

Estudo Poliedral do Problema do Máximo Subgrafo Induzido Comum

Este exemplar corresponde à redação final da
Dissertação devidamente corrigida e defendida
por Breno Piva Ribeiro e aprovada pela Banca
Examinadora.

Campinas, 18 de novembro de 2009.



Prof. Cid C. de Souza (Orientador)

Dissertação apresentada ao Instituto de Com-
putação, UNICAMP, como requisito parcial para
a obtenção do título de Mestre em Ciência da
Computação.

**FICHA CATALOGRÁFICA ELABORADA PELA
BIBLIOTECA DO IMECC DA UNICAMP**

Bibliotecária: Crislene Queiroz Custódio – CRB8 / 7966

Ribeiro, Breno Piva

R354e Estudo poliedral do problema do máximo subgrafo induzido comum /
Breno Piva Ribeiro -- Campinas, [S.P. : s.n.], 2009.

Orientador : Cid Carvalho de Souza

Dissertação (Mestrado) - Universidade Estadual de Campinas,
Instituto de Computação.

1. Máximo subgrafo comum. 2. Combinatória poliédrica. 3.
Programação linear inteira. 4. Algoritmo Branch and Bound. 5.
Algoritmo Branch and Cut. 6. Problema da clique máxima. I. Souza, Cid
Carvalho de. II. Universidade Estadual de Campinas. Instituto de
Computação. III. Título.

Título em inglês: Polyhedral study of the maximum common induced subgraph problem.

Palavras-chave em inglês (Keywords): 1. Maximum common subgraph. 2. Polyhedral combinatorics. 3. Integer programming. 4. Branch and Bound algorithm. 5. Branch and Cut algorithm. 6. Maximum clique problem.

Área de concentração: Otimização combinatória

Titulação: Mestre em Ciência da Computação

Banca examinadora: Prof. Dr. Cid Carvalho de Souza (IC-UNICAMP)
Prof. Dr. Carlos Eduardo Ferreira (IME-USP)
Prof. Dr. Flávio Keidi Miyazawa (IC-UNICAMP)

Data da defesa: 18/09/2009

Programa de Pós-Graduação: Mestrado em Ciência da Computação

TERMO DE APROVAÇÃO

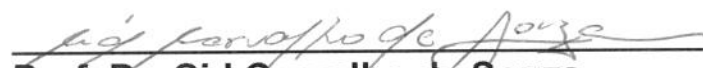
Dissertação Defendida e Aprovada em 18 de setembro de 2009, pela Banca examinadora composta pelos Professores Doutores:



Prof. Dr. Carlos Eduardo Ferreira
IME / Universidade de São Paulo.



Prof. Dr. Flávio Keidi Miyazawa
IC / UNICAMP.



Prof. Dr. Cid Carvalho de Souza
IC / UNICAMP.

Estudo Poliedral do Problema do Máximo Subgrafo Induzido Comum

Breno Piva Ribeiro¹

Novembro de 2009

Banca Examinadora:

- Prof. Cid C. de Souza (Orientador)
- Prof. Carlos Eduardo Ferreira
- Prof. Flávio Keidi Miyazawa
- Prof. Antônio Carlos Moretti (suplente externo)
- Prof. Zanoni Dias (suplente interno)

¹Suporte financeiro de: Bolsa do CNPq 03/2007–09/2007, Bolsa FAPESP (process 2007/53617-4) 10/2007–02/2009.

Resumo

O problema do Máximo Subgrafo Induzido Comum (**MSIC**) pertence à classe \mathcal{NP} -difícil e possui aplicações em diversas áreas. Apesar de sua complexidade, ainda é importante conhecer soluções exatas para instâncias deste problema. Os algoritmos exatos encontrados na literatura buscam resolvê-lo através de técnicas de *backtracking* ou através de sua redução para o problema da Clique Máxima. Neste trabalho procuramos dar uma solução exata para o **MSIC**, tratando-o diretamente através da utilização de modelos de Programação Linear Inteira (**PLI**) e técnicas de combinatória poliédrica. Assim, realizamos um estudo teórico do poliedro do **MSIC** e fomos capazes de encontrar algumas desigualdades válidas fortes, inclusive com provas de que algumas delas representam facetas daquele poliedro. Adicionalmente, provamos que existe uma equivalência entre o modelo **PLI** aqui apresentado para o **MSIC** e uma formulação bem conhecida para o problema da Clique Máxima. Posteriormente, foram implementados algoritmos de *Branch-and-Bound* (**B&B**) e *Branch-and-Cut* (**B&C**) utilizando as desigualdades encontradas e algumas técnicas para tentar tornar os algoritmos mais eficientes. Experimentos foram executados com os algoritmos implementados neste trabalho e, também, com um algoritmo já existente para resolver o problema da Clique, chamado Cliquer. Os resultados foram comparados e, dentre os algoritmos de **PLI**, constatamos que o mais eficiente foi aquele que utilizou uma formulação para o **MSIC** que chamamos de Clique-IS, utilizando **B&B** e técnicas mais básicas que outros algoritmos. Este algoritmo mostrou-se mais eficiente, inclusive, que um algoritmo **PLI** com um modelo baseado no problema da Clique Máxima. Este fato sugere que para uma abordagem baseada em **PLI**, vale a pena utilizar uma formulação do **MSIC** diretamente, ao invés de uma que se apoie na redução deste para o problema da Clique Máxima. Já a comparação do melhor algoritmo desenvolvido neste trabalho com o Cliquer, mostrou que este último é mais eficiente. Para que um algoritmo baseado em **PLI** (utilizando uma formulação com as mesmas variáveis usadas por nós) tivesse alguma chance de vencer um algoritmo combinatório como o Cliquer, seria necessário conhecer mais desigualdades que estivessem ativas na solução ótima do problema.

Abstract

The Maximum Common Subgraph problem (**MSIC**) is in \mathcal{NP} -hard and has applications in several fields. Despite its complexity, it is still important to know exact solutions for instances of this problem. The exact algorithms found in literature try to solve it through backtracking techniques or through its reduction to the Maximum Clique problem. In this work we try to give an exact solution to **MSIC** by addressing it directly, using Linear Integer Programming (**PLI**) and polyhedral combinatorics techniques. So, we performed a study of the **MSIC** polyhedron and we were able to find some strong valid inequalities, including some that were proven to define facets of that polyhedron. Additionally, we proved that an equivalence between the **PLI** model presented here for **MSIC** and a well known formulation for the Maximum Clique problem exists. Later, Branch-and-Bound (**B&B**) and Branch-and-Cut (**B&C**) algorithms were implemented using the inequalities found and some techniques to try to render the algorithms more efficient. Experiments were performed with the algorithms implemented in this work and, also, with an already existing algorithm to solve the Maximum Clique problem, called Cliquer. The results were compared and, among the **PLI** algorithms, we found that the most efficient was the one that used the formulation which we called Clique-IS, using **B&B** and more basic techniques than other algorithms. This algorithm was even more efficient than a **PLI** algorithm with a Clique-based model. This fact suggests that for a **PLI** approach it is worth to use a formulation based on the **MSIC** polyhedron instead of one based on its reduction to the Maximum Clique problem. The comparison of the best algorithm developed in this work with Cliquer, though, showed that the latest is more efficient. In order to some **PLI**-based algorithm (using a formulation with the same variables used by us) to have any chance of outperforming a combinatorial algorithm like Cliquer, it would be necessary to know more inequalities that are active in the problem's optimal solution.

Agradecimentos

Gostaria de agradecer ao CNPq e à FAPESP pelo apoio financeiro que me possibilitou cursar o mestrado fora da minha cidade natal com a tranqüilidade necessária.

Ao meu orientador, Prof. Cid de Souza, por tudo que me ensinou, pela dedicação com que, de fato, me orientou e até pelas muitas vezes necessários puxões de orelha.

Aos colegas do IC, pelo apoio, companhia e momentos de distração, em especial ao pessoal do LOCO, também pelas conversas e discussões. (são muitos nomes pra citar, prefiro não fazer para não correr o risco de esquecer alguém, mas vocês sabem quem são).

Aos companheiros da Plataforma D03: Cristiano, Jaudete, Jean e Rafael, por me atuarem por tanto tempo debaixo do mesmo teto.

Aos meus amigos de Aracaju, principalmente à "gang" e agregados: Adolfo, Ana Paula, Beto, Bruno, Carla, Carol, Danillo, Elfo, Floral, Gabriela, George, Rafinha, Tássia, Teka, Tiago, Tixa e Vírgula e ao pessoal de CC-UFS e agregados: André, Augusto, Bia, Ceará, Henrique, Kalil, Oliver, Pêra, Samuka, Sandra e Xivas. Sem vocês, minha sanidade mental já teria acabado há tempos. Obrigado pelas saídas, conversas, loucuras e confusões.

À toda a minha família, em especial à minha avó, Bernadeth, pelo carinho e apoio; ao meu avô, Nestor, (*in memoriam*) pelo exemplo e inspiração; aos meus irmãos, Gugu, Diogo, Francisco, Luiz, Moema, João Victor, Paulo Henrique e João Eduardo, pelo companheirismo e pelo "trabalho":-); ao meu padrasto, Chiquinho e à minha madrastra, Emília pelo apoio e por, simplesmente, terem passado a fazer parte da minha família.

E, sobretudo, aos meus pais, Jenival e Marta, por terem feito muito mais por mim do que eu seria capaz de citar neste pequeno agradecimento.

Muito obrigado a todos!!

