = \min(J(p))$ e p'' o padrão tal que $J(p'') = J(p) \setminus \{j_{min}\}$. Defina $x^7 = e^{i',j_{min}} + e^{i''',p''}$ e $x^8 = e^{i'',j_{min}} + e^{i''',p''}$. Observe que x^7 e x^8 pertencem a F . Dessa forma, temos $\lambda_{i',j_{min}} = \lambda_{i'',j_{min}}$.

Pelo item anterior, e o fato de $\cap_{pj_{min}} = \cap_{pp'} = 1$, temos que $\lambda_{i',p'} = \lambda_{i'',p'}$. Para $|I'| > 2$, sejam $i''' \in I' \setminus \{i', i''\}$, $S = I' \setminus \{i', i'', i'''\}$ e x^9 uma solução viável para \mathcal{P} que consiste em alocar i' no padrão j_{min} , utilizar os padrões unitários seguintes $(j_{min} + 1, j_{min} + 2, \dots)$ para alocar cada símbolo de S e o padrão p'' de tamanho $|J(p)| - (|I'| - 2) > 0$, que utiliza as últimas posições consideradas por p , para o símbolo i''' . Tome x^{10} outra solução viável de \mathcal{P} obtida a partir de x^9 por trocar i' por i'' , isto é, $x^{10} = x^9 - e^{i',j_{min}} + e^{i'',j_{min}}$. Note que o coeficiente da variável $x_{i''p''}$ é $\max(|I'| - |J(p)| + |J(p'')|, 1) = \max(|I'| - |J(p)| + (|J(p)| - (|I'| - 2)), 1) = \max(2, 1) = 2$. Assim, $f(x^9) = f(x^{10}) = |I'|$. Portanto, x^9 e x^{10} pertencem a F . Daí, $\lambda_{i',j_{min}} = \lambda_{i'',j_{min}}$, seguindo, pelo item anterior, que $\lambda_{i',p'} = \lambda_{i'',p'}$.

Os dois itens acima mostram que existe $\theta \in \mathbb{R}$ tal que $\lambda_{i',p'} = \theta, \forall i' \in I'$ e $\forall p' \in P_{\cap}(p)$ com $\cap_{pp'} = 1$.

Agora, consideramos os padrões com tamanho de interseção com p entre 2 e k . Desejamos mostrar que, sendo $i' \in I'$, temos $\lambda_{i',p'} = \theta$, para p' tal que $2 \leq \cap_{pp'} \leq k$. Sejam $p' \in P_{\cap}(p)$ tal que $2 \leq \cap_{pp'} \leq k$ e $x^{11} \in \mathcal{P}$ uma solução que aloca i' no padrão p' e utiliza padrões unitários, dados por posições de p não conflitantes com p' , para alocar todos os outros símbolos de I' . Observe que esta solução é possível, pois $|I'| - 1 = |J(p)| - k - 1 \leq |J(p)| - \cap_{pp'} - 1$. Note que, em x^{11} , resta uma posição $j \in J(p) \setminus J(p')$ não ocupada. Seja x^{12} uma solução obtida através de x^{11} desalocando i' e realocando-o na posição j , ou seja, $x^{12} = x^{11} - e^{i',p'} + e^{i',j}$. Perceba que $f(x^{11}) = f(x^{12}) = |I'|$ e, assim, ambas as soluções pertencem a F . Logo, $\lambda_{i',p'} = \lambda_{i',j} = \theta$.

Finalmente, analisamos os padrões cujo tamanho de interseção com p é maior que k . Queremos mostrar que, considerando $i' \in I'$, temos $\lambda_{i',p'} = (\cap_{pp'} - k)\theta$, para $p' \in P_{\cap}(p)$ onde $k + 1 \leq \cap_{pp'} \leq |J(p)|$. Sejam $p' \in P_{\cap}(p)$, onde $k + 1 \leq \cap_{pp'} \leq |J(p)|$, e x^{13} uma solução viável para \mathcal{P} que aloca i' segundo o padrão p' , e utiliza $|J(p)| - \cap_{pp'}$ padrões unitários não conflitantes com p' para alocar o mesmo número de símbolos de $I' \setminus \{i'\}$. Considere S os símbolos de $I' \cup \{i'\}$ usados nesta alocação. Observe que esta solução é possível, pois $|I' \setminus \{i'\}| \geq |I'| - 1 = (|J(p)| - k) - 1 = |J(p)| - (k + 1) \geq |J(p)| - \cap_{pp'}$. Tome x^{14} uma solução obtida a partir de x^{13} desalocando i' e alocando $\cap_{pp'} - k$ símbolos de $\{i'\} \cup (I' \setminus S)$ em $\cap_{pp'} - k$ padrões unitários distintos que estão contidos em $J(p') \cap J(p)$. Note que $|\{i'\} \cup (I' \setminus S)| \geq 1 + |I'| - |S| = 1 + |I'| - (1 + |J(p)| - \cap_{pp'}) = \cap_{pp'} + (|I'| - |J(p)|) = \cap_{pp'} - k$. Dessa forma, a cardinalidade de $\{i'\} \cup (I' \setminus S)$ é pelo menos $\cap_{pp'} - k$; portanto, x^{14} é uma solução válida. Perceba que $f(x^{13}) = (|J(p)| - \cap_{pp'}) + (|I'| - |J(p)| + \cap_{pp'}) = |I'|$ e

$f(x^{14}) = |I'|$. Assim, tais soluções pertencem a F . Sendo S' o conjunto de símbolos de $\cap_{pp'} - k$ símbolos de $\{i'\} \cup (I' \setminus S)$ alocados por x^{14} e denotando por p_i o padrão utilizado por $i \in S'$, temos: $\lambda(x^{13}) - \lambda(x^{14}) = \lambda_{i',p'} - (\sum_{i \in S'} \lambda_{i,p_i}) = 0 \Rightarrow \lambda_{i',p'} = (\cap_{pp'} - k)\theta$. Desta maneira, os coeficientes de $f(x)$ com valor $\cap_{pp'} - k$, associados a variáveis de padrões $p' \in P_{\cap}(p)$ com $\cap_{pp'} \geq k + 1$, são iguais a $(\cap_{pp'} - k)\theta$.

Para completar a demonstração, resta verificar o valor de λ_0 . Observe que $\lambda(x^5) = \lambda_0 \Rightarrow \lambda_0 = |I'| \cdot \theta$.

Portanto, a desigualdade (5.6) define faceta quando $2 \leq |I'| \leq |J(p)| - 1$ e $M \geq 3$.

□

Vale observar que a desigualdade (5.6) não induz faceta quando $|I'| = |J(p)|$. Nessa condição, tal desigualdade é dada por:

$$\sum_{i' \in I'} \sum_{p' \in P_{\cap}(p)} \max(\cap_{p'p}, 1)x_{i'p'} + \sum_{i' \in I'} \sum_{p' \in P_{\cap}(p)} \max(\cap_{p'p}, 0)x_{i'p'} \leq |I'| = |J(p)|$$

Como $\cap_{p'p} \geq 1$, para todo $p' \in P_{\cap}(p)$, então a desigualdade se resume em:

$$\sum_{i' \in I} \sum_{p' \in P_{\cap}(p)} \cap_{p'p} x_{i'p'} \leq |J(p)|$$

Vamos mostrar que a desigualdade é dominada.

Tome a desigualdade válida abaixo.

$$\sum_{i' \in I} \sum_{p' \in P_{\cap}(\bar{p})} x_{i'p'} \leq 1, \text{ onde } \bar{p} \text{ é um padrão unitário} \quad (5.7)$$

Tal desigualdade é uma das restrições da formulação. Ela indica que cada posição, dada pelo padrão unitário \bar{p} , deve alocar no máximo um símbolo.

Seja $S = \{\bar{p} \in P_{\cap}(p) : |J(\bar{p})| = 1\} = \{\bar{p}_1, \bar{p}_2, \dots, \bar{p}_{|J(p)|}\}$ o conjunto de padrões unitários definidos pelas posições de p . Considere o conjunto de desigualdades de (5.7) restrito a $\bar{p} \in S$. Ao somá-las, temos:

$$\sum_{i' \in I} \left[\sum_{p' \in P_{\cap}(\bar{p}_1)} x_{i'p'} + \dots + \sum_{p' \in P_{\cap}(\bar{p}_{|J(p)|})} x_{i'p'} \right] \leq |J(p)| \quad (5.8)$$

Note que se $p' \in P_{\cap}(\bar{p}_i)$, onde $\bar{p}_i \in S$, então $p' \in P_{\cap}(p)$. Além disso, se $p' \in P_{\cap}(p)$ significa que existe pelo menos um $\bar{p}_i \in S$ tal que $p' \in P_{\cap}(\bar{p}_i)$. Daí, $P_{\cap}(\bar{p}_1) \cup \dots \cup P_{\cap}(\bar{p}_{|J(p)|}) = P_{\cap}(p)$.

Perceba que, sendo $p' \in P_{\cap}(p)$, então a variável $x_{i'p'}$, $\forall i \in I$, é somada $\cap_{p'p}$ na expressão (5.8), uma vez para cada $\bar{p}_i \in S$ que o intercepta. Assim, a desigualdade (5.8) pode ser

expressa por:

$$\sum_{i' \in I} \sum_{p' \in P \cap (p)} \cap_{p'p} x_{i'p'} \leq |J(p)|$$

Portanto, a desigualdade (5.6), quando $|I'| = |J(p)|$, pode obtida utilizando-se um conjunto de desigualdades da formulação. Conseqüentemente, tal desigualdade não pode induzir faceta neste caso.

5.4 Descrição Completa de \mathcal{P} Para $M = 2$ ou $N = 2$

Nesta seção, caracterizamos completamente o poliedro \mathcal{P} para os casos onde $M = 2$ ou $N = 2$. Mostramos que tal poliedro é descrito pelas desigualdades (5.4) e $x \geq 0$.

Estudamos, a seguir, o poliedro \mathcal{P} através do seu Grafo de Conflitos G , onde $V(G) = \{x_{ip} : \forall i \in I \text{ e } \forall p \in Q\}$, $E(G) = \{x_{ip}x_{i'p'} : i = i' \text{ ou } J(p) \cap J(p') \neq \emptyset\}$.

A Figura 5.1 mostra o grafo de conflitos para o caso onde $M = N = 2$. Neste caso, observe que temos 2 símbolos e 2 posições no vetor. Desta maneira, existem apenas 3 padrões de atribuição. O grafo de conflitos consiste de 2 cliques, uma para cada conjunto de vértices associado a um mesmo símbolo, e arestas adicionais de extremos que envolvem vértices associados a variáveis de símbolos diferentes e padrões conflitantes.

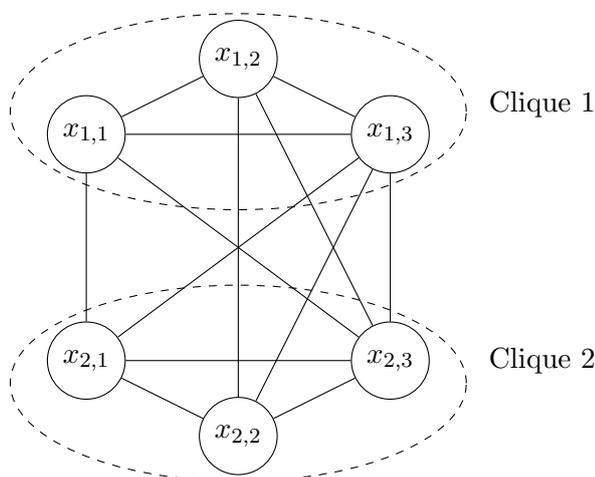


Figura 5.1: Grafo de Conflitos Para o Caso Onde $M = N = 2$

Vamos mostrar que, para $M = 2$ ou $N = 2$, \mathcal{P} é completamente descrito pelo conjunto de restrições de cliques maximais de G e que estas correspondem as desigualdades (5.4).

Lema 5.21. $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$ se, e somente se, x define um conjunto independente em G .

Demonstração. Suponha que x pertença a $\mathcal{P} \cap \mathbb{Z}^{M \times P}$. Se $x = 0$ ou se x atribui um único símbolo i' , trivialmente temos que x define um conjunto independente em G . Caso x aloque mais que um símbolo, observe que, para quaisquer i' e i'' distintos alocados por x segundo os padrões p' e p'' , respectivamente, temos $J(p') \cap J(p'') = \emptyset$. Daí, $x_{i'p'}x_{i''p''} \notin E(G)$ e, conseqüentemente, x deverá definir um conjunto independente em G .

Suponha, agora, que x defina um conjunto independente em G . Nessa condição, devemos ter $\sum_{p \in Q} x_{ip} \leq 1, \forall i \in I$, isto é, no máximo um único vértice do conjunto $\{x_{i1}, x_{i2}, \dots, x_{iP}\}$ pode ser escolhido, pois este define uma clique em G . Além disso, $\sum_{i \in I} \sum_{p \in \{p' \in Q: j \in J(p')\}} x_{ip} \leq 1, \forall j \in J$, pois o conjunto de vértices de G associados a padrões que possuem a posição j e qualquer símbolo i formam uma clique neste grafo, e, assim, podemos escolher no máximo um destes para o conjunto independente. Observando que as duas expressões equivalem, respectivamente, às restrições (4.4) e (4.5), consideradas na definição do poliedro, então, temos que $x \in \mathcal{P} \cap \mathbb{Z}^{M \times P}$. \square

Proposição 5.22. *Para $M = 2$ ou $N = 2$, o Grafo de Conflitos G de \mathcal{P} é um grafo perfeito.*

Demonstração. Para o caso em que $M = 2$, o Grafo de Conflitos G associado a \mathcal{P} é composto pelas cliques $C_1 = \{x_{1,1}, x_{1,2}, \dots, x_{1,P}\}$ e $C_2 = \{x_{2,1}, x_{2,2}, \dots, x_{2,P}\}$ com arestas adicionais entre $x_{1,p'}$ e $x_{2,p''}$ quando $J(p') \cap J(p'') \neq \emptyset$. Iremos mostrar que tal grafo não apresenta buracos ímpares ou anti-buracos ímpares. Conseqüentemente, tal grafo será um grafo perfeito. Considere um ciclo ímpar C de tamanho no mínimo 5 em G . Nessa situação, observe que pelos menos 3 dos vértices de C devem estar contidos em uma das duas cliques C_1 ou C_2 , portanto, C tem K_3 como subgrafo induzido e, dessa forma, não pode se tratar de um buraco ímpar. Considere \bar{G} o grafo complementar de G . Note que as duas cliques C_1 e C_2 , que contém todos os vértices, em G se tornam conjuntos independentes em \bar{G} . Dessa maneira, tal grafo é um grafo bipartido. Daí, \bar{G} não possui ciclos ímpares e, muito menos, buracos ímpares. Portanto, G não possui anti-buracos ímpares. Segue, então, o resultado do teorema.