Sumário

Resumo	vii
Abstract	ix
Agradecimentos	xi
1 Introdução	1
2 Fundamentos e Metodologia	3
2.1 Conceitos de Grafos	3
2.2 PLI e Combinatória Poliédrica	4
3 Estudo Teórico do Poliedro do Máximo Subgrafo Induzido Comum	13
3.1 Mapeamento Único	13
3.2 Mapeamento entre Arestas	21
3.3 Outras desigualdades válidas fortes	38
4 O Problema da Clique Máxima	43
4.1 Modelo PLI	43
4.2 MSIC versus Clique	45
4.3 Cliquer	51
5 Implementação e Resultados Computacionais	53
5.1 Implementação	53
5.1.1 Formulações	56
5.1.2 Estratégias de Lifting	61
5.1.3 Rotinas de Separação	63
5.1.4 Estratégias de Seleção de Variáveis	65
5.1.5 Estratégia de Ramificações Múltiplas	66
5.1.6 Estratégias de Ramificação ao Estilo Cliquer	67

5.2 Resultados Computacionais	68
6 Conclusões	95
Bibliografia	97

Lista de Figuras

2.1	Exemplo de árvore de enumeração.	6
2.2	Exemplo de poda por otimalidade.	7
2.3	Exemplo de poda por limitante.	7
2.4	Efeito da ramificação no politopo.	8
2.5	Formulações alternativas para um problema de PLI	9
2.6	Envoltória convexa.	9
2.7	Restrições ativas na solução ótima.	10
3.1	Diagrama da prova da proposição 3.9.	27
3.2	Diagrama parcial da prova da Proposição 3.11.	33
3.3	Diagrama parcial da prova da Proposição 3.11.	34
4.1	Grafo associação construído a partir de dois grafos.	47
4.2	Grafo associação que contém <i>Odd-Antihole</i>	49
4.3	Grafo associação que contém <i>Odd-Hole</i>	50
5.1	Tempo total de execução para diferentes densidades no Experimento02.	71
5.2	Comparação entre os tempos do Experimento02 e Experimento01.	73
5.3	Comparação entre os tempos do Experimento03 e Experimento02.	74
5.4	Comparação entre os tempos do Experimento04 e Experimento02.	76
5.5	Comparação entre os tempos do Experimento05 e Experimento02.	77
5.6	Comparação entre os tempos do Experimento06 e Experimento02.	79
5.7	Comparação entre os tempos do Experimento07 e Experimento02.	80
5.8	Comparação entre os tempos do Experimento08 e Experimento02.	82
5.9	Comparação entre os tempos do Experimento09 e Experimento02.	83
5.10	Comparação entre os tempos do Experimento10 e Experimento02.	84
5.11	Comparação entre os tempos do Experimento11 e Experimento02.	86
5.12	Comparação entre os tempos do Experimento12 e Experimento11.	87
5.13	Comparação entre os tempos do Experimento13 e Experimento11.	88
5.14	Comparação entre os tempos do Experimento14 e Experimento13.	89
5.15	Comparação entre os tempos do Experimento15 e Experimento13.	90

5.16	Comparação entre os tempos do Experimento16 e Experimento13.	92
5.17	Comparação entre os tempos do Experimento17 e Experimento13.	93
5.18	Comparação entre os tempos do Experimento18 e Experimento13.	94

Capítulo 1

Introdução

Utilizando grafos podemos modelar um grande número de problemas do mundo real e um número significativo desses problemas está relacionado a isomorfismos em grafos. Em particular, o problema do Máximo Subgrafo Induzido Comum (**MSIC**) possui diversas aplicações em áreas como Visão Computacional [15, 29], indexação de vídeo [33], Química e Biologia [37, 12, 5]. Este problema pode ser definido da seguinte forma:

Entrada: dois grafos G e H .
Pergunta: encontre um subgrafo induzido G' em G que seja isomorfo a um subgrafo induzido H' em H e cujo número de vértices seja máximo.

Aqui vale lembrar os conceitos de **isomorfismo** e de **subgrafo induzido**. Dois grafos $G = (V, E)$ e $H = (W, F)$ com $|V| = |W|$ são ditos isomorfos se existe uma bijeção $\pi : V \rightarrow W$ tal que $(i, j) \in E \implies (\pi(i), \pi(j)) \in F$ e $(i, j) \in F \implies (\pi^{-1}(i), \pi^{-1}(j)) \in E$. Além disso, dado um grafo $G = (V, E)$, o grafo $H = (W, F)$ é um subgrafo induzido de G se $W \subseteq V$ e $F = \{(i, j) \in E : i \in W \text{ e } j \in W\}$, ou seja, F é o conjunto de arestas com ambas as extremidades em W . Neste caso, diz-se também que H é o subgrafo induzido por W em G .

Uma outra forma de enunciar o problema seria a seguinte. Dados dois grafos, encontre um mapeamento entre subconjuntos de vértices dos dois grafos de forma que esses subconjuntos sejam de tamanho máximo (o subgrafo é máximo) e as seguintes restrições sejam respeitadas: cada vértice de um grafo pode ser mapeado em no máximo um vértice do outro grafo (os subgrafos são isomorfos) e um par de vértices adjacentes de um dos grafos não pode ser mapeado em um par de vértices não-adjacentes do outro grafo (o subgrafo é induzido nos dois grafos).

O **MSIC** é um problema \mathcal{NP} -difícil e faz parte da clássica lista encontrada em [11]. Portanto, a menos que $\mathcal{P} = \mathcal{NP}$, não existem algoritmos polinomiais para resolvê-lo. Por este motivo, um grande número de heurísticas e algoritmos aproximados foi proposto para

o **MSIC** [18, 24, 32, 31, 30, 34, 36]. No entanto, apesar da sua complexidade, ainda é importante conhecer soluções exatas para instâncias, mesmo que pequenas, do problema. Os métodos mais usados na literatura para resolver este problema a otimalidade baseiam-se em algoritmos de *backtracking* como em [23], [39] e [19], ou na redução do problema a um outro problema \mathcal{NP} -difícil mais estudado, como o problema da Clique, a exemplo do que é feito em [9] e em dois dos algoritmos apresentados em [6].

Neste trabalho procuramos tratar o problema diretamente através de modelos de Programação Linear Inteira (**PLI**) e técnicas de combinatória poliédrica. Neste sentido, realizamos uma investigação teórica do problema do **MSIC**, usando o ferramental fornecido pela teoria poliedral. Com isso, almejamos obter desigualdades válidas fortes que permitam uma melhor formulação do problema como um problema de **PLI**. Posteriormente, verificamos o impacto destes resultados teóricos obtidos no desempenho computacional de algoritmos de *Branch-and-Bound* (**B&B**) e *Branch-and-Cut* (**B&C**) desenvolvidos para resolver o problema e, também, comparamos os resultados destes últimos com aqueles alcançados por um algoritmo combinatório existente na literatura.

O restante do texto está organizado da seguinte forma. No Capítulo 2 apresentamos a metodologia que será usada para o desenvolvimento do trabalho e introduzimos os conceitos básicos necessários para o entendimento do restante do texto. O Capítulo 3 apresenta um estudo poliedral para o **MSIC**, incluindo provas de que algumas desigualdades encontradas durante este trabalho definem facetas da envoltória convexa associadas às soluções do problema. O Capítulo 4 discute a relação entre o **MSIC** e o problema da Clique Máxima. Os resultados computacionais obtidos com a implementação de algoritmos de **B&B** e **B&C** são discutidos no Capítulo 5. Por fim, no Capítulo 6 são apresentadas as conclusões da dissertação.

Capítulo 2

Fundamentos e Metodologia

Neste capítulo procuramos apresentar os conceitos que serão utilizados neste trabalho e que são a base para a metodologia usada para atacar o problema. Na primeira seção, apresentamos os conceitos de grafos que serão necessários e na segunda, apresentamos os conceitos de Programação Linear Inteira (**PLI**) e combinatória poliédrica e como tudo isso será usado para solucionar o problema. Obviamente não temos a pretensão de dar um tratamento completo sobre estes assuntos, o que pode ser encontrado em [2, 35, 26, 38, 10].

2.1 Conceitos de Grafos

Como o **MSIC** é um problema da teoria dos grafos, é natural que alguns conceitos desta área sejam necessários para entender e para resolver o problema. Nesta seção, procuramos apresentar os conceitos e as notações utilizadas ao longo do restante do trabalho.

Um **grafo** G é formado por um **conjunto de vértices** V e um **conjunto de arestas** E , onde $E \subseteq V \times V$. Neste caso, denotamos o grafo por $G = (V, E)$. Neste texto, usaremos ainda a notação $V(G)$ para indicar o conjunto de vértices do grafo G e $E(G)$ para indicar o conjunto de arestas deste mesmo grafo. Chamamos de **grafo complementar** de G , denotado por \overline{G} , o grafo em que o conjunto de vértices é igual a $V(G)$ e $E(\overline{G}) = \{(u, v), u \in V(\overline{G}), v \in V(\overline{G}) : (u, v) \notin E(G)\}$. Aqui assumimos que todos os grafos são simples, isto é, não possuem laços (arestas do tipo (v, v)) e não possuem arestas múltiplas (cada par de vértices pode possuir no máximo uma aresta). Aqui, todos os grafos são não-orientados, assim, apesar das arestas serem representadas por pares ordenados, $(u, v) = (v, u)$ para quaisquer vértices u e v .

Dois vértices $u, v \in V(H)$ são ditos **adjacentes** se a aresta (u, v) pertence a $E(H)$. Neste caso, u e v são denominados **extremidades** (ou extremos) de (u, v) . Definimos como **adjacência** ou **vizinhança** de $u \in V(H)$, denotado por $Adj(u)$, o conjunto de todos os vértices de $V(H)$ que são adjacentes a u . Analogamente, definimos a **não-adjacência**

ou **não-vizinhança** de u , denotado por $\overline{Adj(u)}$, como o conjunto de vértices de $V(H)$ que não são adjacentes a u . Chamamos de grau de u o número de vértices adjacentes a este, ou seja, $|Adj(u)|$. Quando $v \in V(G)$ é tal que $|Adj(v)| = 0$, dizemos que v é um vértice **isolado** e, quando $|Adj(v)| = |V(G)| - 1$, dizemos que v é **universal**.

Um **subgrafo** H de G ($H \subseteq G$) é um grafo tal que $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ e toda aresta de $E(H)$ tem extremidades em $V(H)$. Um grafo G é dito **completo** quando para todo par de vértices $u, v \in V(G)$, a aresta (u, v) pertence a $E(G)$. Chamamos de uma **clique** no grafo G , um subgrafo completo deste grafo e chamamos de **conjunto independente de vértices** (aqui abreviado por IS, do inglês, *independent set*) no grafo G , uma clique em \overline{G} . Uma clique (ou um IS) é dita **maximal** se não é possível adicionar nenhum vértice do grafo a ela, de forma que ela permaneça uma clique (IS). Uma clique é dita **máxima** se não existe no grafo nenhuma clique maior que ela. Obviamente, toda clique máxima é também maximal, porém, o contrário não é verdade.

Em um grafo G , chamamos de **caminho**, uma seqüência (v_0, v_1, \dots, v_k) em que $v_0, v_1, \dots, v_k \in V(G)$ e, para $i = 0 \dots k-1$, v_i é adjacente a v_{i+1} e todos os vértices da seqüência são distintos. Um caminho em que o primeiro e o último vértices são iguais é chamado de **ciclo**. Uma **corda** é uma aresta que incide sobre dois vértices não-consecutivos em um ciclo. Um ciclo que não possui cordas é chamado de **hole**. Quando um *hole* possui um número ímpar de vértices, chamamo-lo de **odd-hole**. Um **anti-oddhole** em G é um *odd-hole* em \overline{G} .

Finalmente, dizemos que um grafo $G = (V, E)$ é **n -partido** se V pode ser particionado em n subconjuntos $(V_0, V_1, \dots, V_{n-1})$ de modo que toda aresta (u, v) em E satisfaz a $u \in V_i$ e $v \in V_j$, sendo $i \neq j$ e i e j pertencentes a $\{0, 1, \dots, n-1\}$.

2.2 PLI e Combinatória Poliédrica

Muitos problemas práticos de otimização - em que se procura maximizar ou minimizar uma determinada função objetivo - podem ser descritos como um problema de Programação Linear (**PL**). Dentre estes encontramos: problemas de escalonamento, planejamento de produção, problemas de corte e empacotamento, transportes, telecomunicações, economia etc., além de problemas teóricos da matemática e computação. Um problema de **PL** pode ser escrito em forma matricial como se segue:

$$\begin{aligned} \max \quad & cx \\ & Ax \leq b, \\ & x \in \mathbb{R}_+^n, \end{aligned}$$

onde, A é uma matriz $m \times n$, c é uma matriz $1 \times n$ (vetor linha) e b é uma matriz $m \times 1$ (vetor coluna). Estes, representam os dados do problema. Além disso, x é uma matriz

$n \times 1$ (vetor coluna) de variáveis. Nesta representação, cx é a função objetivo que, no caso, queremos maximizar (vale lembrar que minimizar uma função é equivalente a maximizar o negativo desta função) e $Ax \leq b$ são as restrições às quais as soluções do problema estão sujeitas.

O problema de **PL** pertence à classe \mathcal{P} , isso significa que esse tipo de problema pode ser resolvido por algoritmos polinomiais como, por exemplo, métodos de pontos interiores. No entanto, na prática, o simplex ainda é muito utilizado, pois, apesar de ter complexidade exponencial no pior caso, geralmente, ele apresenta um bom desempenho.

Apesar da grande variedade de problemas que podem ser modelados com **PL**, em diversas situações algumas das variáveis devem ser inteiras. Por exemplo, em um problema de planejamento de transporte, não faz sentido que o número de veículos alocados para executar uma tarefa seja 2,3. Nesta situação, temos um problema de programação inteira mista (quando algumas das variáveis são inteiras e outras reais) ou um caso particular deste que é o problema de **PLI** (quando todas as variáveis são inteiras). Quando as variáveis assumem, exclusivamente, valores binários, isto é, 0 (zero) ou 1 (um), temos o que é chamado de problema de programação inteira binária ou programação inteira 0-1¹. Contudo, apesar do problema de **PL** poder ser resolvido em tempo polinomial, o de **PLI** pertence à classe \mathcal{NP} -difícil e, portanto, a menos que $\mathcal{P} = \mathcal{NP}$, não existem algoritmos polinomiais para resolvê-lo.

A seguir, apresentamos uma formulação **PLI** bem simples para o **MSIC**. Esse modelo baseia-se naquele proposto em [22]:

$$\max \quad \sum_{i \in V(G)} \sum_{j \in V(H)} y_{ij} \quad (2.1)$$

$$\sum_{i \in V(G)} y_{ij} \leq 1, \quad \forall j \in V(H) \quad (2.2)$$

$$\sum_{j \in V(H)} y_{ij} \leq 1, \quad \forall i \in V(G) \quad (2.3)$$

$$y_{ij} + y_{kj} + y_{il} + y_{kl} \leq 1, \quad \forall (i, k) \in E(G), \forall (l, j) \notin E(H) \quad (2.4)$$

$$y_{ij} + y_{kj} + y_{il} + y_{kl} \leq 1, \quad \forall (i, k) \notin E(G), \forall (l, j) \in E(H) \quad (2.5)$$

$$y_{ij} \in \{0, 1\} \quad \forall i \in V(G), \forall j \in V(H) \quad (2.6)$$

Nesta formulação, as variáveis y_{ij} , para todo $i \in V(G)$ e $j \in V(H)$, assumem valor 1 (um) se, e somente se, o vértice i é mapeado no vértice j e assumem valor 0 (zero) caso contrário. Assim, é fácil perceber porque as variáveis nesse problema precisam ser inteiras, afinal, não faz sentido mapear partes de um vértice de G em vértices diferentes do grafo H .

¹Neste texto não falaremos explicitamente em problemas de programação inteira mista ou programação inteira 0-1, ao invés disso, trataremos todos estes problemas como problemas de **PLI**.

As restrições (2.2) e (2.3) garantem, respectivamente, que todo vértice em $V(H)$ é mapeado em apenas um vértice de $V(G)$ e que todo vértice de $V(G)$ é mapeado em um único vértice de $V(H)$. Já as restrições (2.4) e (2.5) garantem, respectivamente, que dois vértices adjacentes de $V(G)$ não são mapeados ao mesmo tempo em dois vértices não-adjacentes de $V(H)$ e que dois vértices adjacentes de $V(H)$ não são mapeados simultaneamente em um par de vértices não-adjacentes de $V(G)$.

Para resolver problemas de **PLI**, comumente são utilizados algoritmos de *Branch-and-Bound* **B&B**. A idéia por trás de um algoritmo deste tipo é dividir o problema em subproblemas menores que são mais fáceis de resolver e depois usar a resolução destes subproblemas para resolver o problema original. No caso de problemas de **PLI**, uma forma natural de dividir o problema é através da adição de novas restrições (por exemplo, fixando uma variável em um determinado valor). A Figura 2.1 (idêntica àquela encontrada em [38] na página 92) mostra o que chamamos de árvore de enumeração para um problema que seria composto por três variáveis binárias. Podemos perceber que a cada nível o problema original é dividido através da fixação de uma das variáveis em 1 (um) ou 0 (zero).

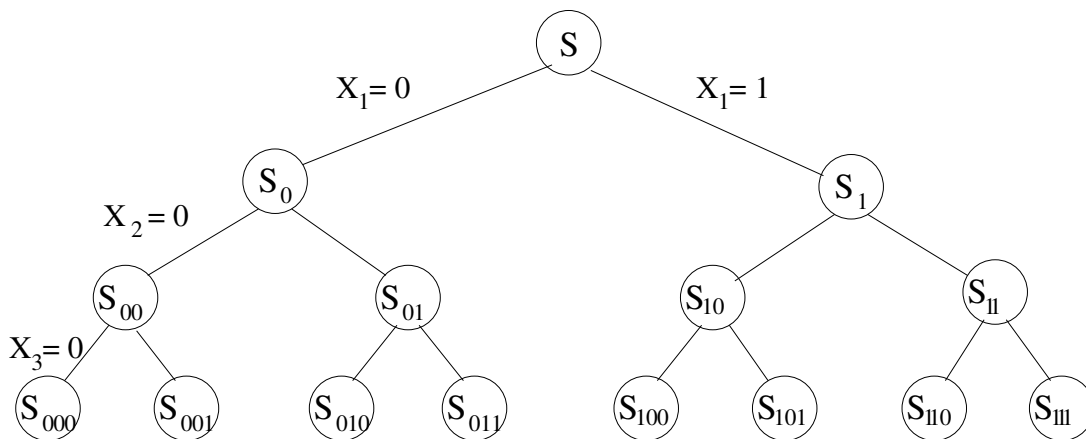


Figura 2.1: Exemplo de árvore de enumeração.

Nas folhas daquela árvore temos todas as possíveis soluções para o problema. No entanto, o número de nós de uma árvore desse tipo cresce exponencialmente com o número de variáveis e, portanto, uma enumeração completa como a exibida na Figura 2.1 é inviável na maioria das vezes. Assim, procura-se fazer uma enumeração implícita através do uso de limitantes. São dois os limitantes de interesse: o **primal** e o **dual**. Para um problema de maximização, o limitante primal (dual) é uma cota inferior (superior) para o valor ótimo. No caso de um problema de minimização, o limitante primal (dual) é uma cota superior (inferior) para o valor ótimo. Um limitante dual muito usado em problemas de **PLI** é aquele fornecido pela relaxação linear - dada pelo problema obtido abandonando-se as restrições de integralidade das variáveis. Tipicamente, um limitante primal pode

ser obtido encontrando-se uma solução viável para o problema, possivelmente através de métodos heurísticos. De posse destes limitantes, podemos chegar à conclusão de que a exploração de alguma subárvore não levará a uma solução melhor que a já encontrada e, assim, podemos podar (deixar de explorar) esta subárvore.