Para o caso onde $N = 2$, observe que o conjunto de padrões Q é $\{1, 2, 3\}$ (o padrão 3 aloca as duas posições). Dessa maneira, o Grafo de Conflitos G é composto pelas cliques $C'_1 = \{x_{1,1}, x_{2,1}, \dots, x_{M,1}\}$ e $C'_2 = \{x_{1,2}, x_{2,2}, \dots, x_{M,2}\} \cup \{x_{1,3}, x_{2,3}, \dots, x_{M,3}\}$ com arestas adicionais de extremos $x_{w1} \in C'_1$ e $x_{k3} \in C'_2, \forall w, k \in M$. Além disso, para cada $i \in M$, têm-se uma aresta que associa os vértices $x_{i1} \in C'_1$ e $x_{i2} \in C'_2$. Utilizando argumentos análogos ao caso onde $M = 2$, temos que G é perfeito. \square

Teorema 5.23. *Para $M = 2$ ou $N = 2$, o poliedro \mathcal{P} é completamente descrito pelas desigualdades (5.4) e $x \geq 0$.*

Demonstração. Pelo Lema 5.21, \mathcal{P} é o politopo de conjuntos independentes de G . Como G é perfeito, para os casos considerados, tal politopo é descrito pelas desigualdades de cliques maximais e restrições de não-negatividade [17] (Corolário 65.2e). Logo, toda desigualdade indutora de faceta diferente de $x_{ip} \geq 0$ é da forma $ax \leq 1$. Pelo Teorema 5.14, $ax \leq 1$ é uma das desigualdades (5.4). \square

Capítulo 6

Heurísticas

Neste capítulo, apresentamos 5 heurísticas para o problema ACV. A primeira, H_1 , é baseada num algoritmo 2-aproximativo do problema RCC [12]. A segunda, H_2 , faz uso do algoritmo de subintervalo de soma máxima para encontrar “bons” padrões de alocação para cada símbolo e aplica um critério guloso para alocá-los. As heurísticas H_3 e H_4 , calculam uma solução da relaxação linear do problema e, baseadas nesta solução, definem os padrões mais promissores de atribuição de cada símbolo. Tais heurísticas diferem basicamente na forma como utilizam estes padrões promissores. A heurística H_3 realiza a atribuição dos símbolos baseando-se na ordenação lexicográfica dos padrões, enquanto que H_4 utiliza uma estratégia gulosa, atribuindo primeiramente os padrões que resultam num maior ganho da função objetivo do problema. A Heurística H_5 aplica conceitos de Relaxação Lagrangeana sobre a Formulação XY que foi apresentada no Capítulo 4. Utilizando o método de subgradientes encontra soluções da Relaxação Lagrangeana e as utiliza para encontrar soluções viáveis para o problema original. Para comparação de desempenho dessas 5 novas heurísticas, descrevemos também a heurística H_0 , proposta em [2].

6.1 A Heurística de Referência - H_0

Apresentamos a seguir a heurística proposta em [2] no contexto prático de aplicação do problema. Este algoritmo foi desenvolvido com o propósito de encontrar “boas” soluções e atender aos critérios de tempo real encontrados na prática, isto é, encontrar uma atribuição o mais próxima possível da ótima dentro de um limite de tempo aceitável no contexto da aplicação. Neste trabalho, tal heurística será utilizada como uma referência de comparação com as demais heurísticas propostas.

O Algoritmo 2 mostra esta heurística de referência, H_0 . Na primeira linha, encontramos

uma atribuição possivelmente não conexa A de ganho máximo, isto é, uma atribuição onde todo $A(j)$ corresponde ao símbolo i de maior ganho na posição j . O laço seguinte, na linha 3, garante que a solução encontrada será viável. Internamente a este laço, na linha 4, calculamos uma alocação virtual da solução encontrada. Uma alocação virtual de uma atribuição A é outra atribuição \hat{A} , obtida a partir de uma operação sobre A , denotada por $f(A)$, que, basicamente, “comprime” um intervalo contíguo de símbolos iguais em A numa única posição em \hat{A} . A Figura 6.1 ilustra esta operação. Definimos também a operação inversa $f^{-1}(\hat{A})$ que é utilizada para obter A a partir \hat{A} .

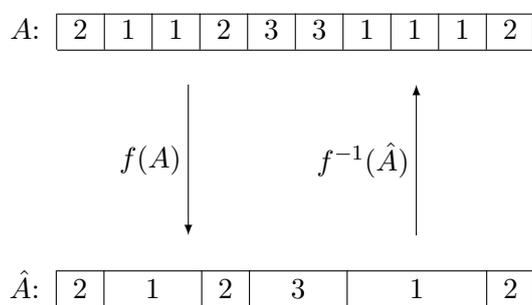


Figura 6.1: Ilustração das operações relacionadas à alocação virtual.

A linha 5 utiliza três regras de geração de alocação virtual aplicadas em \hat{A} para formar o conjunto \hat{C} . Tais regras são aplicadas exaustivamente, isto é, sempre que for possível são aplicadas de modo a formar o conjunto \hat{C} de maior cardinalidade possível. Abaixo, temos as regras de geração de alocação virtual:

- R_1 : se i é um símbolo contido na posição j de uma alocação virtual \hat{A} e sendo $j' > j + 1$ o menor índice onde i ocorre novamente em \hat{A} , então criamos uma nova alocação virtual A' , onde $A'(k) = \hat{A}(k)$, para $k \leq j$ ou $k \geq j'$, e $A'(k) = i$, para $j < k < j'$;
- R_2 : se i é um símbolo contido na posição $j > 1$ de uma alocação virtual \hat{A} de modo que i não ocorre novamente em nenhum $j' < j$, então criamos uma nova alocação A' , onde $A'(k) = \hat{A}(k)$, $\forall k \neq j - 1$ e $A'(k) = i$, para $k = j - 1$.
- R_3 : se i é um símbolo contido na posição $j < N$ de uma alocação virtual \hat{A} de modo que i não ocorre novamente em nenhum $j' > j$, então, criamos uma nova alocação A' , onde $A'(k) = \hat{A}(k)$, $\forall k \neq j + 1$ e $A'(k) = i$, para $k = j + 1$.

As figuras 6.2, 6.3 e 6.4 apresentam as regras descritas anteriormente.

$$\hat{A} \rightarrow \begin{array}{|c|c|c|c|c|} \hline & j & & j' & \\ \hline & i & & i & \\ \hline \end{array} \quad A' \rightarrow \begin{array}{|c|c|c|c|c|} \hline & j & & & j' \\ \hline & i & i & i & i \\ \hline \end{array}$$

Figura 6.2: Regra R_1 para geração de Alocações Virtuais.

$$\hat{A} \rightarrow \begin{array}{|c|c|c|c|c|c|} \hline & & j & & & \\ \hline & & i & & & \\ \hline \end{array} \quad A' \rightarrow \begin{array}{|c|c|c|c|c|c|} \hline & & & j & & \\ \hline & & i & i & & \\ \hline \end{array}$$

Figura 6.3: Regra R_2 para geração de Alocações Virtuais.

$$\hat{A} \rightarrow \begin{array}{|c|c|c|c|c|c|} \hline & & & j & & \\ \hline & & & i & & \\ \hline \end{array} \quad A' \rightarrow \begin{array}{|c|c|c|c|c|c|} \hline & & & & j & \\ \hline & & & i & i & \\ \hline \end{array}$$

Figura 6.4: Regra R_3 para geração de Alocações Virtuais.

Na linha 6, aplicamos a função inversa à de criação de alocação virtual em cada elemento do conjunto \hat{C} para formar o conjunto C . Em seguida, na linha 7, buscamos a atribuição de C que tem o maior valor de função objetivo. Ao final, com a saída do laço, temos a garantia de que a atribuição A é conexa e assim retornamos esta solução.

Algoritmo 2: Heurística H_0

Entrada: M, N e a função ganho ρ

Saída: Uma atribuição conexa

```

1 início
2   Calcule uma atribuição  $A$ , não necessariamente conexa, tal que
    $A(j) \in \arg \max_{i \in I} \{\rho_{ij}\}, \forall j \in J$ ;
3   enquanto  $A$  não é conexa faça
4      $\hat{A} \leftarrow f(A)$ , onde  $f$  é a função de criação de alocação virtual;
5     Tomando como parâmetro  $\hat{A}$ , gere o conjunto  $\hat{C}$  de alocações virtuais
     utilizando as regras  $R_1, R_2$  e  $R_3$ ;
6     Calcule  $C = \{c : c = f^{-1}(\hat{c}), \forall \hat{c} \in \hat{C}\}$ , onde  $f^{-1}$  é a função inversa à de criação
     de alocação virtual;
7     Faça  $A$  a atribuição de  $C$  que gera o maior valor de função objetivo;
8   fim
9   retorna  $A$ ;
10 fim

```

6.2 Uma Heurística Baseada em Multa - H_1

A Heurística H_1 é inspirada no algoritmo 2-aproximativo para o problema RCC [12]. A partir de uma atribuição possivelmente desconexa A de ganho máximo, a heurística H_1 procura obter uma solução conexa \hat{A} com $val(\hat{A})$ o mais próximo possível de $val(A)$. Primeiramente, definimos uma função de multa relacionada à atribuição inicial A , que toma como parâmetros um símbolo i e um intervalo $B \subseteq J$ de v .

A Figura 6.5 ilustra a ideia básica para a definição da função multa. Temos uma atribuição A , um intervalo B e queremos tornar B um intervalo que contém apenas símbolos i . Para isso, devemos substituir de B os símbolos diferentes de i por i . Além disso, como queremos obter uma atribuição conexa, devemos remover os símbolos i que estão fora de B . Dessa maneira, o objetivo da multa é indicar uma estimativa da máxima quantidade perdida em $val(A)$ ao tornar o intervalo B preenchido apenas pelo símbolo i .



Figura 6.5: Ilustração da Ideia Básica Para a Definição da Função de Multa.

Definimos a função multa da seguinte forma:

$$multa_A(i, B) = \rho_A(\overline{J_A(i)} \cap B) - \rho_i(\overline{J_A(i)} \cap B) + \rho_i(J_A(i) \cap \overline{B})$$

onde $J_A(i) = \{j \in J : A(j) = i\}$, $\overline{J_A(i)} = \{j \in J : A(j) \neq i\}$, $\rho_A(W) = \sum_{j \in W} \rho_{A(j)j}$, $\rho_i(W) = \sum_{j \in W} \rho_{ij}$ e $\overline{B} = J \setminus B$. Observe que $\rho_i(J_A(i) \cap \overline{B}) = \rho_i(J_A(i)) - \rho_i(J_A(i) \cap B)$. Então a multa se resume em

$$multa_A(i, B) = \rho_i(J_A(i)) - [\rho_i(B) - \rho_A(\overline{J_A(i)} \cap B)]$$

Portanto, considerando um símbolo $i \in I$, encontrar o intervalo B que minimiza $multa_A(i, B)$ significa encontrar um intervalo B que maximiza $\rho_i(B) - \rho_A(\overline{J_A(i)} \cap B)$. Podemos encontrar este intervalo em $O(N)$. De fato, considerando a sequência $S_i = \langle s_1^i, s_2^i, \dots, s_j^i, \dots, s_N^i \rangle$, associada ao símbolo i , onde $s_j^i = \rho_{ij} - \rho_{A(j)j}$, se $A(j) \neq i$, e $s_j^i = \rho_{ij}$, se $A(j) = i$, tal problema consiste em encontrar a subsequência de soma máxima de S_i (veja o Algoritmo 4) [18] [19].

O Algoritmo 3 apresenta a Heurística H_1 . Na linha 2 deste algoritmo, encontra-se uma atribuição possivelmente não conexa A de ganho máximo, isto é, uma atribuição onde todo

$A(j)$ corresponde ao símbolo i de maior ganho na posição j . No laço da linha 3, determina-se, para cada símbolo $i \in I$, o intervalo de multa mínima relacionada à atribuição calculada anteriormente. A função utilizada para este propósito é descrita pelo Algoritmo 4. Na linha 6, definimos o conjunto dos símbolos elegíveis para a alocação. O laço seguinte trata da alocação dos símbolos. Na linha 8, encontramos o símbolo k associado ao intervalo B_k de maior ganho dentre todos. A seguir (linha 9), fazemos a alocação de k nas posições definidas por B_k . Após isso, removemos o símbolo k do conjunto dos símbolos elegíveis para alocação (linha 10). O laço da linha 11 busca cada símbolo i cujo intervalo B_i é conflitante com B_k e o remove do conjunto de símbolos elegíveis para alocação (linha 13). Ao final, na linha 17, completamos a atribuição mantendo a conexidade.

Para analisar a complexidade da Heurística H_1 , vamos verificar o cenário de pior caso dela. Este cenário é aquele onde o laço enquanto, linha 7, executa o maior número de vezes, isto é, $\min\{M, N\}$. Note que, em uma execução da heurística, as instruções das linhas 8 e 11 executam o mesmo número de vezes e, com uma implementação cuidadosa, estas teriam a mesma complexidade: $O(|I'|)$ (para calcular $\rho_i(B_i)$ em tempo constante, podemos usar vetores v_i de soma acumulada com os ganhos de i , ou seja, cada posição j de v_i é dada por $v_i[j] = \sum_{j'=1}^j \rho_{ij}$). Se $M \leq N$, a complexidade temporal do laço enquanto, linha 7, será $O(M + (M - 1) + \dots + 1) = O(\frac{1+M}{2}M) = O(M^2)$. Caso $M > N$, será $O(M + (M - 1) + \dots + [M - (N - 1)]) = O(\frac{\{2M - [N - 1]\}N}{2})$. Observe que, neste caso, $O(\frac{\{2M - [N - 1]\}N}{2}) \subseteq O(M^2)$. Note que o laço da linha 3 requer $O(MN)$. Portanto, de modo geral, a complexidade da Heurística H_1 é $O(MN + M^2)$.

Algoritmo 3: Heurística H_1 **Entrada:** M, N e a função ganho ρ **Saída:** Uma atribuição conexa

```

1 início
2   Calcule uma atribuição  $A$ , não necessariamente conexa, tal que
    $A(j) \in \arg \max_{i \in I} \{\rho_{ij}\}, \forall j \in J$ ;
3   para cada  $i \in I$  faça
4      $B_i = \text{subseqSomaMaxima}(\langle s_1^i, s_2^i, \dots, s_j^i, \dots, s_N^i \rangle)$ , onde  $s_j^i = \rho_{ij} - \rho_{A(j)j}$ , se
      $A(j) \neq i$ , e  $s_j^i = \rho_{ij}$ , se  $A(j) = i$ ;
5   fim
6    $I' = \{1, 2, \dots, M\}$ ;
7   enquanto  $I' \neq \emptyset$  e há posições não alocadas faça
8     Seja  $k$ , tal que  $k \in \arg \max_{i \in I'} \{\rho_i(B_i)\}$ ;
9     Aloque  $k$  nas posições definidas por  $B_k$ ;
10     $I' = I' \setminus \{k\}$ ;
11    para todo  $i \in I'$  faça
12      se  $B_i \cap B_k \neq \emptyset$  então
13         $I' = I' \setminus \{i\}$ ;
14      fim
15    fim
16  fim
17  Se há posições não alocadas, atribua símbolos mantendo a conectividade;
18 fim

```

Algoritmo 4: Algoritmo subseqSomaMaxima**Entrada:** Uma sequência $\langle s_1, s_2, \dots, s_N \rangle$ **Saída:** A subsequência de soma máxima

```

1 início
2    $somaMax \leftarrow 0$ ;  $somaAtual \leftarrow 0$ ;
3    $j \leftarrow 1$ ;  $k \leftarrow 1$ ;  $p \leftarrow 1$ ;  $kTmp \leftarrow 1$ ;  $pTmp \leftarrow 1$ ;
4   enquanto  $j \leq N$  faça
5      $somaAtual \leftarrow somaAtual + s_j$ ;
6     se  $somaAtual \geq 0$  então
7        $pTmp \leftarrow j$ ;
8       se  $somaAtual > somaMax$  então
9          $somaMax \leftarrow somaAtual$ ;
10         $k \leftarrow kTmp$ ;  $p \leftarrow pTmp$ ;
11      fim
12    senão
13       $somaAtual \leftarrow 0$ ;
14       $kTmp \leftarrow j + 1$ ;  $pTmp \leftarrow j + 1$ ;
15    fim
16     $j \leftarrow j + 1$ ;
17  fim
18  retorna  $B_i = \{k, k + 1, \dots, p\}$ 
19 fim

```

6.3 A Heurística H_2

Nesta seção, apresentamos a Heurística H_2 , que, assim como H_1 , propõe-se a ter um baixo custo computacional. Para cada símbolo $i \in I$, definimos uma sequência $\langle s_1^i, s_2^i, \dots, s_N^i \rangle$, associada a i , e encontramos o subintervalo de soma máxima B_i desta sequência. Tomamos o símbolo $k \in \arg \max_{i \in I} \{\rho_i(B_i)\}$ e o alocamos nas posições definidas por B_k , onde $\rho_i(B_i) = \sum_{j \in B_i} \rho_{ij}$. Dessa maneira, perceba que a ideia central das heurísticas H_1 e H_2 são idênticas, isto é, as duas calculam subintervalos de soma máxima e aplicam um critério guloso para a alocação dos símbolos. Estas diferem basicamente no modo segundo o qual a sequência de cada símbolo é construída.