As Figuras 2.2 e 2.3 (também idênticas àquelas encontradas em [38]) mostram exemplos de como as informações dos limitantes podem ser utilizadas para podar a árvore de enumeração. Nessas figuras, os valores à direita dos nós mostram os limitantes a eles associados. Perceba que quando novos limitantes são encontrados nos nós filhos, os limitantes dos nós pais são alterados. Na Figura 2.2 vemos que a subárvore com raiz em S_1 foi podada por otimalidade, já que os limitantes inferior e superior naquele nó são iguais. Já na Figura 2.3 a subárvore com raiz em S_1 foi podada porque o limitante superior nesse nó é menor que o limitante inferior global e, portanto, não existe possibilidade dessa subárvore levar à solução ótima do problema original. Esse tipo de poda é chamada de poda por limitante, por motivos óbvios.

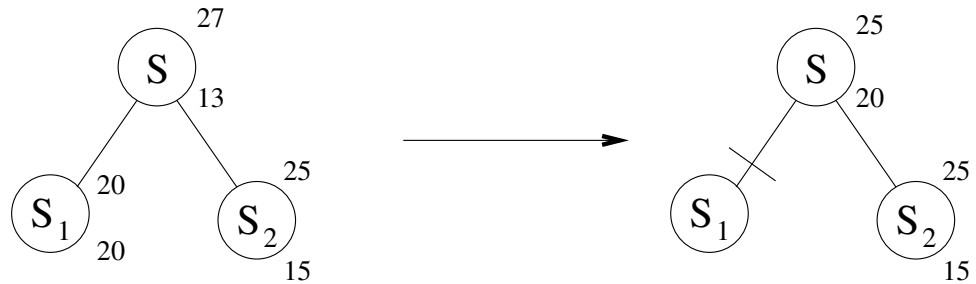


Figura 2.2: Exemplo de poda por otimalidade.

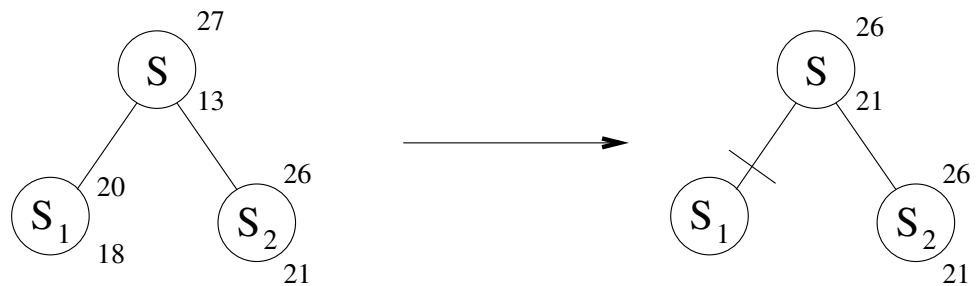


Figura 2.3: Exemplo de poda por limitante.

Observando a Figura 2.4 podemos ter uma visão do efeito do processo de ramificação no polítopo formado pelas restrições da relaxação linear de um problema de **PL**. O problema possui apenas duas variáveis e na resolução da relaxação linear foi encontrada a solução ótima $x_1 = 183/80$ e $x_2 = 7/2$ que não é inteira. Então, o problema é dividido (feita a ramificação) em dois subproblemas. No primeiro é adicionada a restrição $x_1 \leq 2$ e

no segundo é adicionada a restrição $x_1 \geq 3$. Os novos subproblemas são resolvidos e o processo se repete até que uma solução inteira ótima seja encontrada.

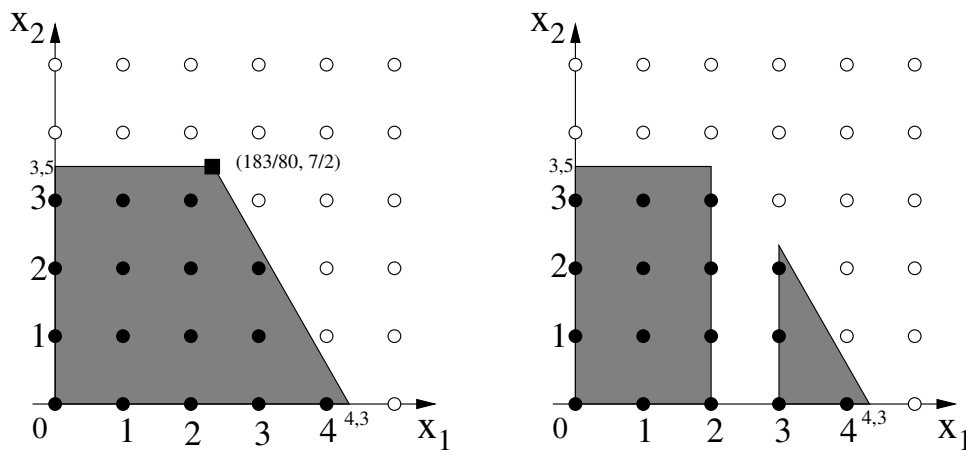


Figura 2.4: Efeito da ramificação no politopo.

Pelo que foi explicado do processo de **B&B** percebe-se que quanto mais podarmos a árvore de enumeração, menos computações precisarão ser feitas e, conseqüentemente, melhor será o desempenho do algoritmo. Como vimos, a possibilidade de fazer podas depende dos limitantes, logo, quanto melhores os limitantes, melhor o desempenho. No caso de relaxações lineares, a qualidade do limitante depende da força da formulação.

Para entender a questão da força de uma formulação deve-se notar que o conjunto de soluções viáveis de um problema de **PLI** (X) pode ser descrito por um número infinito de formulações diferentes. Essa idéia pode ser melhor entendida observando a Figura 2.5. Nesta figura vemos duas formulações que descrevem politopos diferentes mas que têm a mesma intersecção com \mathbb{Z}^n , ou seja, representam o mesmo conjunto de soluções inteiras. Assim, a questão é: qual dentre as infinitas formulações é a mais forte? Essa pergunta pode ser respondida a partir da seguinte observação. Vamos supor que as variáveis são limitadas e, portanto, o poliedro formado pelo conjunto de suas soluções viáveis é um politopo. Neste caso, existe uma solução ótima que se encontra em um ponto extremo do politopo. Assim, se a formulação correspondesse à envoltória convexa do conjunto de soluções inteiras viáveis, denotada por $conv(X)$ (veja Figura 2.6), o problema poderia ser resolvido por um algoritmo de **PL**.

No entanto, para muitos problemas combinatórios, a envoltória convexa só pode ser descrita por um número de desigualdades que cresce exponencialmente em relação ao número de variáveis. Em termos práticos, isto inviabiliza o uso direto do sistema linear que descreve esta envoltória em um resolvedor linear. Em termos teóricos, isto pode representar uma grande dificuldade para se chegar a obter um tal sistema. Contudo, para encontrar a solução ótima não é preciso conhecer toda a envoltória convexa, apenas

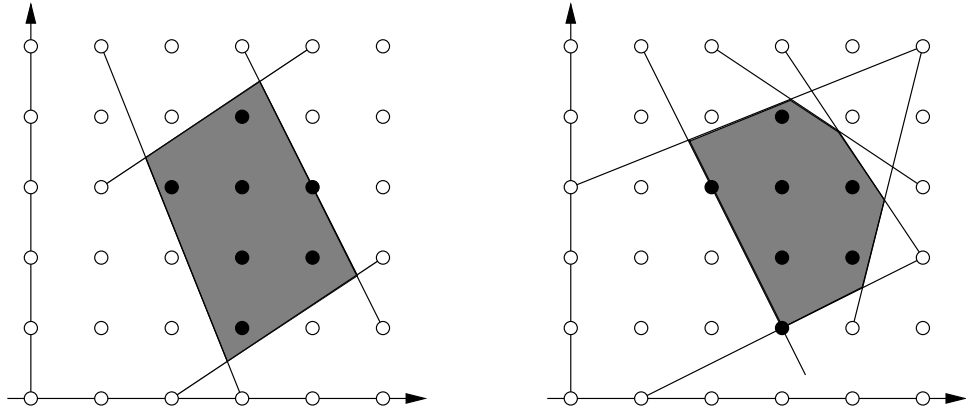


Figura 2.5: Formulações alternativas para um problema de **PLI**.

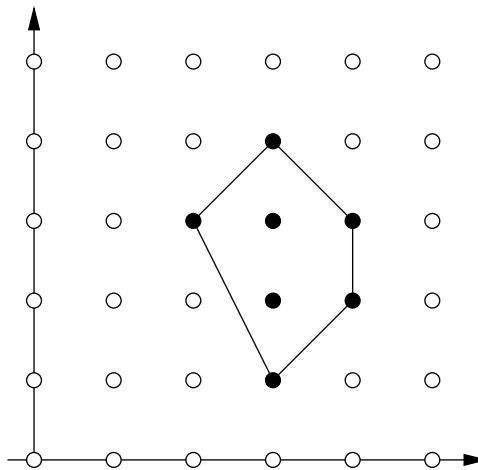


Figura 2.6: Envoltória convexa.

aquelas desigualdades que estão ativas no ponto que representa tal solução. Os algoritmos de planos de corte (**APC**) se apoiam nessa idéia que é ilustrada na Figura 2.7.