A sequência $\langle s_1^i, s_2^i, \dots, s_j^i, \dots, s_N^i \rangle$ utilizada por H_2 e associada ao símbolo i é definida da seguinte forma:

$$s_j^i = \rho_{ij} - \max_{i' \in I \setminus \{i\}} \{\rho_{i'j}\}, \forall j \in J$$

Pela definição da sequência anterior, observe que, ao considerar um certo símbolo $i \in I$ e uma certa posição $j \in J$, se o ganho de i for o maior em j dentre todos os símbolos, então s_j^i é positivo e, desta maneira, é interessante alocá-lo nesta posição, pois trata-se do símbolo de maior ganho. Caso i não tenha o maior ganho em j , então s_j^i é negativo e, portanto, a princípio não se tem vantagem em fazer a alocação de i em j , pois há um símbolo, diferente de i , que tem maior ganho em j . Perceba, então, que s_j^i representa uma referência de quão vantajoso é alocar o símbolo i na posição j e, sendo assim, ao considerar o subintervalo de soma máxima teremos encontrado um “bom” intervalo para alocar este símbolo.

Observe que a sequência de cada símbolo i considerada pela Heurística H_1 é diferente daquela utilizada por H_2 . A sequência definida pela primeira considera a aplicação de uma função de multa sobre uma atribuição inicial, possivelmente desconexa, de ganho máximo, isto é, a função multa define os símbolos da sequência tomando como parâmetro a atribuição inicial. A segunda heurística, por outro lado, considera apenas a diferença entre os ganhos de i com os ganhos dos demais símbolos. Assim, H_1 utiliza uma atribuição inicial como referência para calcular a sequência $\langle s_1^i, s_2^i, \dots, s_N^i \rangle$ de cada símbolo i e H_2 aplica um critério que relaciona diretamente o ganho do símbolo i com o ganho dos demais.

O Algoritmo 5 expressa em pseudo-código a Heurística H_2 . O conjunto I' inicializado na linha 2 define o conjunto dos símbolos elegíveis para alocação no laço da linha 3. A cada iteração deste laço, para cada i , consideramos a sequência $\langle s_1^i, s_2^i, \dots, s_N^i \rangle$ (linha 4). Na linha seguinte, linha 5, encontramos o símbolo k que possui o intervalo B_k de maior ganho

dentre todos em I' . Na linha 6, alocamos k nas posições definidas por B_k . A seguir, linha 7, removemos o símbolo recém alocado k do conjunto dos símbolos elegíveis para alocação I' . Após isto, na linha 8, ajustamos a função de ganho para garantir que, nas iterações subsequentes, não serão encontrados subintervalos B_i que possuem interseção com B_k . Ao final, se for necessário, completamos a atribuição (linha 10).

Observe que, a cada iteração, a Heurística H_2 redefine a sequência $\langle s_1^i, s_2^i, \dots, s_N^i \rangle$ de cada símbolo i e recalcula os subintervalos de soma máxima. Diferentemente de H_1 , que faz isso uma única vez. Por outro lado, é de se esperar que H_2 encontre melhores soluções, pois ajustamos a função ganho ao final de cada iteração para eliminar a influência dos símbolos já alocados no cálculo do subintervalo de soma máxima.

Vamos analisar o pior caso da Heurística H_2 para definir sua complexidade temporal. O cenário de pior caso corresponde àquele onde são executados o maior número de iterações do laço enquanto, na linha 3, ou seja, $\min\{M, N\}$ iterações. A cada iteração, na linha 4, definimos a sequência de cada símbolo; fazemos isso em $O(NM)$. Na linha 5, calculamos o subintervalo de soma máxima para cada símbolo e encontramos o símbolo k cuja alocação no seu respectivo subintervalo dá o maior ganho dentre todos. Isso requer também $O(NM)$. As linhas seguintes demandam tempo de $O(N)$. Desta maneira, a complexidade de H_2 é, de modo geral, $O(\min\{N, M\}NM)$.

Algoritmo 5: Heurística H_2

Entrada: M, N e a função ganho ρ

Saída: Uma atribuição conexa

```

1 início
2    $I' = \{1, 2, \dots, M\}$ ;
3   enquanto  $I' \neq \emptyset$  e há posições não alocadas faça
4      $\forall i \in I'$ , seja  $\langle s_1^i, s_2^i, \dots, s_j^i, \dots, s_N^i \rangle$ , tal que  $s_j^i = \rho_{ij} - \max_{i' \in I \setminus \{i\}} \{\rho_{i'j}\}, \forall j \in J$ ;
5     Seja  $k$ , tal que  $k \in \arg \max_{i \in I'} \{\rho_i(B_i)\}$ , onde  $B_i$  é o subintervalo de soma
      máxima de  $\langle s_1^i, s_2^i, \dots, s_N^i \rangle$ ;
6     Aloque  $k$  nas posições definidas pelo intervalo  $B_k$ ;
7      $I' = I' \setminus \{k\}$ ;
8     Faça  $\rho_{kj} = \infty, \forall j \in B_k$ , e  $\rho_{kj} = 0, \forall j \notin B_k$ ;
9   fim
10  Se há posições não alocadas, atribua símbolos mantendo a conectividade;
11 fim

```

6.4 A Heurística H_3

O Algoritmo 6 apresenta a Heurística H_3 . Inicialmente, na linha 2, calculamos uma solução ótima A_R da relaxação linear do problema de Formulação X. De posse desta solução, podemos definir o “padrão mais promissor” para cada $i \in I$, isto é, encontramos o padrão P_i que resulta no maior ganho de função objetivo dentre aqueles padrões que foram utilizados na atribuição deste símbolo segundo a solução A_R . Na linha 5, consideramos um vetor L formado pelos padrões P_1, P_2, \dots, P_M . Na linha seguinte fazemos a ordenação lexicográfica dos padrões, de modo que, para dois padrões P_i e P_j , P_i deve anteceder a P_j se, e somente se, o valor binário do padrão P_i é maior ou igual ao valor binário do padrão P_j . Por exemplo, o padrão $P_i = (0, 0, 1, 1)$ deve anteceder $P_j = (0, 0, 1, 0)$, pois seu valor binário é 3, enquanto o de P_j é 2. Nas linhas 8 e 9 são feitas as atribuições dos símbolos seguindo a ordenação definida. Por fim, na linha 10, finalizamos a atribuição, possivelmente alocando as últimas posições. Devido ao fato desta heurística resolver um PL, sua complexidade temporal corresponde à complexidade do método de resolução do PL. Por exemplo, aplicando o método dos pontos interiores, temos que a complexidade de H_3 será polinomial.

Algoritmo 6: Heurística H_3

Entrada: M, N e a função ganho ρ

Saída: Uma atribuição conexa

```

1 início
2   Calcule uma solução ótima  $A_R$  para a relaxação linear do problema;
3    $\forall i \in I$ , seja  $S_i \leftarrow \{p : x_{ip} > 0\}$ , onde  $x_{ip}$  é uma variável da solução  $A_R$ ;
4    $\forall i \in I$ , seja  $P_i$ , tal que  $P_i \in \arg \max_{p \in S_i} \{\sum_{j \in J(p)} \rho_{ij}\}$ ;
5    $L \leftarrow [P_1, P_2, \dots, P_M]$ ;
6   Faça a ordenação lexicográfica de  $L$ ;
7   Sejam  $a_i$  e  $b_i$  os extremos inicial e final, respectivamente, do padrão  $P_i$ ;
8   Seja  $P_e \leftarrow L[1]$ . Aloque  $e$  seguindo o padrão de extremos 1 e  $b_e$ ;
9   Para  $k = 2, \dots, M$ , seja  $P_{l'} = L[k - 1]$  e  $P_l = L[k]$ . Se  $b_l > b_{l'}$ , aloque  $l$  com o
   padrão de extremos  $b_{l'} + 1$  e  $b_l$ ;
10  Seja  $u$  o último símbolo alocado. Se  $b_u < N$ , aloque  $u$  nas  $N - b_u$  últimas posições;
11 fim
```

6.5 Uma Heurística Gulosa - H_4

O Algoritmo 7 apresenta a Heurística H_4 . Esta heurística calcula uma solução ótima da relaxação linear do problema de Formulação X na linha 2. Em seguida, com a solução encontrada, para cada símbolo $i \in I$ é definido o conjunto S_i , que consiste dos padrões que foram alocados para i na solução relaxada. O conjunto I' , inicializado na linha 4, representa

os candidatos a serem alocados a cada iteração do laço da linha 5. No interior deste laço, para cada $i \in I'$, encontra-se P_i que é o padrão que resulta no maior ganho de i dentre os padrões de S_i . Na linha 7, P_e representa o padrão de maior ganho, dentre todos aqueles P_i calculados na linha anterior. Em seguida, na linha 8, é alocado o símbolo e conforme o padrão P_e . Quando alocamos um símbolo segundo um padrão, removemos este símbolo do conjunto de candidatos a alocação I' , na linha 9, e removemos os padrões que possuem intersecção com este padrão recém alocado, na linha 13. Ao remover um padrão que possui intersecção, com P_e , adicionamos, possivelmente, outros 2 novos padrões que representam a “parte esquerda” e a “parte direita” do padrão que foi removido, desconsiderando a intersecção. Em outras palavras, substituímos p tal que $J(p) \cap J(e) \neq \emptyset$ por um ou dois padrões, conforme $J(p) \setminus J(P_e)$ defina um ou dois intervalos contínuos, respectivamente. Ao final, na linha 19, se houver posições não alocadas, completamos a atribuição cuidadosamente mantendo a conexidade. Como H_3 , esta heurística também resolve um PL, de modo que sua complexidade será dada pelo método de resolução do PL.

Algoritmo 7: Heurística H_4

Entrada: M, N e a função ganho ρ
Saída: Uma atribuição conexa

```

1 início
2   Calcule uma solução ótima  $A_R$  para a relaxação linear do problema;
3    $\forall i \in I$ , seja  $S_i \leftarrow \{p : x_{ip} > 0\}$ , onde  $x_{ip}$  é uma variável da solução  $A_R$ ;
4    $I' \leftarrow \{1, 2, \dots, M\}$ ;
5   enquanto  $I' \neq \emptyset$  e há posições não alocadas faça
6      $\forall i \in I'$ , seja  $P_i$ , tal que  $P_i \in \arg \max_{p \in S_i} \{\sum_{j \in J(p)} \rho_{ij}\}$ ;
7     Seja  $P_e$ , tal que  $e \in \arg \max_{i \in I'} \{\sum_{j \in J(p)} \rho_{ij}\}$ ;
8     Aloque  $e$  com o padrão  $P_e$ ;
9      $I' \leftarrow I' \setminus \{e\}$ ;
10    para todo  $i \in I'$  faça
11      para todo  $p \in S_i$  faça
12        se  $J(P_e) \cap J(p) \neq \emptyset$  então
13           $S_i \leftarrow S_i \setminus \{p\}$ ;
14          Acrescente a  $S_i$  os possíveis dois padrões associados  $J(p) \setminus J(P_e)$ ;
15        fim
16      fim
17    fim
18  fim
19  Se há posições não alocadas, atribua símbolos mantendo a conexidade;
20 fim
```

6.6 Uma Heurística Baseada em Relaxação Lagrangeana - H_5

Mostramos, a seguir, uma heurística baseada nos conceitos de Relaxação Lagrangeana [6]. Considere a formulação XY apresentada no Capítulo 4. Observe que as únicas restrições que relacionam as variáveis x e y são as restrições (4.8). Vamos relaxá-las e penalizá-las na função objetivo através de multiplicadores $\lambda_p \geq 0, \forall p \in Q$. Dessa maneira, a Relaxação Lagrangeana do modelo anterior será:

$$\max \sum_{i=1}^M \sum_{p=1}^P g_{ip} x_{ip} + \sum_{p=1}^P \lambda_p y_p$$

Sujeito a:

$$\sum_{p=1}^P x_{ip} \leq 1, \forall i \in I \quad (6.1)$$

$$y_p + y_{p'} \leq 1, \forall p, p' \in Q : J(p) \cap J(p') \neq \emptyset \quad (6.2)$$

$$x_{ip} \in \{0, 1\}, \forall i \in I \text{ e } \forall p \in Q \quad (6.3)$$

$$y_p \in \{0, 1\}, \forall p \in Q \quad (6.4)$$

onde $g_{ip} = \sum_{j=1}^N \rho_{ij} t_{jp} - \lambda_p$.

Observe que neste modelo relaxado as duas variáveis x e y são independentes e, desta forma, podemos dividi-lo em dois:

$$(P_1) \left\{ \begin{array}{l} \max \sum_{i=1}^M \sum_{p=1}^P g_{ip} x_{ip} \\ \text{Sujeito a:} \\ \sum_{p=1}^P x_{ip} \leq 1, \forall i \in I \\ x_{ip} \in \{0, 1\}, \forall i \in I \text{ e } \forall p \in Q \end{array} \right.$$

$$(P_2) \left\{ \begin{array}{l} \max \sum_{p=1}^P \lambda_p y_p \\ \text{Sujeito a:} \\ y_p + y_{p'} \leq 1, \forall p, p' \in Q : J(p) \cap J(p') \neq \emptyset \\ y_p \in \{0, 1\}, \forall p \in Q \end{array} \right.$$

Observe ainda que qualquer combinação de duas soluções ótimas para (P_1) e (P_2) corresponde a uma solução ótima para o problema da Relaxação Lagrangeana. Perceba que ambos os problemas são fáceis de resolver.

A resolução de (P_1) pode ser feita a partir da determinação de $p_i \in \arg \max_{p \in Q} g_{ip}, \forall i \in I$. De fato, uma solução ótima para (P_1) é $x = \sum \{e^{ip_i} : i \in I, g_{ip_i} > 0\}$, cujo valor é $\sum_{i \in I} \max\{0, g_{ip_i}\}$.

Para a resolução de (P_2) , consideramos o grafo $G = (V, E)$ tal que $V = Q$ e $E = \{uv : u, v \in V, J(u) \cap J(v) \neq \emptyset\}$. Uma solução ótima para (P_2) corresponde a encontrar um

conjunto independente de peso máximo em G , onde o peso do vértice p é $w_p = \lambda_p$. Observe que G é um grafo de intervalo, pois possui representação em intervalo. Tal representação é direta, devido ao fato de cada vértice estar associado a um padrão (que define um intervalo do vetor) e cada aresta considera justamente a interseção entre os intervalos. Sabemos que o problema de encontrar conjuntos independentes de peso máximo em grafos gerais é NP-Difícil, no entanto, para grafos de intervalos, o problema é fácil e se conhecem algoritmos polinomiais para resolvê-lo. Em 1992, Ju Yuan Hsiao mostrou um algoritmo que resolve este problema em tempo linear no número de vértices [20].

Seja $L(\lambda)$ o valor da relaxação Lagrangeana para um dado vetor de multiplicadores λ , ou seja, $L(\lambda)$ é a soma dos valores ótimos de (P_1) e (P_2) para λ . Sabemos que $L(\lambda)$ é um limite superior para o valor ótimo do problema original, $\forall \lambda \geq 0$. Sendo assim, desejamos encontrar $\lambda \geq 0$ tal que $L(\lambda)$ seja o menor possível. Este problema é conhecido como Problema Lagrangeano Dual. Para resolvê-lo, um dos métodos mais aplicados na prática, e que utilizamos na heurística proposta, é o método de subgradientes. Este método faz uso de um procedimento iterativo para definir λ . A partir da escolha de um multiplicador inicial λ^0 , o valor da componente i de λ na iteração k é calculada da seguinte forma:

$$\lambda_i^k = \max\{0, \lambda_i^{k-1} - t^{k-1} G(x^{k-1}, y^{k-1})\}, \text{ para } k > 0 \quad (6.5)$$

onde t^{k-1} é um escalar, que representa o tamanho de passo utilizado na iteração $k - 1$, e $G(x^{k-1}, y^{k-1})$ é um vetor subgradiente de $L(\lambda^{k-1})$, dados por:

$$G(x^k, y^k) = (g_1, \dots, g_p, \dots, g_P), \text{ com } g_p = y_p^k - \sum_{i=1}^M x_{ip}^k, \forall p \in Q \quad (6.6)$$

$$t^k = \frac{\epsilon \cdot (L(\lambda^{k-1}) - LB)}{\|G(x^{k-1}, y^{k-1})\|^2}, \quad (6.7)$$

sendo x^k, y^k as soluções ótimas dos problemas (P_1) e (P_2) , respectivamente, para λ^k .

O tamanho de passo t^k é ajustado em toda iteração k , e o parâmetro LB é um limite inferior. O valor ϵ é um escalar entre 0 e 2 e tem influência na velocidade de convergência do método.

O Algoritmo 8 expressa a Heurística H_5 . A cada iteração é calculada uma solução viável para a instância de entrada, a partir da solução da Relaxação Lagrangeana proposta anteriormente. Na linha 2, inicializamos o parâmetro λ . A seguir, a cada iteração i do laço, na linha 4, calculamos as soluções ótimas dos problemas (P_1) e (P_2) utilizando o λ corrente. Na linha 5, de posse das soluções x e y , obtemos uma atribuição viável, utilizando tais

soluções (explicaremos posteriormente como fazer isto). A solução x indica um conjunto de símbolos alocados, possivelmente, em padrões conflitantes. A solução y indica um conjunto de padrões não conflitantes. Após a determinação de uma solução viável, na linha 6, encontramos um limite inferior para ser usado no cálculo de t^i . O limite inferior pode ser obtido com o uso de uma heurística qualquer ou utilizando o valor da melhor solução obtida até o momento. Na linha 7, definimos o novo parâmetro λ da maneira que mostramos anteriormente e, finalmente, na linha 9, retornamos a melhor solução viável encontrada.

Algoritmo 8: Heurística H_5

Entrada: M, N e a função ganho ρ
Saída: Uma atribuição conexa

```

1 início
2    $\lambda \leftarrow \lambda^0$ ;
3   para  $i=1$  até  $\gamma$  faça
4     seja  $x$  e  $y$  soluções ótimas para  $(P_1)$  e  $(P_2)$  obtidas com o  $\lambda$  definido,
       respectivamente;
5      $A_i \leftarrow \text{calcularAtribuiçãoViável}(x, y)$ ;
6     obtenha  $LB$  e compute  $t^i$ ;
7     atualize o parâmetro  $\lambda$ ;
8   fim
9   retorna  $\arg \max_{A \in \{A_1, \dots, A_\gamma\}} \{val(A)\}$ ;
10 fim

```

Propomos o Algoritmo 9 como uma alternativa para a obtenção de uma solução viável para o problema. Este algoritmo utiliza a solução ótima y do problema (P_2) . Na linha 2, consideramos um grafo bipartido completo G , onde uma das partes é formada pelo conjunto de símbolos I e a outra é composta pelo conjunto de padrões escolhidos por y . A seguir, linha 3, utilizamos um algoritmo para calcular um emparelhamento de peso máximo nas arestas de G , tomando como peso da aresta ip o valor do ganho ao alocar i no padrão p .

Para esta finalidade, há diversos algoritmos conhecidos na literatura. Um dos mais populares e bastante utilizado na prática é o algoritmo Húngaro [10] [11]. Ao final, linha 4, o algoritmo retorna a atribuição que consiste em alocar cada símbolo i , contido em uma das arestas do emparelhamento, no respectivo padrão p ao qual foi emparelhado.

Podemos também imaginar outros algoritmos que utilizam a solução x , ao invés da solução y , para encontrar uma “boa” solução viável. Poderíamos, por exemplo, utilizar um critério guloso para alocar os símbolos de maior ganho nos padrões que foram estabelecidos na solução x .

A Heurística H_5 executa γ iterações, que se trata de um parâmetro da heurística. Ava-

liamos, então, a complexidade temporal da heurística para cada iteração. A complexidade de se calcular a solução ótima de (P_1) corresponde a $O(MP) = O(MN^2)$. Existem diversos algoritmos em tempo linear no número de vértices para calcular conjunto independente de peso máximo em grafos de intervalo. Desta forma, a complexidade de se obter uma solução ótima para (P_2) é $O(P) = O(N^2)$. Para calcular uma solução viável, propomos o Algoritmo 9, que utiliza o método Húngaro para cálculo do emparelhamento de peso máximo. Podemos encontrar facilmente implementações deste método de complexidade $O(N'^3)$, onde N' é o tamanho da maior parte do grafo bipartido completo. Como y seleciona apenas padrões não conflitantes, o tamanho da maior parte será no máximo $\max\{M, N\}$. Dessa maneira, a complexidade para encontrar esta solução será $O([M + N]^3)$. As demais operações da heurística requerem tempo constante, portanto a complexidade de H_5 é $O([M + N]^3)$.

Algoritmo 9: calcularAtribuicaoViavel

Entrada: A solução y do problema (P_2)

Saída: Uma atribuição conexa

1 **início**

2 Seja $G = (V_1 \cup V_2, V_1 \times V_2)$ um grafo bipartido completo tal que $V_1 = I$,
 $V_2 = \{p : y_p = 1\}$;

3 Calcule um emparelhamento $M = \{i_1p_1, \dots, i_kp_k\}$ de peso máximo nas arestas de
 G , onde o peso de $ip \in V_1 \times V_2$ é dado por $w_{ip} = \sum_{j=1}^N \rho_{ij} \cdot t_{jp}$;

4 **retorna** $\sum_{k'=1}^k e^{i_{k'}p_{k'}}$;

5 **fim**

Capítulo 7

Resultados Computacionais

Os resultados computacionais apresentados neste capítulo foram obtidos num ambiente computacional Ubuntu 14.04 LTS de 64 bits com hardware equipado com processador intel® Core™ i7 com *clock* de 3.4 GHz e 16 GB de memória DDR3. Os algoritmos foram implementados na linguagem Java versão 1.8 e foi utilizado o CPLEX 12.6.1 como ferramenta para resolução dos problemas de programação linear e inteira.

7.1 Instâncias

Neste trabalho utilizamos dois tipos de classes de instâncias. A primeira, refere-se aquelas obtidas através do simulador desenvolvido em [2]. Este simulador permite a especificação de um cenário de comunicação celular através da definição de parâmetros relacionados ao sistema como, por exemplo, o número de usuários, taxas de transmissão e serviços do sistema. Baseado nestes parâmetros, o software faz uma simulação de um ambiente celular moderno com as características definidas e, através de modelos matemáticos, pode-se calcular as taxas de transmissão dos usuários que representam a função ganho ρ de ACV. Chamaremos genericamente as instâncias desta classe de Instâncias Simuladas.

A segunda classe de instâncias são aquelas obtidas aleatoriamente com valores de ganhos uniformemente distribuídos no intervalo $[0, 200]$. Iremos nos referir a estas instâncias como Instâncias Aleatórias.

Usando as instâncias desejamos verificar o desempenho das heurísticas propostas em termos da qualidade de solução e de tempo de computação, tanto para situações mais próximas da realidade (através de instâncias simuladas) como também estudá-las num contexto mais amplo (através de instâncias aleatórias).

Para cada classe, dividimos as instâncias em 3 grandes grupos: $N = 20$, $N = 60$ e

$N = 100$. Para um dado N fixo, definimos 6 valores para M : $N/5$, $2N/5$, $3N/5$, $4N/5$, $5N/5$ e $6N/5$. Para cada par (N, M) foram geradas 30 instâncias simuladas e 30 instâncias aleatórias. Assim, cada grupo com N fixo tem um total de 180 instâncias de cada classe.

7.2 Apresentação e Avaliação de Resultados

Neste capítulo, apresentamos os resultados através de gráficos, para uma visualização mais rápida. No Apêndice A, mostramos em tabelas os dados base para esses gráficos, que podem ser consultados para uma verificação mais precisa de seus valores. Desejamos avaliar as heurísticas em termos da qualidade de solução gerada e do tempo de computação demandada. Para isso, definimos a Razão de Aproximação (RA) de uma solução como sendo a razão (Valor da Solução / Valor Ótimo). As unidades de tempo são sempre especificadas em milissegundos.

Para cada par (N, M) e para cada classe de instâncias, construímos o Intervalo de Confiança (IC) de 95% de confiabilidade para os resultados de razão de aproximação de cada heurística. Utilizaremos estes intervalos como um critério de avaliação e comparação da qualidade de solução das heurísticas.

7.3 Resultados Para $N = 20$

Os valores de M para $N = 20$ são: 4, 8, 12, 16, 20 e 24. A Figura 7.1 apresenta os resultados referentes a Razão de Aproximação. Na região à esquerda, temos os resultados obtidos para as instâncias simuladas. Na região direita, estão os resultados para as instâncias aleatórias. As figuras 7.2 e 7.3 mostram os resultados de tempo de computação para as instâncias simuladas e aleatórias, respectivamente. No lado direito destas figuras temos o gráfico restrito aos resultados de tempo somente para as heurísticas H_0 , H_1 e H_2 . Resolvemos destacar estas três por conta de não haver uma percepção exata do comportamento delas no gráfico geral (à esquerda da mesma figura), por se tratar das heurísticas com menor tempo de computação dentre todas.

Observamos que, para as instâncias simuladas, todas as heurísticas retornam soluções de boa qualidade, sempre acima de 96% do valor ótimo. Além disso, notamos que as heurísticas H_3 e H_4 são aquelas que apresentaram as melhores soluções. Em contrapartida, são elas que tiveram o maior tempo de computação. As heurísticas H_0 e H_1 são as mais rápidas, dentre todas, e H_1 juntamente com H_2 são as que apresentaram as soluções de menor qualidade. No entanto, esta diferença não é tão significativa, em torno de 2 ou 3% das demais.

Para as instâncias aleatórias, a qualidade das soluções da heurística de referência, H_0 , passa a ser bastante afetada, piorando com o aumento de M , chegando a obter uma razão de aproximação média de 75% para as instâncias com $M = 20$. O comportamento das demais se assemelha aos obtidos para as instâncias simuladas, com razão de aproximação sempre maior que 90% do valor ótimo. O mesmo ocorre para o tempo de computação, onde notamos curvas de tempo bem semelhantes às curvas observadas para as instâncias simuladas.

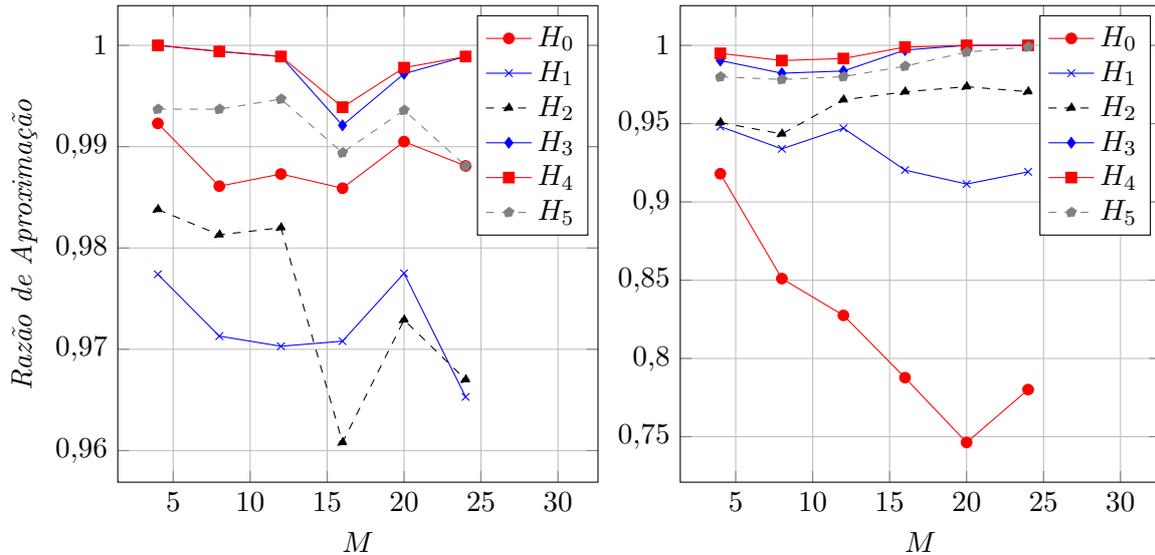


Figura 7.1: Gráfico da Razão de Aproximação para $N = 20$ (Instâncias Simuladas à Esquerda, Instâncias Aleatórias à Direita).

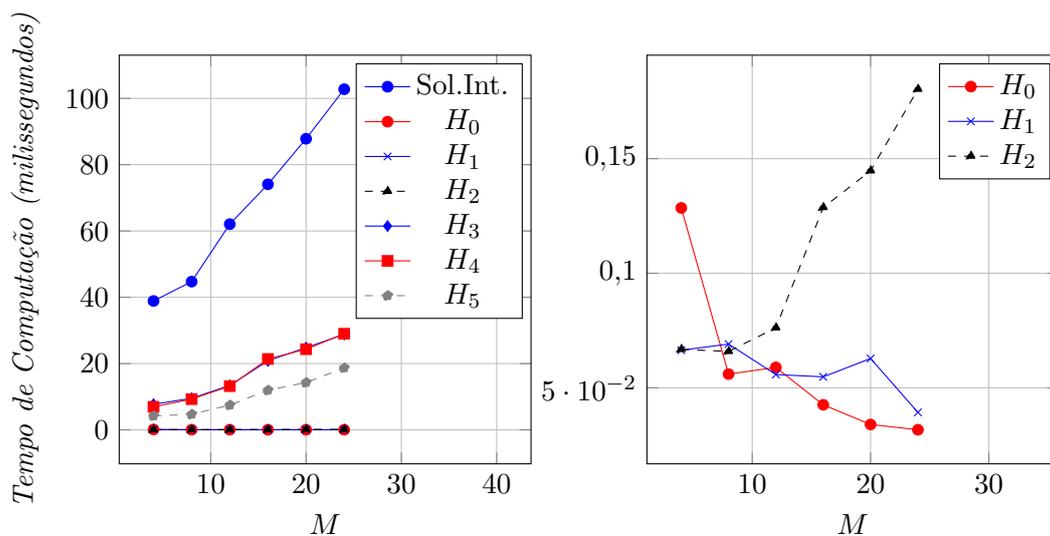


Figura 7.2: Gráficos do Tempo de Computação para $N = 20$ (Instâncias Simuladas).