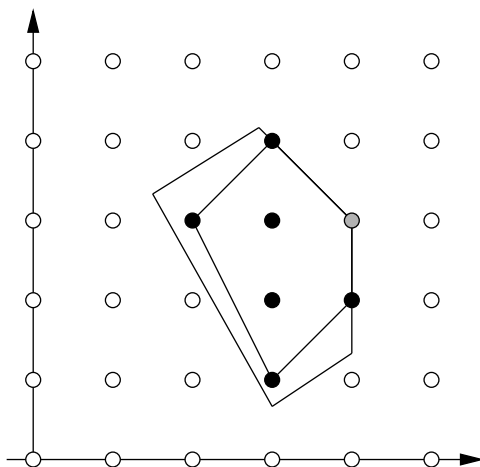


Figura 2.7: Restrições ativas na solução ótima.

Antes de explicarmos como funciona um **APC**, faz-se necessário definir um conceito básico da área conhecida como combinatória poliédrica que é o de desigualdade válida. Esta área tem como tema central o estudo de desigualdades válidas e da estrutura facial dos politopos associados à envoltória convexa das soluções inteiras de problemas formulados como **PLI**. Uma desigualdade é dita válida quando ela é satisfeita por todas as soluções viáveis do problema. Dada uma família de desigualdades válidas para um problema, a idéia de um **APC** é fazer uma reformulação automática do problema através da adição de desigualdades pertencentes a esta família. Como já foi dito, para encontrar a solução inteira ótima usando um algoritmo de **PL** não é preciso conhecer toda a envoltória convexa, assim, dado um problema e uma solução não-inteira ótima da sua relaxação linear, o que fazemos é procurar uma desigualdade da família que é violada pela solução (problema da separação) acrescentando-a à relaxação linear corrente. Desta forma, buscamos aproximar o poliedro da envoltória convexa na região próxima à solução inteira ótima.

A eficácia de um **APC** depende de quão fortes são as desigualdades pertencentes à família utilizada. Quanto mais fortes as desigualdades adicionadas à formulação, melhores os limitantes duais obtidos, em especial, desigualdades que definem faces próprias da envoltória convexa com maior dimensão tendem a produzir melhores limitantes. Preferencialmente, procura-se obter desigualdades que definem facetas da envoltória convexa.

Neste ponto precisamos definir conceitos como: face, face própria, dimensão de um poliedro, poliedro de dimensão cheia e faceta e de posse desses conceitos podemos descrever como provar que uma desigualdade representa uma faceta. Dada uma desigualdade $\pi x \leq$

π_0 válida para o poliedro $P \subseteq \mathbb{R}^n$ definido pela formulação de um problema, chamamos de face definida por $\pi x \leq \pi_0$ em P o conjunto de pontos pertencentes a P que satisfazem a desigualdade na igualdade, ou seja, $F(\pi, \pi_0) = \{x \in P : \pi x = \pi_0\}$ e dizemos que F é uma face própria de P se $F \neq P$ e $F \neq \emptyset$.

Dado um poliedro P (ou uma face F , que também é um poliedro), chamamos de dimensão de P , denotada por $\dim(P)$, o número máximo de pontos afim independentes em P menos um. Assim, dizemos que uma face própria F de P é uma faceta se ela tem dimensão igual a $\dim(P) - 1$. Um poliedro $P = \{x \in \mathbb{R}^n : A^=x = b^=, A^{\leq}x = b^{\leq}\}$, onde $(A^=, b^=)$ e (A^{\leq}, b^{\leq}) são, respectivamente, os conjuntos das igualdades e das desigualdades de P , então, este poliedro é dito de dimensão cheia se ele tem dimensão igual a n . Pode-se mostrar que $\dim(P) = \dim(\mathbb{R}^n) - \rho(A^=)$, onde $\rho(A^=)$ é o posto da matriz $A^=$. Assim, P é de dimensão cheia se, e somente se, $A^=x = b^=$ for vazio, isto é, se não houver igualdades na representação de P .

Apresentaremos a seguir, dois métodos que podem ser usados para provar que uma desigualdade representa uma faceta. O entendimento do segundo método, em particular, é essencial para a compreensão das provas contidas no Capítulo 3. O primeiro método consiste em simplesmente usar a definição e encontrar $n + 1$ pontos pertencentes ao conjunto de soluções viáveis do problema que satisfazem a restrição na igualdade e mostrar que eles são afim independentes (método direto). O segundo método baseia-se no seguinte teorema que pode ser encontrado em [26], página 91 e que a partir deste ponto será referenciado como **teorema de caracterização de facetas**:

Teorema 2.1 *Seja $P = \{x \in \mathbb{R}^n : A^=x = b^=, A^{\leq}x = b^{\leq}\} = \{x \in \mathbb{R}^n : a_i^-x = b_i^-, i \in M^= \wedge a_i^{\leq}x = b_i^{\leq}, i \in M^{\leq}\}$, onde $M^=$ é o conjunto de índices das linhas de $A = A^= \cup A^{\leq}$ que são igualdades e seja $F = \{x \in P : \pi x = \pi_0\}$ uma face própria de P . As duas afirmações a seguir são equivalentes:*

(i) F é uma faceta de P .

(ii) Se $\lambda x = \lambda_0$ para todo $x \in F$ então

$$(\lambda, \lambda_0) = (\alpha\pi + uA^=, \alpha\pi_0 + ub^=) \text{ para algum } \alpha \in \mathbb{R} \text{ e algum } u \in \mathbb{R}^{|M^=|} \quad (2.7)$$

Como as duas afirmações no Teorema 2.1 são equivalentes, se utilizando a equação (2.7) podemos chegar a relações idênticas (entre os componentes de λ e λ_0) às que obtemos usando o fato de $\lambda x = \lambda_0$ para todo $x \in F$, então, isso prova que F é uma faceta. Este método é chamado indireto. Perceba que se o poliedro tem dimensão cheia, a equação (2.7) torna-se: $(\lambda, \lambda_0) = (\alpha\pi, \alpha\pi_0)$, o que facilita significativamente a prova.

Frequentemente quando estamos realizando o estudo poliedral de um problema P , encontramos desigualdades que definem uma face $F = \{x \in P : \pi x = \pi_0\}$ onde $\dim(F) <$

$\dim(P) - 1$, isto é, F não é uma faceta de $\text{conv}(P)$. Nestas situações, às vezes é possível fortalecer a desigualdade $\pi x \leq \pi_0$, que representa F , de modo que ela passe a definir uma face de dimensão maior e, possivelmente, uma faceta. Isso é feito, modificando alguma componente π_i de π , de modo que a nova desigualdade represente uma face F' tal que $\dim(F') > \dim(F)$ e $F \subset F'$. Esse procedimento é chamado de *lifting* e este conceito será muito utilizado nos próximos capítulos.

Neste trabalho, procuramos desenvolver um algoritmo de *Branch-and-Cut* **B&C** para o **MSIC**. Em sua versão mais simples, um **B&C** pode ser entendido como um algoritmo de **B&B** usando **PL** em que a cada nó da árvore de enumeração é executado um **APC**. A motivação para a realização deste tipo de investigação é que utilizando algoritmos de **B&C** várias instâncias de porte razoável de problemas \mathcal{NP} -difíceis foram resolvidas de forma exata na literatura (cf. [1, 7, 13, 14, 27, 28]). Resenhas mais recentes sobre aplicações bem sucedidas de algoritmos baseados em combinatória poliédrica podem ser encontrados em [17, 20, 4].

Capítulo 3

Estudo Teórico do Poliedro do Máximo Subgrafo Induzido Comum

Como mencionado anteriormente, o **MSIC** pertence à classe \mathcal{NP} -difícil, assim, todos os algoritmos exatos conhecidos para resolvê-lo possuem complexidade exponencial. Na prática, este problema tem sido resolvido de forma exata fazendo-se sua redução para o problema da clique máxima.

No presente trabalho procuramos desenvolver um algoritmo para o **MSIC** baseado em um estudo teórico do seu poliedro e não através da sua redução para o problema da clique máxima como tem sido feito na literatura. Com isso, esperávamos ser possível obter um algoritmo que demonstrasse melhor desempenho computacional em relação aos algoritmos já existentes.

Neste capítulo apresentaremos as diversas desigualdades válidas fortes que foram encontradas no estudo teórico do poliedro do **MSIC**, condições sob as quais essas desigualdades são facetas e as respectivas provas. O restante do capítulo será organizado de forma que cada seção trate de um grupo específico de desigualdades. Assim, na Seção 3.1 trataremos das desigualdades que garantem que um vértice de um dos grafos é mapeado em um único vértice do outro grafo; na Seção 3.2 trataremos das desigualdades que determinam que um par de vértices adjacentes de um dos grafos não pode ser mapeado em um par de vértices não-adjacentes do outro grafo e na Seção 3.3 trataremos de outros tipos de desigualdades válidas.

3.1 Mapeamento Único

No modelo apresentado no Capítulo 2, as desigualdades (2.2) e (2.3) têm o objetivo de garantir que cada vértice de um grafo seja mapeado em um único vértice do outro grafo. Sob certas condições aquelas desigualdades definem facetas, porém, antes de provarmos

este fato, precisamos provar a dimensão da envoltória convexa de P , onde P é o conjunto das soluções inteiras do **MSIC** como definido naquele modelo.

Proposição 3.1 *$\text{conv}(P)$ possui dimensão $|V(G)| \times |V(H)|$, ou seja, o poliedro possui dimensão cheia.*

Prova: Seja e_{ij} o vetor que possui 0 (zero) em todas as posições exceto na posição ij .

Então, como todos os vetores e_{ij} , onde $i \in V(G)$ e $j \in V(H)$, satisfazem as restrições de P e o vetor nulo também satisfaz estas restrições, temos $|V(G)| \times |V(H)| + 1$ vetores que, obviamente, são afim independentes (os vetores e_{ij} formam a matriz identidade e o vetor nulo é afim independente com qualquer outro vetor).