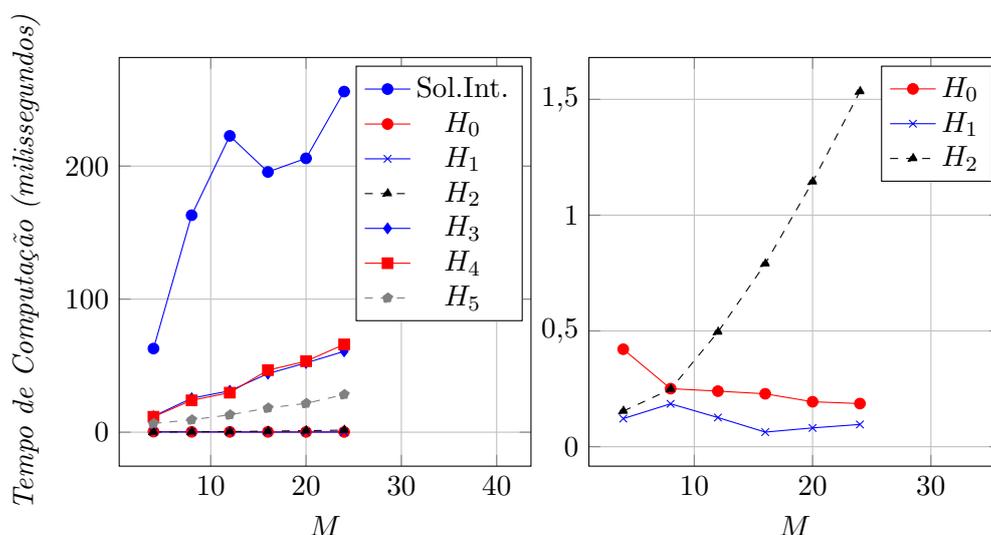


Figura 7.3: Gráficos do Tempo de Computação para $N = 20$ (Instâncias Aleatórias).

As tabelas 7.1 e 7.2 mostram os Intervalos de Confiança (IC) para os resultados de razão de aproximação das classes de instâncias simuladas e aleatórias, respectivamente. Considerando uma classe de instâncias específica, são definidos um IC para cada par (N, M) e para cada heurística. Estes intervalos são definidos com uma confiabilidade de 95%. Isto significa que, se o experimento for reproduzido 100 vezes com amostras similares (de mesma classe e valores de M e N), é de se esperar que a média da razão de aproximação obtida para cada heurística esteja dentro do respectivo IC em 95 destes 100 experimentos, ou seja, cada experimento tem 95% de chance de apresentar média da razão de aproximação dentro do IC. Desta maneira, desejamos heurísticas que apresentem um IC curto com extremo superior próximo de 1.

Notamos que, para as instâncias simuladas, as heurísticas H_4 e H_3 foram as que apresentaram os ICs mais curtos. Além disso, tais intervalos têm extremo superior maior ou igual a 1. Destacamos também a heurística H_5 que apresenta ICs curtos com extremo direito bem próximo de 1. Dentre todas, a heurística H_2 foi a que apresentou o maior número de ICs mais longos.

Para as instâncias aleatórias, as heurísticas H_4 e H_3 continuam a definir os ICs mais curtos e dentre as demais se destaca H_5 . Observamos que, para este conjunto de instâncias, a heurística H_0 apresenta os ICs mais longos crescentes com o valor de M , diferente do que ocorre para as instâncias simuladas, onde seus ICs são consideravelmente mais curtos.

Tabela 7.1: Intervalos de Confiança Para a Razão de Aproximação (Instâncias Simuladas com $N = 20$).

M	IC (H_0)	IC (H_1)	IC (H_2)	IC (H_3)	IC (H_4)	IC (H_5)
4	$0,992 \pm 0,007$	$0,977 \pm 0,013$	$0,984 \pm 0,017$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,994 \pm 0,005$
8	$0,986 \pm 0,008$	$0,971 \pm 0,014$	$0,981 \pm 0,012$	$0,999 \pm 0,001$	$0,999 \pm 0,001$	$0,994 \pm 0,004$
12	$0,987 \pm 0,008$	$0,970 \pm 0,021$	$0,982 \pm 0,011$	$0,999 \pm 0,002$	$0,999 \pm 0,002$	$0,995 \pm 0,005$
16	$0,986 \pm 0,009$	$0,971 \pm 0,017$	$0,961 \pm 0,024$	$0,992 \pm 0,008$	$0,994 \pm 0,008$	$0,989 \pm 0,008$
20	$0,991 \pm 0,007$	$0,977 \pm 0,013$	$0,973 \pm 0,018$	$0,997 \pm 0,005$	$0,998 \pm 0,004$	$0,994 \pm 0,005$
24	$0,988 \pm 0,008$	$0,965 \pm 0,016$	$0,967 \pm 0,018$	$0,999 \pm 0,002$	$0,999 \pm 0,002$	$0,988 \pm 0,007$

Tabela 7.2: Intervalos de Confiança Para a Razão de Aproximação (Instâncias Aleatórias com $N = 20$).

M	IC (H_0)	IC (H_1)	IC (H_2)	IC (H_3)	IC (H_4)	IC (H_5)
4	$0,918 \pm 0,024$	$0,948 \pm 0,023$	$0,951 \pm 0,016$	$0,990 \pm 0,011$	$0,995 \pm 0,005$	$0,980 \pm 0,008$
8	$0,851 \pm 0,028$	$0,934 \pm 0,015$	$0,943 \pm 0,014$	$0,982 \pm 0,014$	$0,990 \pm 0,009$	$0,978 \pm 0,005$
12	$0,828 \pm 0,038$	$0,947 \pm 0,010$	$0,965 \pm 0,010$	$0,984 \pm 0,009$	$0,992 \pm 0,006$	$0,980 \pm 0,005$
16	$0,788 \pm 0,035$	$0,920 \pm 0,014$	$0,970 \pm 0,008$	$0,997 \pm 0,004$	$0,999 \pm 0,001$	$0,987 \pm 0,004$
20	$0,746 \pm 0,039$	$0,911 \pm 0,014$	$0,974 \pm 0,008$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,996 \pm 0,002$
24	$0,780 \pm 0,045$	$0,919 \pm 0,020$	$0,970 \pm 0,010$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,999 \pm 0,001$

As tabelas 7.3 e 7.4, resumizam os resultados obtidos para as 180 instâncias simuladas e as 180 instâncias aleatórias, respectivamente, para $N = 20$. O tempo médio de computação da solução ótima para as instâncias simuladas foi de 68,387 milissegundos, que é 3,9 vezes maior que o tempo médio da heurística mais lenta, H_3 , e 1.179 vezes maior que o tempo médio da heurística mais rápida, H_1 . Notamos que houve instâncias em que H_1 e H_2 obtiveram soluções com razão de aproximação bem próximas de 75%.

Para as instâncias aleatórias, o tempo de computação da solução ótimo foi em média 184,34 milissegundos. Este valor é 4,78 vezes maior que o maior tempo médio, de H_4 , e 1.645 vezes maior que o menor tempo médio, de H_1 . Observamos que houve instâncias aleatórias onde a heurística H_0 obteve soluções com razão de aproximação próxima de 46% e algumas fizeram H_1 obter solução com 72% de proximidade da solução ótima.

Tabela 7.3: Resultados Médios Gerais Para $N = 20$ (Instâncias Simuladas).

$N = 20$	H_0	H_1	H_2	H_3	H_4	H_5
Razão de Aprox. Média	0,988	0,972	0,975	0,998	0,998	0,992
Razão de Aprox. Min.	0,896	0,750	0,757	0,887	0,887	0,907
Razão de Aprox. Max.	1,000	1,000	1,000	1,000	1,000	1,000
Tempo Médio (ms)	0,059	0,058	0,110	17,516	17,359	10,222
Tempo Min. (ms)	0,009	0,019	0,020	4,266	4,547	2,244
Tempo Max. (ms)	0,686	0,200	0,649	36,424	33,762	23,416

Tabela 7.4: Resultados Médios Gerais Para $N = 20$ (Instâncias Aleatórias).

$N = 20$	H_0	H_1	H_2	H_3	H_4	H_5
Razão de Aprox. Média	0,818	0,930	0,962	0,992	0,996	0,987
Razão de Aprox. Min.	0,469	0,722	0,856	0,862	0,915	0,921
Razão de Aprox. Max.	1,000	1,000	1,000	1,000	1,000	1,000
Tempo Médio (ms)	0,254	0,112	0,727	37,578	38,529	16,173
Tempo Min. (ms)	0,084	0,046	0,055	6,592	6,326	3,051
Tempo Max. (ms)	1,195	0,458	2,568	95,893	89,805	38,171

7.4 Resultados Para $N = 60$

Para as instâncias com $N = 60$, apresentamos os resultados obtidos da mesma forma que foram apresentados para $N = 20$. Os gráficos da Figura 7.4 mostram as razões de aproximação médias para as instâncias simuladas, à esquerda, e as instâncias aleatórias, à direita. Nas figuras 7.5 e 7.6 temos os resultados referentes ao tempo médio de computação.

De imediato notamos resultados bem similares aos obtidos anteriormente. Para as instâncias simuladas, as heurísticas apresentaram boas razões de aproximação média, sempre no mínimo 94% do valor ótimo. Notamos também que H_3 e H_4 continuam a apresentar as soluções mais próximas das soluções ótimas e requerem, em média, uma quantidade de tempo consideravelmente maior que as outras heurísticas. De todas as heurísticas a que apresentou soluções de menor qualidade foi a H_2 . Quanto ao tempo de computação, notamos que o comportamento das heurísticas foi o mesmo ocorrido anteriormente, onde as heurísticas H_0 , H_1 e H_2 apresentam os menores tempos e H_1 se mostra como a mais rápida de todas.

Considerando o conjunto de instâncias aleatórias, observamos, mais uma vez, que todas as heurísticas propostas superam H_0 em qualidade de solução. As heurísticas H_3 e H_4 se mostram aquelas com melhores soluções e maiores tempos de computação. Podemos destacar a heurística H_2 , pelo seu tempo de computação baixo e por superar as heurísticas H_0 e H_1 para a maior parte das instâncias, sempre apresentando uma razão de aproximação média acima de 90%. Dentre todas, a heurística H_1 foi aquela de menor tempo de computação médio.

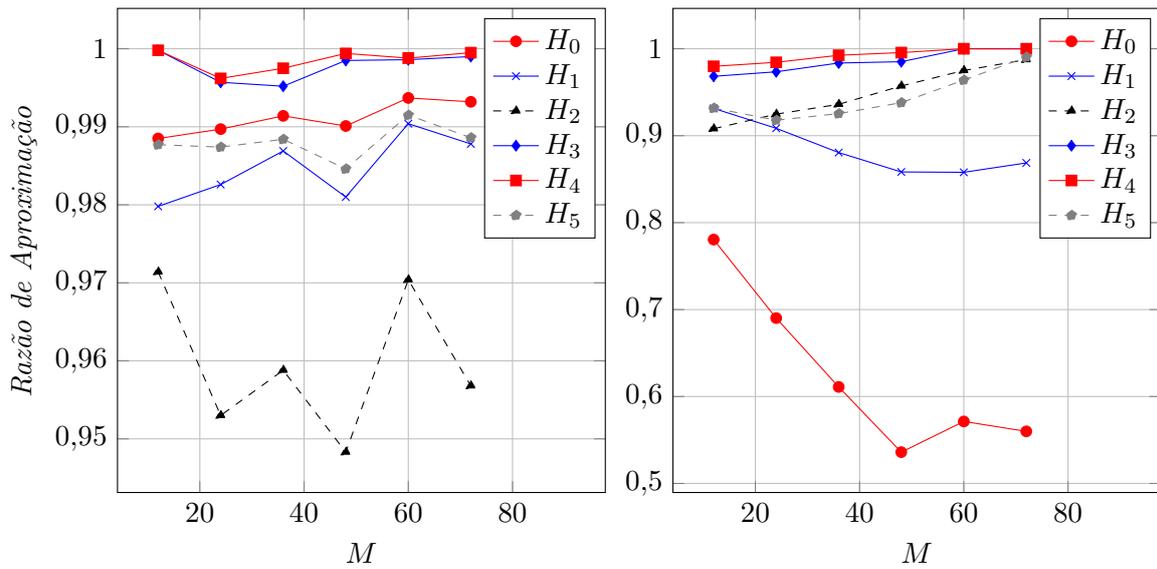


Figura 7.4: Gráfico da Razão de Aproximação para $N = 60$ (Instâncias Simuladas à Esquerda, Instâncias Aleatórias à Direita).

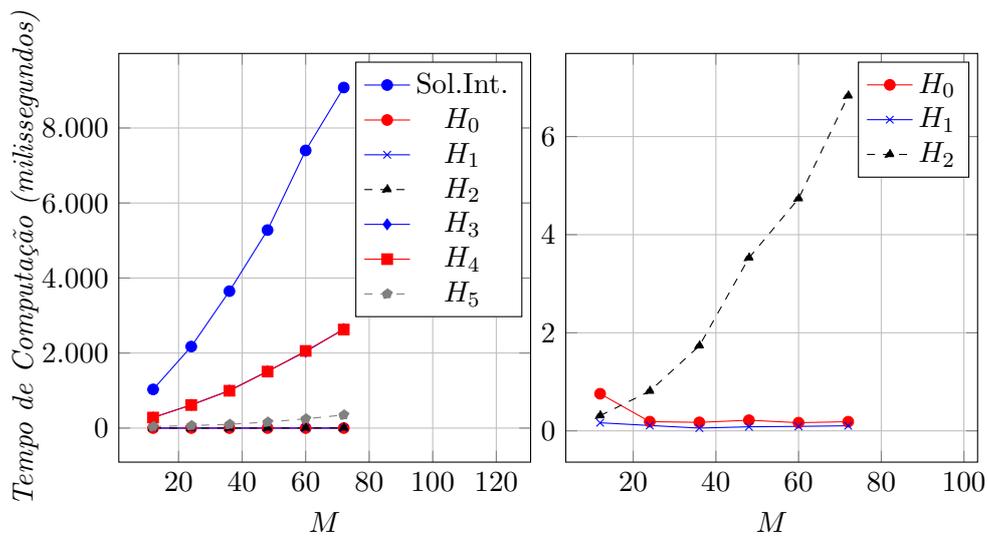


Figura 7.5: Gráficos do Tempo de Computação para $N = 60$ (Instâncias Simuladas).

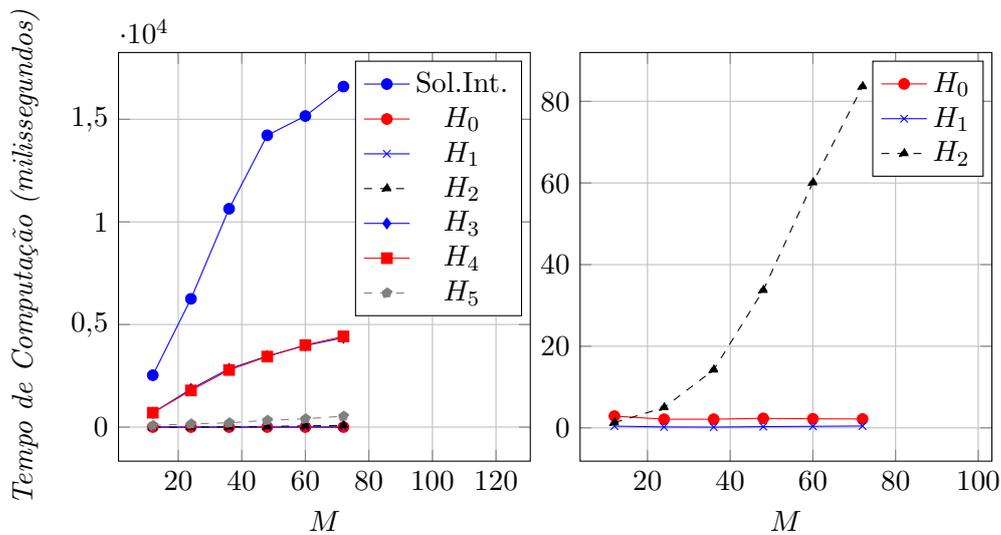


Figura 7.6: Gráficos do Tempo de Computação para $N = 60$ (Instâncias Aleatórias).

As tabelas 7.5 e 7.6 mostram os Intervalos de Confiança (IC) para os resultados de razão de aproximação das classes de instâncias simuladas e aleatórias com $N = 60$, respectivamente. Observamos resultados similares, com algumas pequenas diferenças, aos de $N = 20$. As heurísticas H_4 e H_3 continuam a apresentar os menores ICs que tem extremos superiores bem próximos de 1 para os dois tipos de classes de instâncias. Ao desconsiderar H_4 e H_3 , a heurística H_0 apresenta os menores ICs, para as instâncias simuladas, e a heurística H_5 os mais curtos, para as aleatórias. Notamos também que H_2 apresenta os intervalos mais longos para as instâncias simuladas e H_0 os mais longos para as aleatórias.

Tabela 7.5: Intervalos de Confiança Para a Razão de Aproximação (Instâncias Simuladas com $N = 60$).

M	IC (H_0)	IC (H_1)	IC (H_2)	IC (H_3)	IC (H_4)	IC (H_5)
12	$0,989 \pm 0,008$	$0,980 \pm 0,013$	$0,971 \pm 0,022$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,988 \pm 0,007$
24	$0,990 \pm 0,008$	$0,983 \pm 0,011$	$0,953 \pm 0,035$	$0,996 \pm 0,008$	$0,996 \pm 0,007$	$0,987 \pm 0,008$
36	$0,991 \pm 0,005$	$0,987 \pm 0,006$	$0,959 \pm 0,029$	$0,995 \pm 0,006$	$0,998 \pm 0,002$	$0,988 \pm 0,005$
48	$0,990 \pm 0,005$	$0,981 \pm 0,009$	$0,948 \pm 0,029$	$0,999 \pm 0,002$	$0,999 \pm 0,001$	$0,985 \pm 0,006$
60	$0,994 \pm 0,003$	$0,990 \pm 0,005$	$0,970 \pm 0,020$	$0,999 \pm 0,002$	$0,999 \pm 0,002$	$0,992 \pm 0,004$
72	$0,993 \pm 0,004$	$0,988 \pm 0,007$	$0,957 \pm 0,025$	$0,999 \pm 0,001$	$1,000 \pm 0,000$	$0,989 \pm 0,007$

Tabela 7.6: Intervalos de Confiança Para a Razão de Aproximação (Instâncias Aleatórias com $N = 60$).

M	IC (H_0)	IC (H_1)	IC (H_2)	IC (H_3)	IC (H_4)	IC (H_5)
12	$0,780 \pm 0,026$	$0,931 \pm 0,011$	$0,908 \pm 0,015$	$0,968 \pm 0,019$	$0,980 \pm 0,011$	$0,932 \pm 0,011$
24	$0,690 \pm 0,034$	$0,909 \pm 0,010$	$0,925 \pm 0,010$	$0,974 \pm 0,013$	$0,984 \pm 0,008$	$0,918 \pm 0,007$
36	$0,611 \pm 0,028$	$0,881 \pm 0,009$	$0,936 \pm 0,009$	$0,984 \pm 0,010$	$0,993 \pm 0,004$	$0,925 \pm 0,006$
48	$0,536 \pm 0,036$	$0,858 \pm 0,009$	$0,957 \pm 0,006$	$0,985 \pm 0,011$	$0,996 \pm 0,003$	$0,938 \pm 0,005$
60	$0,571 \pm 0,029$	$0,858 \pm 0,012$	$0,975 \pm 0,005$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,964 \pm 0,004$
72	$0,560 \pm 0,039$	$0,869 \pm 0,008$	$0,987 \pm 0,003$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,990 \pm 0,002$

As tabelas 7.7 e 7.8 resumem os resultados obtidos para as 180 instâncias simuladas e as 180 instâncias aleatórias, respectivamente, com $N = 60$. O tempo médio de computação da solução ótima para o conjunto de instâncias simuladas foi de 4.768 milissegundos. Este valor é 3,5 vezes maior que o tempo médio da heurística de maior tempo, H_3 , e 47.207 vezes maior que o da heurística de menor tempo, H_1 . Observamos que, para algumas instâncias, H_2 obteve soluções cuja razão de aproximação foi em torno de 64%.

O tempo de computação médio da solução ótima para as instâncias aleatórias com $N = 60$ foi de 10.898 milissegundos, que é 3,8 vezes maior que o tempo médio de computação da heurística de maior tempo, H_3 , e 32.924 vezes maior que o tempo médio de computação da de menor tempo, H_1 . Podemos notar que a razão de aproximação média para a heurística de referência H_0 foi 62,5%, que é consideravelmente menor que a das demais heurísticas, e para algumas instâncias esta heurística obteve soluções cuja aproximação é 33,8% da ótima.

Tabela 7.7: Resultados Médios Gerais para $N = 60$ (Instâncias Simuladas).

$N = 60$	H_0	H_1	H_2	H_3	H_4	H_5
Razão de Aprox. Média	0,991	0,985	0,960	0,998	0,999	0,988
Razão de Aprox. Min.	0,900	0,854	0,641	0,890	0,901	0,921
Razão de Aprox. Máx.	1,000	1,000	1,000	1,000	1,000	1,000
Tempo Médio (ms)	0,281	0,101	2,991	1349,018	1346,800	160,786
Tempo Min. (ms)	0,025	0,029	0,040	253,732	253,777	34,072
Tempo Max. (ms)	4,880	0,434	26,548	3253,351	3659,490	480,116

Tabela 7.8: Resultados Médios Gerais para $N = 60$ (Instâncias Aleatórias).

$N = 60$	H_0	H_1	H_2	H_3	H_4	H_5
Razão de Aprox. Média	0,625	0,884	0,948	0,985	0,992	0,945
Razão de Aprox. Min.	0,338	0,789	0,808	0,834	0,903	0,874
Razão de Aprox. Máx.	0,917	0,998	0,999	1,000	1,000	0,999
Tempo Médio (ms)	2,316	0,331	32,980	2863,254	2854,650	286,017
Tempo Min. (ms)	0,789	0,069	0,642	454,779	440,368	57,302
Tempo Max. (ms)	24,637	1,318	139,187	5864,211	5738,542	858,880

7.5 Resultados Para $N = 100$

Os gráficos da Figura 7.7 mostram as curvas que descrevem as razões de aproximação médias para as instâncias simuladas, à esquerda, e para as instâncias aleatórias, à direita. Nas figuras 7.8 e 7.9 temos os resultados referentes ao tempo médio de computação. Os valores que M assume neste conjunto de instâncias são: 20, 40, 60, 80, 100 e 120.

Os resultados apresentados nos gráficos se assemelham aos resultados anteriores, tanto em qualidade de solução quanto em tempo de computação. Dentre todas as heurísticas, H_3 e H_4 continuam a apresentar as melhores soluções e a consumir o maior tempo. Percebemos que, para as instâncias simuladas, todas as heurísticas apresentam bons resultados médios de aproximação, sempre acima de 92% do valor ótimo, e a heurística H_2 , mais um vez, se mostra aquela de menor qualidade de solução. Os resultados para as instâncias aleatórias também são bem similares. Todas as heurísticas propostas apresentam soluções melhores que as soluções de H_0 . No que diz respeito ao tempo de computação, observamos que, independente do tipo de classe de instância, a heurística H_1 é mais rápida de todas.

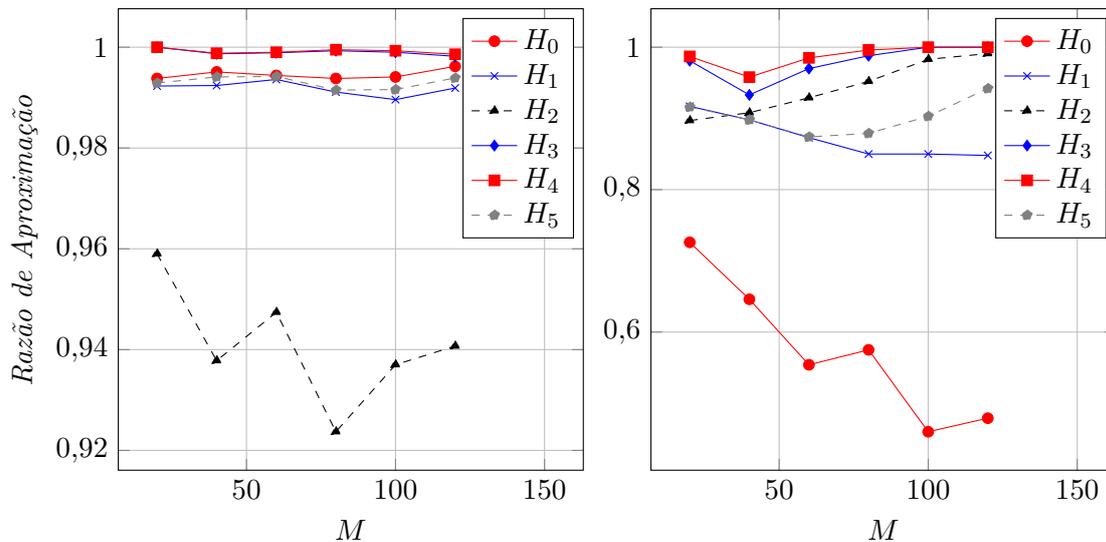


Figura 7.7: Gráficos da Razão de Aproximação para $N = 100$ (Instâncias Simuladas à Esquerda, Instâncias Aleatórias à Direita).

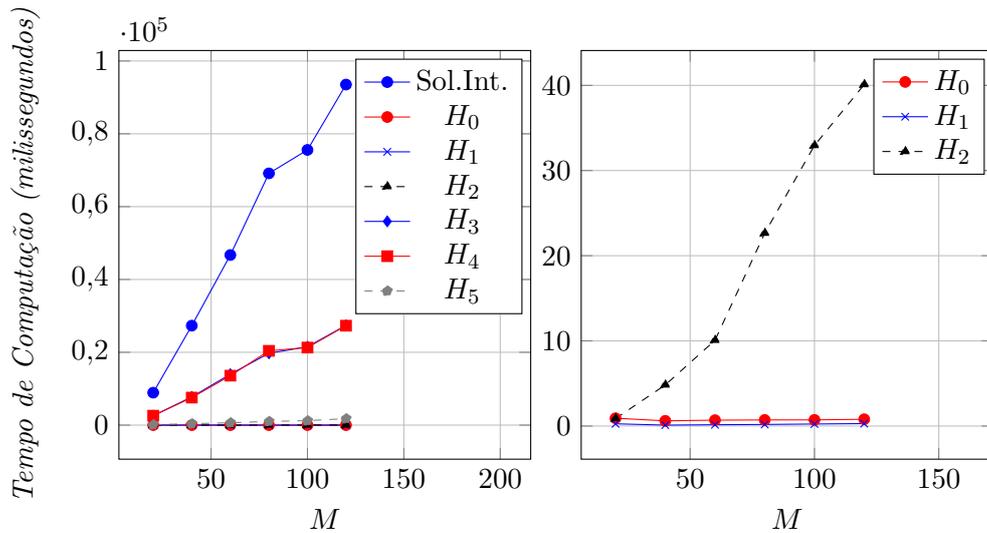


Figura 7.8: Gráficos do Tempo de Computação para $N = 100$ (Instâncias Simuladas).

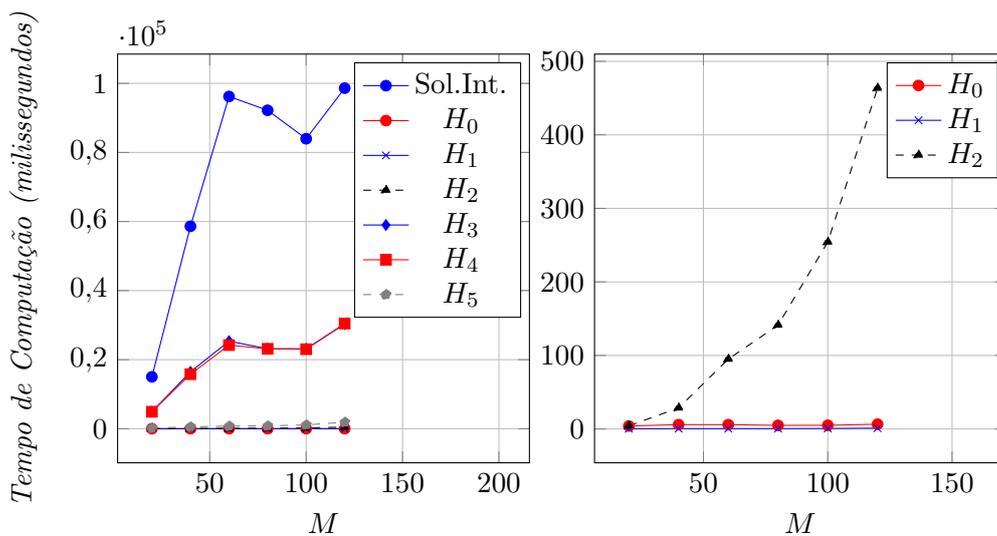


Figura 7.9: Gráficos do Tempo de Computação para $N = 100$ (Instâncias Aleatórias).

As tabelas 7.9 e 7.10 mostram os Intervalos de Confiança (IC) para os resultados de razão de aproximação das classes de instâncias simuladas e aleatórias com $N = 100$, respectivamente. Os resultados obtidos, mais uma vez, se assemelham aos anteriores. As heurísticas H_4 e H_3 continuam a apresentar os menores ICs, com extremos superiores próximos de 1, para as duas classes de instâncias. Ao desconsiderar estas duas, a heurística H_0 apresenta os menores ICs para as instâncias simuladas e H_2 os menores para as instâncias aleatórias. Curiosamente H_2 apresenta os intervalos mais longos para as instâncias simuladas e H_0 os mais longos para as aleatórias.

Tabela 7.9: Intervalos de Confiança Para a Razão de Aproximação (Instâncias Simuladas com $N = 100$).

M	IC (H_0)	IC (H_1)	IC (H_2)	IC (H_3)	IC (H_4)	IC (H_5)
20	$0,994 \pm 0,006$	$0,992 \pm 0,005$	$0,959 \pm 0,030$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,993 \pm 0,005$
40	$0,995 \pm 0,003$	$0,992 \pm 0,004$	$0,938 \pm 0,043$	$0,999 \pm 0,001$	$0,999 \pm 0,001$	$0,994 \pm 0,003$
60	$0,994 \pm 0,003$	$0,994 \pm 0,003$	$0,947 \pm 0,030$	$0,999 \pm 0,001$	$0,999 \pm 0,001$	$0,994 \pm 0,003$
80	$0,994 \pm 0,003$	$0,991 \pm 0,004$	$0,924 \pm 0,039$	$0,999 \pm 0,001$	$1,000 \pm 0,001$	$0,992 \pm 0,004$
100	$0,994 \pm 0,003$	$0,990 \pm 0,005$	$0,937 \pm 0,033$	$0,999 \pm 0,001$	$0,999 \pm 0,001$	$0,992 \pm 0,004$
120	$0,996 \pm 0,002$	$0,992 \pm 0,005$	$0,941 \pm 0,037$	$0,998 \pm 0,002$	$0,999 \pm 0,001$	$0,994 \pm 0,003$

Tabela 7.10: Intervalos de Confiança Para a Razão de Aproximação (Instâncias Aleatórias com $N = 100$).