Portanto, pela definição de dimensão de um poliedro, temos que a dimensão de $\text{conv}(P)$ é $|V(G)| \times |V(H)|$. \square

Tendo determinado a dimensão de $\text{conv}(P)$ podemos provar que a desigualdade (2.2) representa uma faceta sob as seguintes condições. Dado $j \in V(H)$ fixo:

$$\text{Se existe um vértice universal } k \text{ em } G, \text{ então } j \text{ deve ser universal.} \quad (3.1)$$

$$\text{Se existe um vértice isolado } k \text{ em } G, \text{ então } j \text{ deve ser isolado.} \quad (3.2)$$

Proposição 3.2 *A restrição (2.2) representa uma faceta $F = \{x \in \text{conv}(P) : \sum_{i \in V(G)} y_{ij} = 1\}$, de $\text{conv}(P)$ se, e somente se, as condições (3.1) e (3.2) forem satisfeitas.*

Prova: (\Leftarrow)

Inicialmente devemos mostrar que a face representada pela desigualdade (2.2) é própria, isto é, que a face não é vazia e que existe algum ponto em $\text{conv}(P)$ que não pertence à face. Para a face em questão temos que o vetor nulo não pertence à face e qualquer vetor que possua 0 (zero) em todas as posições exceto na posição $ij : i \in V(G)$ pertencem à face.

Seja $F' = \{x \in \text{conv}(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (2.2) representa uma faceta de $\text{conv}(P)$, então deve existir uma constante positiva $\alpha \in \mathbb{R}$ que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 1)$ onde, $\forall i \in V(G)$, $\gamma_{ij} = 1$ e $\gamma_{ij'} = 0$ com $j \neq j'$. Ou seja, devemos provar que $\pi_0 = \alpha$ e, $\forall i \in V(G)$, que $\pi_{ij} = \alpha = \pi_0$ e $\pi_{ij'} = 0$ com $j \neq j'$.

Se a desigualdade (2.2) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$. Assim, temos:

$$(1) \ (\dots, 0, 1_{ij}, 0, \dots) \in F, \text{ onde } i \in V(G) \Rightarrow (\dots, 0, 1_{ij}, 0, \dots) \in F' \Rightarrow \pi_{ij} = \pi_0.$$

- (2) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $i \in V(G)$, $k \in V(G)$, $l \in V(H) \setminus j$, e i é escolhido de tal forma que se $(l, j) \in E(H)$, então, $(i, k) \in E(G)$ e se $(l, j) \notin E(H)$, então, $(i, k) \notin E(G)$, o que sempre é possível por causa das condições (3.1) e (3.2) $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (2), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in V(G)$ e $l \in V(H) \setminus j$.

(\Rightarrow)

Se (3.1) não é satisfeita, então em (2) não seria possível escolher um i tal que se $(l, j) \notin E(H)$, então, $(i, k) \notin E(G)$ em que k é o vértice universal. Portanto, não existe nenhum vetor em F que tenha o valor 1 na posição kl o que é equivalente a dizer que no poliedro definido por F existe uma igualdade $y_{kl} = 0$. Sendo assim, sua dimensão é menor que $|V(G)| \times |V(H)| - 1$, logo, não é faceta de $\text{conv}(P)$.

Da mesma forma que para o caso de (3.1) não ser satisfeita, se (3.2) não é satisfeita, então em (2) também não seria possível escolher um i tal que se $(l, j) \in E(H)$, então, $(i, k) \in E(G)$ em que k é o vértice isolado. Portanto, não existe nenhum vetor em F que tenha o valor 1 na posição kl o que é equivalente a dizer que no poliedro definido por F existe uma igualdade $y_{kl} = 0$. Sendo assim, sua dimensão é menor que $|V(G)| \times |V(H)| - 1$, logo, não é faceta de $\text{conv}(P)$.

Portanto, se $(\neg(3.1)$ ou $\neg(3.2))$ a desigualdade (2.2) não representa uma faceta de $\text{conv}(P)$. \square

Para a restrição (2.3) as condições e a prova de que a desigualdade representa uma faceta sob aquelas condições são semelhantes às da restrição (2.2) e por isso não iremos apresentá-las aqui.

Agora que provamos que as desigualdades (2.2) e (2.3) representam facetas quando (3.1) e (3.2) são satisfeitas, pensamos se não seria possível modificar estas desigualdades de modo que elas continuem representando facetas mesmo quando estas condições forem violadas. Aplicando procedimentos de *lifting* sobre as variáveis que aparecem quando aquelas condições não são satisfeitas chegamos às seguintes desigualdades:

$$\sum_{i \in V(G)} y_{ij} + \sum_{l \in C(j) \setminus \{j\}} \sum_{k \in D_G} y_{kl} + \sum_{l \in I(j) \setminus \{j\}} \sum_{k \in U_G} y_{kl} \leq 1, \quad (3.3)$$

$$\forall j \in V(H), \forall C(j), I(j) \in H$$

$$\sum_{j \in V(H)} y_{ij} + \sum_{l \in D_H} \sum_{k \in C(i) \setminus \{i\}} y_{kl} + \sum_{l \in U_H} \sum_{k \in I(i) \setminus \{i\}} y_{kl} \leq 1, \quad (3.4)$$

$$\forall i \in V(G), \forall C(i), I(i) \in G$$

onde:

- $C(i)$ é uma clique maximal que contém o vértice i ;

- $C(j)$ é uma clique maximal que contém o vértice j ;
- $I(i)$ é um conjunto independente maximal que contém o vértice i ;
- $I(j)$ é um conjunto independente maximal que contém o vértice j ;
- U_H é o conjunto dos vértices universais de H ($U_H \neq \emptyset \Rightarrow D_H = \emptyset$);
- U_G é o conjunto dos vértices universais de G ($U_G \neq \emptyset \Rightarrow D_G = \emptyset$);
- D_H é o conjunto dos vértices isolados de H ($D_H \neq \emptyset \Rightarrow U_H = \emptyset$); e
- D_G é o conjunto dos vértices isolados de G ($D_G \neq \emptyset \Rightarrow U_G = \emptyset$).

Antes de provar qualquer coisa a respeito dessas desigualdades definirem facetas, é necessário provar que estas desigualdades são válidas para o poliedro do **MSIC**. No caso das desigualdades anteriores isso não foi necessário porque elas foram obtidas diretamente da definição do problema e, por isto, a prova era trivial.

Proposição 3.3 *A desigualdade (3.3) é válida para o poliedro do MSIC.*

Prova: Primeiramente devemos notar que $D_G \neq \emptyset \Rightarrow U_G = \emptyset$ e $U_G \neq \emptyset \Rightarrow D_G = \emptyset$. Ainda é possível que $D_G = U_G = \emptyset$ (mas neste caso, as desigualdades seriam iguais a (2.2) e (2.3)).

Suponha que $D_G \neq \emptyset$, então, assuma por contradição que (3.3) não é válida para o poliedro do **MSIC**. Logo, deve existir algum ponto no poliedro para o qual pelo menos duas variáveis no lado esquerdo da desigualdade assumem valor 1 (um) ao mesmo tempo. Para isso acontecer, existem três casos possíveis:

Caso (1): Duas variáveis assumem valor 1 (um) no primeiro somatório. Se isso acontecesse significaria que mais de um vértice de G foi mapeado em j . No entanto, pela definição do **MSIC** isso não é possível.

Caso (2): Duas variáveis assumem valor 1 (um) no segundo somatório. Isso significaria que ou um vértice de G foi mapeado em dois vértices de H , ou que um par de vértices não adjacentes de G (vértices isolados) foi mapeado em um par de vértices adjacentes de H (pertencentes a $C(j)$). Porém, pela definição do **MSIC**, isso também não deve ocorrer.

Caso (3): Uma variável do primeiro somatório e outra do segundo somatório assumem valor 1 (um) ao mesmo tempo. Isso significaria que um vértice i de G foi mapeado em j e ao mesmo tempo, um vértice isolado k de G foi mapeado em um vértice l de H adjacente a j . Portanto, haveria um par de vértices não adjacentes de G

mapeado em um par de vértices adjacentes de H . Porém, isso contraria a definição de subgrafo induzido.

Para o caso em que $U_G \neq \emptyset$ o raciocínio é análogo e quando $D_G = U_G = \emptyset$ a prova recai sobre o caso (1).

Portanto, não existe nenhum ponto do poliedro do **MSIC** para o qual a desigualdade não é satisfeita, o que contraria a hipótese. Logo, a desigualdade (3.3) é válida para aquele poliedro. \square

Agora que provamos que a desigualdade é válida para o poliedro em questão, podemos tentar provar que a desigualdade representa uma faceta. No entanto, isso não é verdade em geral, mas, existem condições necessárias e suficientes para que isso ocorra. Estas condições são:

$$|U_G| \neq 1 \vee I(j) = \overline{Adj(j)} \text{ e} \quad (3.5)$$

$$|D_G| \neq 1 \vee C(j) = Adj(j) \quad (3.6)$$

Proposição 3.4 *A restrição (3.3) induz uma faceta $F = \{x \in conv(P) : \sum_{i \in V(G)} y_{ij} + \sum_{l \in C(j) \setminus \{j\}} \sum_{k \in D_G} y_{kl} + \sum_{l \in I(j) \setminus \{j\}} \sum_{k \in U_G} y_{kl} = 1\}$ de $conv(P)$ se, e somente se, (3.5) e (3.6) forem satisfeitas.*

Prova: (\Leftarrow)

A face definida pela desigualdade (3.3) é própria, pois o vetor nulo não pertence à face e pertence a $conv(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto na posição ij , para algum $i \in V(G)$ pertence à face.

Seja $F' = \{x \in conv(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.3) representa uma faceta de $conv(P)$, então deve existir uma constante $\alpha \in \mathbb{R}$ e positiva que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 1)$ onde $\gamma_{ij'} = 1$ se $j' = j \forall i \in V(G)$ ou $(i \in D_G \text{ e } j' \in C(j))$ ou $(i \in U_G \text{ e } j' \in I(j))$ e $\gamma_{ij'} = 0$, caso contrário. Ou seja, devemos mostrar que $\pi_0 = \alpha$ e $\pi_{ij'} = \alpha = \pi_0$ se $j' = j$ ou $(i \in D_G \text{ e } j' \in C(j))$ ou $(i \in U_G \text{ e } j' \in I(j))$ e $\pi_{ij'} = 0$, caso contrário.

Se a desigualdade (3.3) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$. Assim, temos:

- (1) $(\dots, 0, 1_{ij'}, 0, \dots) \in F$, onde $(i \in V(G) \text{ e } j' = j)$ ou $(i \in D_G \text{ e } j' \in C(j))$ ou $(i \in U_G \text{ e } j' \in I(j)) \Rightarrow (\dots, 0, 1_{ij'}, 0, \dots) \in F' \Rightarrow \pi_{ij'} = \pi_0$.

- (2) $(\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in V(G)$, $i \in V(G) \setminus k$, $l \in V(H) \setminus j$, $j' = j$, então, se $k \notin D_G$ e $k \notin U_G$, i é escolhido de tal forma que se $(l, j') \in E(H)$, então, $(i, k) \in E(G)$, o que sempre é possível pois k não é isolado e se $(l, j') \notin E(H)$, então, $(i, k) \notin E(G)$, o que também sempre é possível pois k não é universal $\Rightarrow (\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij'} + \pi_{kl} = \pi_0$, mas de (1) e (2), temos: $\pi_{ij'} = \pi_0$ e $\pi_{ij'} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in V(G)$, $k \notin D_G$, $k \notin U_G$ e $l \in V(H) \setminus j$.
- (3) $(\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in D_G$, $i \in V(G) \setminus k$, $l \notin C(j)$, $j' \in C(j)$, então, se $|D_G| = 1$, escolhemos $j' = j$ e de (3.6) temos que $(l, j') \notin E(H)$. Se $|D_G| > 1$, então, i é escolhido em D_G e j' é escolhido de tal forma que $(l, j') \notin E(H)$, o que sempre é possível, pois $C(j)$ é maximal. Perceba que para $k \in D_G$, uma das duas situações sempre acontece devido a (3.6). $\Rightarrow (\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij'} + \pi_{kl} = \pi_0$, mas de (1) e (3), temos: $\pi_{ij'} = \pi_0$ e $\pi_{ij'} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in D_G$, $l \notin C(j)$.
- (4) $(\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in U_G$, $i \in V(G) \setminus k$, $l \notin I(j)$, $j' \in I(j)$, então, se $|U_G| = 1$, escolhemos $j' = j$ e de (3.5) temos que $(l, j') \in E(H)$. Se $|U_G| > 1$, então, i é escolhido em U_G e j' é escolhido de tal forma que $(l, j') \in E(H)$, o que sempre é possível, pois $I(j)$ é maximal. Perceba que para $k \in U_G$, uma das duas situações sempre acontece por causa de (3.5). $\Rightarrow (\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij'} + \pi_{kl} = \pi_0$, mas de (1) e (4), temos: $\pi_{ij'} = \pi_0$ e $\pi_{ij'} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in U_G$, $l \notin I(j)$.

(\Rightarrow)

Se (3.5) não é satisfeita, então para algum $k \in U_G$ e $l \notin I(j)$ não adjacente a j não seria possível escolher um $i \in V(G)$ e $j' = j$, com i não adjacente a k , pois k é universal; i não poderia ser universal pois só existe um vértice isolado que é o k ; e i , certamente não poderia ser isolado, pois o k é universal. Portanto, não existe nenhum vetor em F que tenha o valor 1 na posição kl o que é equivalente a dizer que no poliedro definido por F existe uma igualdade $y_{kl} = 0$. Sendo assim, sua dimensão é menor que $|V(G)| \times |V(H)| - 1$, logo, não é faceta de $\text{conv}(P)$.

Se (3.6) não é satisfeita, então para algum $k \in D_G$ e $l \notin C(j)$ adjacente a j não seria possível escolher um $i \in V(G)$ e $j' = j$, com i adjacente a k , pois k é isolado; i não poderia ser isolado pois só existe um vértice isolado que é o k ; e i , certamente não poderia ser universal, pois o k é isolado. Portanto, não existe nenhum vetor em F que tenha o valor 1 na posição kl o que é equivalente a dizer que no poliedro definido por F existe uma igualdade $y_{kl} = 0$. Sendo assim, sua dimensão é menor que $|V(G)| \times |V(H)| - 1$, logo, não é faceta de $\text{conv}(P)$.

Portanto, se $(\neg(3.5)$ ou $\neg(3.6))$ a desigualdade (3.3) não representa uma faceta de $\text{conv}(P)$. □

A prova de que a desigualdade (3.4) é válida, as condições para que ela represente uma faceta e a prova deste fato são semelhantes às apresentadas para a desigualdade (3.3) e por isso iremos suprimi-las.

Apesar das condições que tornam as desigualdades apresentadas acima facetas serem verdadeiras em um grande número de instâncias do **MSIC**, ainda é possível obter desigualdades que representem facetas quando aquelas condições não são satisfeitas. As desigualdades (3.7) e (3.8) apresentadas a seguir representam facetas quando as condições (3.6) ou (3.5), respectivamente, não forem satisfeitas. Da mesma forma, as desigualdades (3.9) e (3.10) representam facetas quando as condições necessárias e suficientes para que a desigualdade (3.4) represente uma faceta não forem satisfeitas.

$$\sum_{i \in V(G)} y_{ij} + \sum_{l \in \text{Adj}(j)} y_{kl} \leq 1, \quad (3.7)$$

$$\forall j \in V(H), \forall D_G, \text{ com } D_G = \{k\}$$

$$\sum_{i \in V(G)} y_{ij} + \sum_{l \notin \text{Adj}(j)} y_{kl} \leq 1, \quad (3.8)$$

$$\forall j \in V(H), \forall U_G, \text{ com } U_G = \{k\}$$

$$\sum_{j \in V(H)} y_{ij} + \sum_{k \in \text{Adj}(i)} y_{kl} \leq 1, \quad (3.9)$$

$$\forall i \in V(G), \forall D_H, \text{ com } D_H = \{l\}$$

$$\sum_{j \in V(H)} y_{ij} + \sum_{k \notin \text{Adj}(i)} y_{kl} \leq 1, \quad (3.10)$$

$$\forall i \in V(G), \forall U_H, \text{ com } U_H = \{l\}$$

Assim como fizemos para as desigualdades anteriores, antes de provar que as desigualdades representam facetas, devemos primeiramente provar que as desigualdades são válidas para o poliedro do **MSIC**. Então, temos:

Proposição 3.5 *A desigualdade (3.7) é válida para o poliedro do MSIC.*

Prova: Assuma por contradição que (3.7) não é válida para o poliedro do **MSIC**. Logo, deve existir algum ponto no poliedro para o qual pelo menos duas variáveis no lado esquerdo da desigualdade assumem valor 1 (um) ao mesmo tempo. Para isso acontecer, existem três casos possíveis:

Caso (1): Duas variáveis assumem valor 1 (um) no primeiro somatório. Se isso acontecesse significaria que mais de um vértice de G foi mapeado em j . No entanto, pela definição do **MSIC** isso não é possível.

Caso (2): Duas variáveis assumem valor 1 (um) no segundo somatório. Isso significaria que ou um vértice de G foi mapeado em dois vértices de H , ou que um par de vértices não adjacentes de G (vértices isolados) foi mapeado em um par de vértices adjacentes de H (pertencentes a $Adj(j)$). Porém, pela definição do **MSIC**, isso também não deve ocorrer.

Caso (3): Uma variável do primeiro somatório e outra do segundo somatório assumem valor 1 (um) ao mesmo tempo. Isso significaria que um vértice i de G foi mapeado em j e ao mesmo tempo, um vértice isolado k de G foi mapeado em um vértice l de H adjacente a j . Portanto, haveria um par de vértices não adjacentes de G mapeado em um par de vértices adjacentes de H . Porém, isso contraria a definição de subgrafo induzido.

Portanto, não existe nenhum ponto do poliedro do **MSIC** para o qual a desigualdade não é satisfeita, o que contraria a hipótese. Logo, a desigualdade (3.7) é válida para aquele poliedro. \square

De forma similar, podemos demonstrar que as demais desigualdades também são válidas para o poliedro do **MSIC**, por isso, omitiremos estas demonstrações.

A seguir apresentamos a demonstração de que a desigualdade (3.7) representa uma faceta de $conv(P)$. Perceba que para esta prova não definimos qualquer condição necessária para que aquele fato seja verdade. Isto acontece porque, pela própria definição da restrição, o conjunto de vértices isolados é unitário e, nessas condições, a desigualdade sempre representa uma faceta.

Proposição 3.6 *A restrição (3.7) representa uma faceta $F = \{x \in conv(P) : \sum_{i \in V(G)} y_{ij} + \sum_{l \in Adj(j)} y_{kl} = 1\}$ de $conv(P)$.*

Prova: A face definida pela desigualdade (3.7) é própria, pois o vetor nulo não pertence à face e pertence a $conv(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto na posição $ij : i \in V(G)$ pertence à face. Seja $F' = \{x \in conv(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.7) representa uma faceta de $conv(P)$, então deve existir uma constante $\alpha \in \mathbb{R}$ e positiva que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 1)$ onde $\gamma_{ij'} = 1$ se $j' = j \forall i \in V(G)$ ou $(i \in D_G$ e $j' \in Adj(j))$ e $\gamma_{ij'} = 0$, caso contrário. Ou seja, devemos mostrar que $\pi_0 = \alpha$ e $\pi_{ij'} = \alpha = \pi_0$ se $j' = j$ ou $(i \in D_G$ e $j' \in Adj(j))$ e $\pi_{ij'} = 0$, caso contrário.