M	IC (H_0)	IC (H_1)	IC (H_2)	IC (H_3)	IC (H_4)	IC (H_5)
20	$0,716 \pm 0,027$	$0,910 \pm 0,012$	$0,899 \pm 0,011$	$0,977 \pm 0,016$	$0,986 \pm 0,010$	$0,910 \pm 0,012$
40	$0,653 \pm 0,027$	$0,893 \pm 0,008$	$0,911 \pm 0,008$	$0,964 \pm 0,017$	$0,974 \pm 0,012$	$0,893 \pm 0,008$
60	$0,584 \pm 0,032$	$0,872 \pm 0,009$	$0,933 \pm 0,005$	$0,961 \pm 0,013$	$0,977 \pm 0,008$	$0,876 \pm 0,008$
80	$0,520 \pm 0,030$	$0,849 \pm 0,010$	$0,959 \pm 0,003$	$0,976 \pm 0,009$	$0,989 \pm 0,005$	$0,877 \pm 0,005$
100	$0,507 \pm 0,028$	$0,839 \pm 0,007$	$0,980 \pm 0,003$	$0,999 \pm 0,002$	$1,000 \pm 0,000$	$0,905 \pm 0,004$
120	$0,534 \pm 0,033$	$0,855 \pm 0,008$	$0,990 \pm 0,001$	$1,000 \pm 0,000$	$1,000 \pm 0,000$	$0,942 \pm 0,004$

As tabelas 7.11 e 7.12 resumem os resultados médios obtidos para as 180 instâncias simuladas e para as 180 instâncias aleatórias, respectivamente, com $N = 100$. O tempo médio de computação da solução ótima para o conjunto de instâncias simuladas foi de 53.529 milissegundos. Este valor é 3,4 vezes maior que o tempo médio da heurística de maior tempo, H_3 , e 262.397 vezes maior que o da heurística de menor tempo, H_1 . Observamos que, para algumas instâncias, H_2 obteve soluções cuja razão de aproximação foi em torno de 60%, todas as demais encontram solução com no mínimo 92% do valor da solução ótima. O tempo de computação médio da solução ótima para as instâncias aleatórias com $N = 100$ foi de 74.108 milissegundos, que é 3,6 vezes maior que o tempo médio de computação da heurística de maior tempo, H_3 , e 129.786 vezes maior que o tempo médio de computação da de menor tempo, H_1 . Podemos notar que a razão de aproximação média para a heurística de referência H_0 foi 58,6%, que é consideravelmente menor que a das demais heurísticas, e para algumas instâncias esta heurística obteve soluções cuja aproximação é 30,5% da ótima.

7.6 Comportamento das Heurísticas em Função de N

Os gráficos mostrados na Figura 7.10 apresentam o comportamento da média das razões de aproximação obtidas para as heurísticas em função de N (quantidade de posições do vetor). À direita da figura temos o gráfico para as instâncias simuladas e, à esquerda, o

Tabela 7.11: Resultados Médios Gerais para $N = 100$ (Instâncias Simuladas).

$N = 100$	H_0	H_1	H_2	H_3	H_4	H_5
Razão de Aprox. Média	0,995	0,992	0,941	0,999	0,999	0,993
Razão de Aprox. Min.	0,917	0,926	0,600	0,986	0,986	0,949
Razão de Aprox. Máx.	1,000	1,000	1,000	1,000	1,000	1,000
Tempo Médio (ms)	0,741	0,204	18,574	15490,749	15456,465	862,743
Tempo Min. (ms)	0,047	0,036	0,102	2463,794	2453,095	105,438
Tempo Max. (ms)	5,881	0,638	155,905	36755,274	39427,780	3110,478

Tabela 7.12: Resultados Médios Gerais para $N = 100$ (Instâncias Aleatórias).

$N = 100$	H_0	H_1	H_2	H_3	H_4	H_5
Razão de Aprox. Média	0,586	0,870	0,945	0,979	0,988	0,900
Razão de Aprox. Min.	0,305	0,790	0,844	0,812	0,878	0,829
Razão de Aprox. Máx.	0,825	0,960	0,997	1,000	1,000	0,961
Tempo Médio (ms)	5,349	0,571	164,469	20558,572	20255,474	0888,455
Tempo Min. (ms)	2,151	0,074	2,493	3158,890	3249,167	139,364
Tempo Max. (ms)	21,902	4,237	908,099	42932,631	41586,062	3101,344

gráfico para as instâncias aleatórias. Nas figuras 7.11 e 7.12 são mostrados os gráficos das médias dos tempos de computação das heurísticas para as instâncias simuladas e aleatórias, respectivamente. Na região à direita destas figuras, temos em detalhes as curvas exclusivas para as heurísticas H_0 , H_1 e H_2 .

Podemos observar que, para as instâncias simuladas, todas as heurísticas apresentam bons resultados médios, sempre acima de 94% do valor ótimo. Para as instâncias aleatórias, percebemos que, de modo geral, as heurísticas propostas superam de forma considerável as soluções encontradas pela heurística H_0 , que apresenta resultados bem piores aos encontrados para as instâncias simuladas.

Quanto ao tempo de computação, notamos que, obviamente, este aumenta com o aumento de N . Para as heurísticas H_0 e H_1 este aumento é menos acentuado quando comparamos às demais. Além disso, podemos observar que o tempo de resolução da solução inteira tem um crescimento mais rápido que o tempo requerido pelas heurísticas e que o tempo de computação de H_3 e H_4 são praticamente os mesmos. Observamos ainda que a heurística H_1 se destaca por conta do tempo de computação bem curto, quando comparado com às outras, e da sutil inclinação crescente da sua curva de tempo.

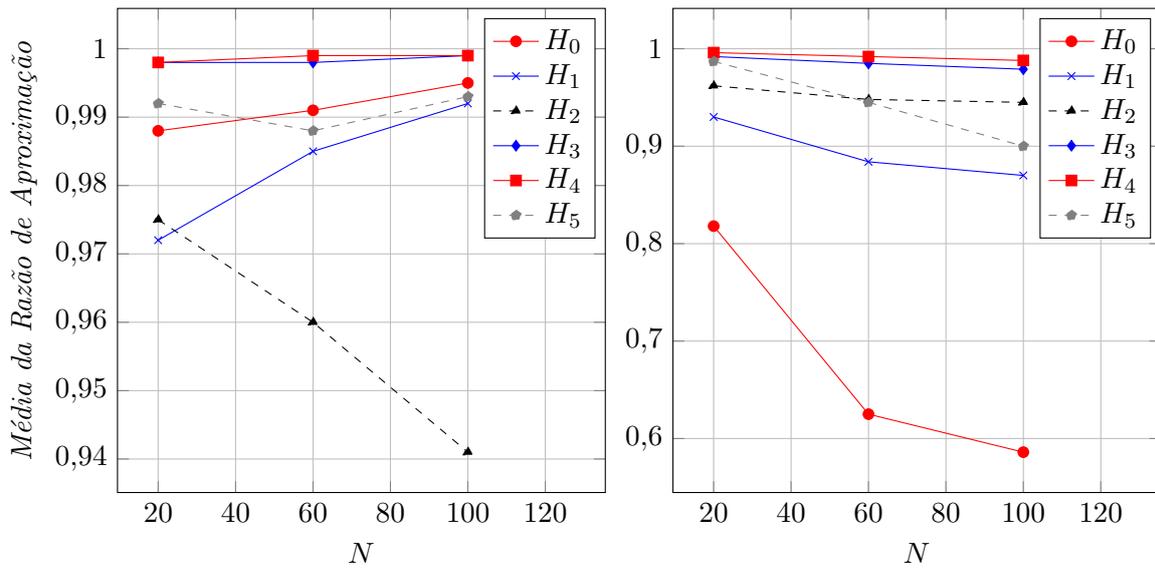


Figura 7.10: Gráfico da Média Razão de Aproximação.

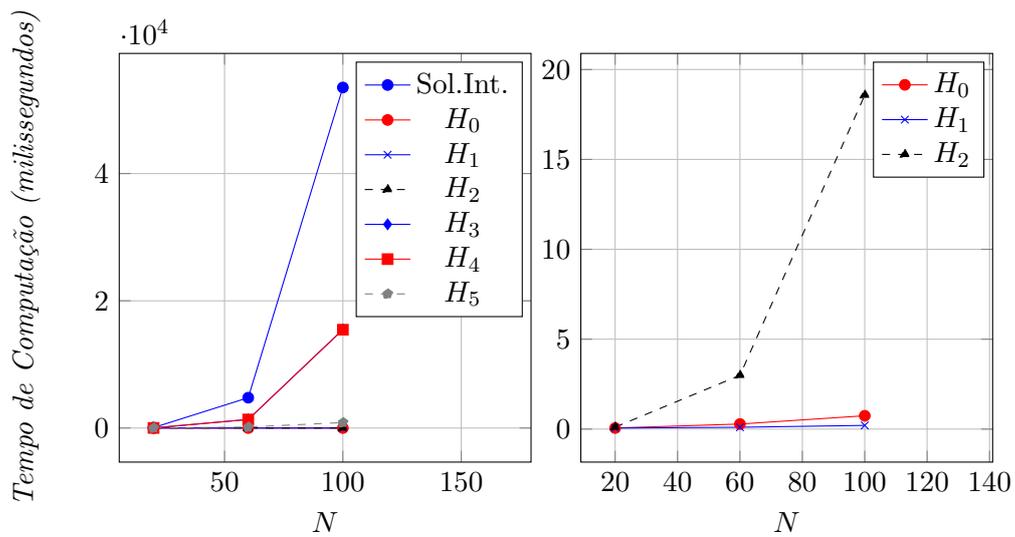


Figura 7.11: Gráficos de Média do Tempo de Computação (Instâncias Simuladas).

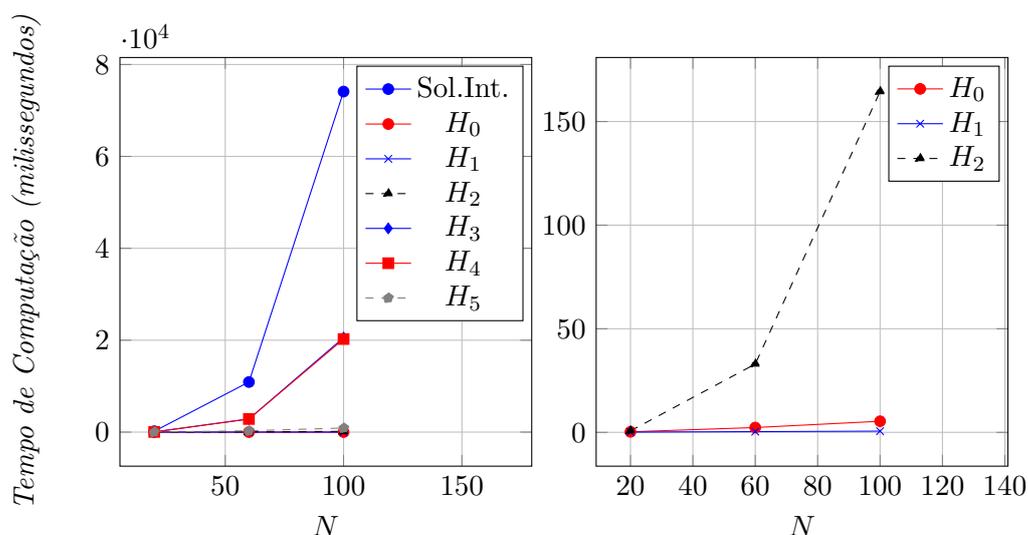


Figura 7.12: Gráficos de Média do Tempo de Computação (Instâncias Aleatórias).

7.7 Conclusões

Por estes resultados, podemos observar que as heurísticas H_3 e H_4 sempre apresentaram melhores resultados que a heurística encontrada na literatura, H_0 , quando comparamos a qualidade de solução obtida. Este resultado já era de se esperar, pois, como as heurísticas H_3 e H_4 utilizam uma solução ótima relaxada, tem-se uma noção global da proximidade da solução ótima inteira, diferentemente das heurísticas H_0 e H_1 que podem ser vistas como heurísticas de baixo custo que buscam encontrar uma “boa solução” com base numa atribuição inicial que pode, potencialmente, estar bem longe da solução ótima.

Em relação ao tempo de computação, observamos que a heurística H_1 tem o menor tempo de computação dentre todas e apresenta resultados próximos aos da heurística H_0 para instâncias simuladas e a supera para instâncias aleatórias. Além disso, como era esperado, as heurísticas H_3 e H_4 apresentam os maiores tempos de computação, já que necessitam calcular uma solução ótima relaxada, e este processo muitas vezes é excessivamente demorado para os padrões da aplicação. A heurística H_2 também pode ser destacada pelo baixo tempo de computação, apesar de ser maior que os de H_0 e H_1 .

Pelos resultados apresentados, o que de imediato observamos é que o custo de se obter “boas soluções” com as heurísticas H_3 e H_4 corresponde ao custo de se computar uma solução ótima relaxada. Além disso, percebemos também que, devido ao fato das heurísticas H_1 e H_2 possuírem tempo de computação consideravelmente baixo, estas podem ser uma alternativa para o contexto de aplicação, pois o tempo de resposta do algoritmo é de fundamental

importância na prática.

Os resultados sugerem que as instâncias simuladas do problema, de modo geral, são mais fáceis de resolver que instâncias geradas aleatoriamente. Observe que os tempos médios de computação das heurísticas, ou da solução ótima, são sempre menores para as instâncias simuladas. Além disso, todas as heurísticas conseguem encontrar soluções em que a média da razão de aproximação é pelo menos 90%, diferente ao que ocorre para as instâncias aleatórias.

Capítulo 8

Considerações Finais e Trabalhos Futuros

Neste capítulo, fazemos nossas considerações finais e projetamos algumas possibilidades de continuidade com relação ao tema abordado.

8.1 Considerações Finais

Nesta seção, resumimos o que foi feito neste trabalho e fazemos algumas conclusões a respeito dos resultados obtidos.

Neste trabalho, definimos o problema ACV que foi motivado por uma aplicação de telefonia móvel. De imediato, mostramos que este problema é NP-Difícil a partir de uma redução do problema da Recoloração Convexa de Caminhos (RCC). Além disso, apresentamos um algoritmo aproximativo com fator $1/k$ para um caso particular deste problema, k -ACV.

Devido às exigências de tempo de computação encontradas na aplicação do problema, desenvolvemos 5 heurística para ACV. A heurística H_1 foi inspirada num algoritmo 2-aproximativo do problema RCC. Os testes computacionais mostraram que esta é a mais rápida de todas, inclusive mais rápida que a heurística encontrada na literatura, H_0 . A heurística H_2 possui certa similaridade com H_1 , no que diz respeito à ideia central, que consiste em calcular e utilizar subintervalos de soma máxima para determinar as posições de alocação dos símbolos. Podemos observar que a heurística H_2 também é bastante rápida e, juntamente com H_1 , pode ser uma boa alternativa para uso no contexto de aplicação. As heurísticas H_3 e H_4 utilizam uma solução da relaxação linear para alocar os símbolos. Os testes computacionais mostraram que estas são aquelas de maiores tempo de computação. Tal fato é justificado pelo custo de computação desta solução relaxada. Por outro lado, as soluções encontradas por estas heurísticas são muito próximas das soluções ótimas. A

heurística H_5 aplica Relaxação Lagrangeana sobre a Formulação XY e utiliza o método dos subgradientes para encontrar soluções desta relaxação; A partir de tal solução determina uma solução viável para o problema original. Podemos observar, pelos testes computacionais, que este algoritmo não é tão rápido quanto H_1 e H_2 , mas, na maior parte das vezes, encontra melhores soluções que estas.

Apresentamos também 3 formulações de Programa Linear Inteira para o problema ACV. A primeira, Formulação Z, foi inspirada em uma formulação do problema RCC. A segunda, Formulação X, anteriormente sugerida na literatura, utiliza uma matriz de padrões de atribuição para definir a alocação dos símbolos. A terceira, Formulação XY, também utiliza uma matriz de padrões de atribuição e acrescenta um conjunto de variáveis adicionais, em relação a X, para modelar o problema. Mostramos que a Formulação X corresponde ao problema mestre obtido a partir da aplicação da Decomposição de Dantzig-Wolfe à Formulação Z. Além disso, propomos uma forma para fortalecer a Formulação XY e mostramos que esta torna-se a Formulação X. Como consequência, chegamos a conclusão que $RL(X) = RL(XY Fortalecida) \leq RL(Z)$, onde $RL(F)$ é o valor da relaxação linear do problema F .

Definimos e estudamos o poliedro associado a formulação X. Mostramos que este poliedro tem dimensão plena e provamos que as únicas desigualdades definidoras de facetas com coeficientes negativos são dadas por $x_{ip} \geq 0$. Além disso, definimos uma classe de desigualdades válidas com lado direito igual a 1 e provamos que esta juntamente com as restrições de não-negatividade compreende todas as desigualdades indutoras de facetas com coeficientes binários. Em seguida, generalizamos estas desigualdades, obtendo uma classe mais ampla, também indutora de facetas do poliedro. Ao final, caracterizamos completamente a envoltória convexa dos pontos viáveis para os casos onde $M = 2$ ou $N = 2$.

8.2 Trabalhos Futuros

Apresentamos, a seguir, algumas possibilidades em relação à continuidade do estudo do problema ACV.

Motivados por características relacionadas à aplicação prática do problema, podemos estudar ACV com algumas restrições adicionais. Por exemplo, é muito comum em sistemas de telefonia móvel a garantia de uma qualidade de serviço; Desta maneira, o acréscimo de uma restrição que garante uma taxa mínima de transferência para cada cliente alocado passa a ser relevante. Assim, se i' é alocado no padrão p' , então o ganho total de i' no padrão p'

deve ser pelo menos um valor mínimo estabelecido, isto é, $\sum_{p \in J(p')} \rho_{i'p} \geq \omega$, onde ω reflete a taxa mínima de qualidade de serviço.

Destacamos também algumas possibilidades para continuidade do estudo poliédrico do problema ACV. Sabemos que existem outras desigualdades válidas que definem facetas. Assim, a busca por novas desigualdades indutoras de facetas é uma opção imediata de estudo. Além disso, o desenvolvimento de heurísticas de separação para as desigualdades estabelecidas e a aplicação do método *branch and cut* podem ser outras possibilidades de trabalho futuro num outro contexto computacional.

Podemos ainda pensar na generalização do problema ACV para alocação de símbolos em matrizes, ao invés de vetores. Desta maneira, podemos definir o problema da Atribuição Conexa em Matrizes (ACM). Neste novo problema, a primeira tarefa a ser feita é definir o que é uma atribuição conexa em matrizes. Com isso, similarmente ao que ocorre em ACV, devemos maximizar o ganho da atribuição A (agora, uma função bidimensional, onde $A(k, j)$ define o símbolo i atribuído à posição (k, j) da matriz) e garantir a conexidade desta.

Apêndice A

Resultados de Experimentos Computacionais

As primeiras 6 tabelas abaixo apresentam os resultados referentes a razão de aproximação (RA). As 6 últimas apresentam os resultados de tempo de computação (sempre apresentados em milissegundos). Cada tabela diz respeito a uma classe de instâncias específica com um valor de N fixo. Por exemplo, a primeira, A.1, está relacionada às instâncias reais com $N = 20$. Em cada linha de cada tabela, têm-se as médias dos resultados obtidos para as 30 instâncias relacionadas ao par M e N correspondente.

Tabela A.1: Médias da Razão de Aproximação Para $N = 20$ (Instâncias Reais).

M	N	RA (H_0)	RA (H_1)	RA (H_2)	RA (H_3)	RA (H_4)	RA (H_5)
4	20	0,9923	0,9774	0,9838	1,0000	1,0000	0,9937
8	20	0,9861	0,9713	0,9813	0,9994	0,9994	0,9937
12	20	0,9873	0,9703	0,9820	0,9989	0,9989	0,9947
16	20	0,9859	0,9708	0,9608	0,9921	0,9939	0,9894
20	20	0,9905	0,9775	0,9729	0,9972	0,9978	0,9936
24	20	0,9881	0,9653	0,9670	0,9989	0,9989	0,9881

Tabela A.2: Médias da Razão de Aproximação Para $N = 20$ (Instâncias Artificiais).

M	N	RA (H_0)	RA (H_1)	RA (H_2)	RA (H_3)	RA (H_4)	RA (H_5)
4	20	0,9180	0,9481	0,9507	0,9903	0,9949	0,9799
8	20	0,8510	0,9339	0,9434	0,9822	0,9903	0,9782
12	20	0,8275	0,9471	0,9652	0,9836	0,9916	0,9800
16	20	0,7877	0,9203	0,9703	0,9970	0,9989	0,9867
20	20	0,7463	0,9114	0,9736	1,0000	1,0000	0,9957
24	20	0,7801	0,9192	0,9704	1,0000	1,0000	0,9990

Tabela A.3: Médias da Razão de Aproximação Para $N = 60$ (Instâncias Reais).

M	N	RA (H_0)	RA (H_1)	RA (H_2)	RA (H_3)	RA (H_4)	RA (H_5)
12	60	0,9885	0,9798	0,9714	0,9998	0,9998	0,9877
24	60	0,9897	0,9826	0,9530	0,9957	0,9962	0,9874
36	60	0,9914	0,9869	0,9588	0,9952	0,9975	0,9884
48	60	0,9901	0,9810	0,9483	0,9985	0,9994	0,9846
60	60	0,9937	0,9904	0,9704	0,9986	0,9988	0,9915
72	60	0,9932	0,9878	0,9568	0,9990	0,9995	0,9886

Tabela A.4: Médias da Razão de Aproximação Para $N = 60$ (Instâncias Artificiais).

M	N	RA (H_0)	RA (H_1)	RA (H_2)	RA (H_3)	RA (H_4)	RA (H_5)
12	60	0,7805	0,9314	0,9080	0,9683	0,9798	0,9318
24	60	0,6902	0,9085	0,9248	0,9735	0,9844	0,9178
36	60	0,6110	0,8806	0,9361	0,9836	0,9925	0,9254
48	60	0,5359	0,8583	0,9572	0,9851	0,9956	0,9380
60	60	0,5713	0,8579	0,9750	1,0000	1,0000	0,9640
72	60	0,5599	0,8687	0,9875	0,9999	1,0000	0,9904

Tabela A.5: Médias da Razão de Aproximação Para $N = 100$ (Instâncias Reais).

M	N	RA (H_0)	RA (H_1)	RA (H_2)	RA (H_3)	RA (H_4)	RA (H_5)
20	100	0,9938	0,9923	0,9590	1,0000	1,0000	0,9929
40	100	0,9951	0,9924	0,9378	0,9987	0,9988	0,9941
60	100	0,9944	0,9936	0,9474	0,9989	0,9990	0,9942
80	100	0,9938	0,9911	0,9237	0,9993	0,9995	0,9915
100	100	0,9941	0,9896	0,9370	0,9990	0,9993	0,9916
120	100	0,9962	0,9919	0,9407	0,9982	0,9986	0,9939

Tabela A.6: Médias da Razão de Aproximação Para $N = 100$ (Instâncias Artificiais).

M	N	RA (H_0)	RA (H_1)	RA (H_2)	RA (H_3)	RA (H_4)	RA (H_5)
20	100	0,7158	0,9096	0,8987	0,9768	0,9858	0,9096
40	100	0,6530	0,8933	0,9109	0,9637	0,9743	0,8933
60	100	0,5845	0,8718	0,9330	0,9609	0,9768	0,8760
80	100	0,5203	0,8493	0,9592	0,9756	0,9893	0,8773
100	100	0,5067	0,8392	0,9801	0,9989	0,9998	0,9046
120	100	0,5341	0,8547	0,9896	1,0000	1,0000	0,9415

Tabela A.7: Médias do Tempo de Computação Para $N = 20$ (Instâncias Reais).

M	N	T. (Inteira)	T. (H_0)	T. (H_1)	T. (H_2)	T. (H_3)	T. (H_4)	T. (H_5)
4	20	38,8791	0,1285	0,0664	0,0668	7,7263	6,9464	4,2035
8	20	44,7296	0,0560	0,0691	0,0659	9,5312	9,2800	4,7180
12	20	62,0632	0,0589	0,0558	0,0762	13,4103	13,1992	7,4740
16	20	74,0797	0,0426	0,0548	0,1287	20,8323	21,3873	11,9576
20	20	87,8142	0,0340	0,0628	0,1447	24,8121	24,3390	14,2891
24	20	102,7581	0,0317	0,0393	0,1803	28,7849	29,0041	18,6892

Tabela A.8: Médias do Tempo de Computação Para $N = 20$ (Instâncias Artificiais).

M	N	T. (Inteira)	T. (H_0)	T. (H_1)	T. (H_2)	T. (H_3)	T. (H_4)	T. (H_5)
4	20	62,8525	0,4210	0,1215	0,1536	11,9818	11,5943	6,5528
8	20	163,0831	0,2508	0,1853	0,2483	25,5161	23,9597	9,3235
12	20	222,6908	0,2400	0,1260	0,4952	31,0965	29,6901	13,0500
16	20	195,5727	0,2286	0,0631	0,7899	44,0294	46,6162	18,1661
20	20	205,8257	0,1942	0,0813	1,1438	52,0794	53,3102	21,6570
24	20	255,9993	0,1864	0,0962	1,5336	60,7627	66,0054	28,2879

Tabela A.9: Médias do Tempo de Computação Para $N = 60$ (Instâncias Reais).

M	N	T. (Inteira)	T. (H_0)	T. (H_1)	T. (H_2)	T. (H_3)	T. (H_4)	T. (H_5)
12	60	1031,2475	0,7567	0,1648	0,3100	281,3290	281,1152	40,1693
24	60	2173,0744	0,1880	0,1091	0,8092	619,6262	614,2166	66,9218
36	60	3649,6339	0,1732	0,0565	1,7341	1002,2648	995,1243	100,7321
48	60	5278,2159	0,2170	0,0827	3,5263	1514,2861	1506,0925	159,7681
60	60	7400,9845	0,1668	0,0909	4,7342	2036,7643	2057,0367	246,4544
72	60	9077,8524	0,1868	0,1032	6,8300	2639,8401	2627,2162	350,6718

Tabela A.10: Médias do Tempo de Computação Para $N = 60$ (Instâncias Artificiais).

M	N	T. (Inteira)	T. (H_0)	T. (H_1)	T. (H_2)	T. (H_3)	T. (H_4)	T. (H_5)
12	60	2529,1492	2,8954	0,4278	1,2797	695,7880	701,6319	80,8174
24	60	6249,4521	2,1200	0,2299	4,9934	1856,7761	1788,6871	155,6875
36	60	10640,1686	2,1334	0,1927	14,2119	2840,1306	2783,7177	209,0857
48	60	14219,7628	2,3370	0,3009	33,7105	3460,8403	3436,6357	331,7775
60	60	15158,5615	2,2264	0,3772	60,0721	3975,0159	3995,5650	404,9085
72	60	16593,9138	2,1825	0,4587	83,6117	4350,9702	4421,6607	533,8236

Tabela A.11: Médias do Tempo de Computação Para $N = 100$ (Instâncias Reais).

M	N	T. (Inteira)	T. (H_0)	T. (H_1)	T. (H_2)	T. (H_3)	T. (H_4)	T. (H_5)
20	100	8914,128	0,915	0,261	0,878	2595,649	2588,512	138,578
40	100	27317,837	0,611	0,108	4,823	7741,813	7556,418	377,024
60	100	46716,827	0,689	0,149	10,069	13988,733	13580,835	643,444
80	100	69131,248	0,716	0,186	22,641	19714,801	20404,124	1025,195
100	100	75572,569	0,721	0,237	32,940	21522,264	21298,625	1216,286
120	100	93521,936	0,793	0,284	40,092	27381,232	27310,274	1775,929

Tabela A.12: Médias do Tempo de Computação Para $N = 100$ (Instâncias Artificiais).

M	N	T. (Inteira)	T. (H_0)	T. (H_1)	T. (H_2)	T. (H_3)	T. (H_4)	T. (H_5)
20	100	15025,620	4,005	0,330	4,240	4915,321	4914,923	206,371
40	100	58622,270	5,884	0,439	28,628	16464,185	15784,228	531,891
60	100	96209,925	5,856	0,425	94,972	25367,251	24213,154	767,109
80	100	92203,478	4,891	0,526	141,239	23145,806	23140,613	827,468
100	100	83970,410	4,983	0,666	254,252	23160,625	23024,635	1111,740
120	100	98618,063	6,475	1,042	463,485	30298,246	30455,292	1886,149

Referências Bibliográficas

- [1] Joel Soares and Manoel Campêlo. O problema da atribuição conexa em vetores. *Congresso Nacional de Matemática Aplicada à Indústria - CNMAI*, pages 1–4, 2015.
- [2] Francisco Rafael Marques Lima. *Maximizing spectral efficiency under minimum satisfaction constraints on multiservice wireless networks*. PhD thesis, Universidade Federal do Ceará, 2012.
- [3] Karla Roberta P Sampaio Lima. *Recoloração convexa de caminhos*. PhD thesis, Universidade de Sao Paulo, 2011.
- [4] Douglas B. West. *Introduction to Graph Theory*. Prentice Hall, 2 edition, September 2000.
- [5] John-Adrian Bondy and U. S. R. Murty. *Graph theory*. Graduate texts in mathematics. Springer, New York, London, 2007. OHX.
- [6] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA, 1988.
- [7] C.E. Ferreira. *Uma introdução sucinta a algoritmos de aproximação*. Publicações matemáticas. IMPA, 2001.
- [8] N. MACULAN and M.H.C. Fampa. *Otimização linear*. EdUnB, 2006.
- [9] Carlos Eduardo Ferreira and Yoshiko Wakabayashi. *Combinatória poliédrica e planos-de-corte faciais*. UNICAMP-Instituto de Computacao, 1996.
- [10] H. W. Kuhn and Bryn Yaw. The hungarian method for the assignment problem. *Naval Res. Logist. Quart*, pages 83–97, 1955.
- [11] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network flows: theory, algorithms, and applications*. Prentice Hall, 1993.
- [12] Shlomo Moran and Sagi Snir. Convex recolorings of strings and trees: Definitions, hardness results and algorithms. *Journal of Computer and System Sciences*, 74(5):850 – 869, 2008.
- [13] Richard Karp, Ming Li, Shlomo Moran, and Sagi Snir. Efficient approximation of convex recolorings. *Journal of Computer and System Sciences*, 73(7):1078 – 1089, 2007.

-
- [14] Iyad A. Kanj and Dieter Kratsch. *Computing and Combinatorics: 15th Annual International Conference, COCOON 2009 Niagara Falls, NY, USA, July 13-15, 2009 Proceedings*, chapter Convex Recoloring Revisited: Complexity and Exact Algorithms, pages 388–397. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.
- [15] Karla Roberta Lima and Yoshiko Wakabayashi. Convex recoloring of paths. *Electronic Notes in Discrete Mathematics*, 37:165 – 170, 2011. Proceedings of LAGOS’11 – VI Latin-American Algorithms, Graphs and Optimization Symposium.
- [16] Karla Roberta Lima and Yoshiko Wakabayashi. Convex recoloring of paths. *Discrete Applied Mathematics*, 164, Part 2:450 – 459, 2014.
- [17] A. Schrijver. *Combinatorial Optimization Polyhedra and Efficiency*, volume B. Springer, 2003.
- [18] Kun-Mao Chao and Louxin Zhang. *Sequence Comparison: Theory and Methods*. Springer Publishing Company, Incorporated, 1 edition, 2008.
- [19] Kadane’s algorithm. http://www.algorithmist.com/index.php/Kadane's_Algorithm. Acessada em: 25-07-2016.
- [20] Ju Yuan Hsiao, Chuan Yi Tang, and Ruay Shiung Chang. An efficient algorithm for finding a maximum weight 2-independent set on interval graphs. *Inf. Process. Lett.*, 43(5):229–235, 1992.