Se a desigualdade (3.7) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$. Assim, temos:

- (1) $(\dots, 0, 1_{ij'}, 0, \dots) \in F$, onde $(i \in V(G) \text{ e } j' = j)$ ou $(i \in D_G \text{ e } j' \in Adj(j))$
 $\Rightarrow (\dots, 0, 1_{ij'}, 0, \dots) \in F' \Rightarrow \pi_{ij'} = \pi_0$.
- (2) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in V(G)$, $i \in V(G) \setminus k$, $l \in V(H) \setminus j$,
então, se $k \notin D_G$, i é escolhido de tal forma que se $(l, j) \in E(H)$, então, $(i, k) \in E(G)$,
o que sempre é possível pois k não é isolado e se $(l, j) \notin E(H)$, então, $(i, k) \notin E(G)$,
o que também sempre é possível pois k não é universal (pois existe um vértice isolado no grafo)
 $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$,
mas de (1) e (2), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in V(G)$, $k \notin D_G$,
 $l \in V(H) \setminus j$.
- (3) $(\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in D_G$, $i \in V(G) \setminus k$, $j' = j$, $l \in V(H) \setminus (j \cup Adj(j))$
 $\Rightarrow (\dots, 0, 1_{ij'}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij'} + \pi_{kl} = \pi_0$, mas
de (1) e (3), temos: $\pi_{ij'} = \pi_0$ e $\pi_{ij'} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall i \in D_G$, $k \in D_G$,
 $l \in V(H) \setminus (j \cup Adj(j))$.

□

As provas de que as desigualdades (3.8), (3.9) e (3.10) são facetas são semelhantes à apresentada para a desigualdade (3.7) e, por isto, serão omitidas.

3.2 Mapeamento entre Arestas

Quando definimos o problema do **MSIC** falamos em mapeamentos de vértices, porém, o termo mapeamento de arestas não tinha sido usado até o presente momento. O mapeamento de uma aresta do grafo G em uma aresta do grafo H é o mapeamento dos vértices sobre os quais a aresta do grafo G incide, nos vértices do grafo H sobre os quais a aresta daquele grafo incide. Assim, como os grafos que tratamos aqui não são direcionados, o mapeamento da aresta (i, j) do grafo G na aresta (k, l) do grafo H pode ser entendido como o mapeamento de i em k e j em l , ou, de i em l e j em k . Analogamente, o mapeamento de uma não-aresta em outra não-aresta é o mapeamento de um par de vértices não-adjacentes de um grafo em outro par de vértices não-adjacentes do outro grafo.

Neste capítulo iremos tratar das desigualdades que foram obtidas a partir das desigualdades do modelo inicial que tinham como objetivo garantir que uma aresta não fosse mapeada numa não-aresta ou vice-versa, tal mapeamento iria de encontro à definição do **MSIC**.

As desigualdades apresentadas no modelo inicial eram muito fracas e, por isso, não tentaremos encontrar condições sob as quais aquelas desigualdades representam facetas. Ao invés disso, apresentaremos generalizações daquelas desigualdades que são mais fortes e

condições sob as quais as novas desigualdades representam facetas, bem como as provas de que as desigualdades de fato representam facetas sob aquelas condições. As desigualdades que a partir deste momento serão tratadas como desigualdades Clique-IS (IS: *independent set*, em inglês, que significa: conjunto independente) são:

$$\sum_{i \in C} \sum_{j \in I} y_{ij} \leq 1, \quad \forall \text{ clique maximal } C \in G, \forall \text{ IS maximal } I \in H \quad (3.11)$$

$$\sum_{i \in I} \sum_{j \in C} y_{ij} \leq 1, \quad \forall \text{ IS maximal } I \in G, \forall \text{ clique maximal } C \in H \quad (3.12)$$

Estas desigualdades determinam que dada uma clique maximal em um dos grafos e um IS maximal no outro grafo, só pode existir o mapeamento de um dos vértices da clique em um dos vértices do IS ou vice-versa. Como mencionado anteriormente, estas desigualdades são generalizações das desigualdades apresentadas no modelo original que podem ser pensadas como desigualdades correspondentes a cliques e ISs com apenas dois vértices.

As restrições (3.11) e (3.12) representam facetas se as seguintes condições forem satisfeitas:

$$\begin{aligned} &\text{Para todo vértice } k \text{ de } G \text{ que não esteja em } C \\ &\quad \text{existe algum vértice em } C \text{ adjacente a } k; \end{aligned} \quad (3.13)$$

$$\begin{aligned} &\text{Para todo vértice } l \text{ de } H \text{ que não esteja em } I \\ &\quad \text{existe algum vértice em } I \text{ não-adjacente a } l. \end{aligned} \quad (3.14)$$

Então podemos apresentar a seguinte proposição:

Proposição 3.7 *A restrição (3.11) representa uma faceta $F = \{x \in \text{conv}(P) : \sum_{i \in C} \sum_{j \in I} y_{ij} = 1\}$ de $\text{conv}(P)$ se, e somente se, a condição (3.13) ou a condição (3.14) for verdadeira.*

Prova: (\Leftarrow)

A face definida pela desigualdade (3.11) é própria, pois o vetor nulo não pertence à face e pertence a $\text{conv}(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto na posição $ij : i \in C$ e $j \in I$ (note que C e I nunca são vazios a não ser que os próprios grafos G e H sejam vazios) pertence à face. Seja $F' = \{x \in \text{conv}(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.11) representa uma faceta de $\text{conv}(P)$, então deve existir uma constante $\alpha \in \mathbb{R}$ e positiva que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 1)$ onde $\gamma_{ij} = 1$ se $i \in C$ e $j \in I$ e $\gamma_{ij} = 0$, caso contrário. Ou seja, devemos provar que $\pi_0 = \alpha$, $\pi_{ij} = \alpha = \pi_0$ se $i \in C$ e $j \in I$ e $\pi_{ij} = 0$, caso contrário.

Se a desigualdade (3.11) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$, assim, temos:

- (1) $(\dots, 0, 1_{ij}, 0, \dots) \in F$, onde $i \in C$ e $j \in I \Rightarrow (\dots, 0, 1_{ij}, 0, \dots) \in F' \Rightarrow \pi_{ij} = \pi_0$.
- (2) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $i \in C$, $k \in C$, $j \in I$, $l \notin I$, e j é escolhido de tal forma que $(j, l) \in E(H)$, o que sempre é possível pois I é maximal.
 $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (2), temos:
 $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C$ e $l \notin I$.
- (3) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $i \in C$, $k \notin C$, $j \in I$ e $l \in I$, e i é escolhido de tal forma que $(i, k) \notin E(G)$, o que sempre é possível pois C é maximal.
 $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (3), temos:
 $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \notin C$ e $l \in I$.
- (4) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $i \in C$, $k \notin C$, $j \in I$, $l \notin I$, e i e j são escolhidos de tal forma que ou $(i, k) \in E(G)$ e $(j, l) \in E(H)$, o que sempre é possível se (3.13) for satisfeita, pois I é maximal, ou, $(i, k) \notin E(G)$ e $(j, l) \notin E(H)$, o que sempre é possível se (3.14) for satisfeita, pois C é maximal
 $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (4), temos:
 $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \notin C$ e $l \notin I$.

(\Rightarrow)

Se (3.13) e (3.14) não são satisfeitas, então em (4) para algum k e l não seria possível escolher um i tal que $(i, k) \in E(G)$ e $(j, l) \in E(H)$ ou $(i, k) \notin E(G)$ e $(j, l) \notin E(H)$. Portanto, não existe nenhum vetor em F que tenha o valor 1 na posição kl o que é equivalente a dizer que no poliedro definido por F existe uma igualdade $y_{kl} = 0$ e. Sendo assim, sua dimensão é menor que $|V(G)| \times |V(H)| - 1$, logo, não é faceta de $\text{conv}(P)$.

Portanto, se $(\neg(3.13) \text{ e } \neg(3.14))$ a desigualdade (3.11) não representa uma faceta de $\text{conv}(P)$. \square

Como foi demonstrado, as desigualdades (3.11) e (3.12) apesar de mais fortes que (2.4) e (2.5) - já que são generalizações destas -, não representam facetas de $\text{conv}(P)$ no caso geral, mas apenas quando a condição (3.13) ou (3.14) é satisfeita. Desta maneira, é natural pensar na possibilidade de encontrar desigualdades que representem facetas quando essas condições não forem satisfeitas. Analisando a prova acima é possível perceber que as

desigualdades podem ser reforçadas, utilizando um procedimento de *lifting*, para os casos em que as condições (3.13) e (3.14) não são satisfeitas. O resultado da aplicação destes procedimentos de *lifting* são as desigualdades que chamaremos SeqClique-SeqIS que são apresentadas a seguir:

$$\sum_{h=0}^{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)-1} \sum_{i \in C(G_h)} \sum_{j \in I(H_h)} y_{ij} \leq 1, \quad \forall \mathcal{SC}(G), \forall \mathcal{SI}(H) \quad (3.15)$$

$$\sum_{h=0}^{\min(|\mathcal{SC}(H)|, |\mathcal{SI}(G)|)-1} \sum_{i \in I(G_h)} \sum_{j \in C(H_h)} y_{ij} \leq 1, \quad \forall \mathcal{SI}(G), \forall \mathcal{SC}(H) \quad (3.16)$$

onde:

- $\mathcal{SC}(G) = (C(G_0), L(G_0), C(G_1), L(G_1), \dots, C(G_p), L(G_p))$, onde:

- (1) $G_0 = G$;
- (2) $C(G_i)$ é clique maximal em G_i , $i = 0 \dots p$;
- (3) $L(G_i) = \{v \in G_i : v \notin C(G_i) \text{ e } \exists u \in C(G_i) \text{ com } (u, v) \in E(G_i)\}^1$, $i = 0 \dots p$.
Ou seja, $L(G_i)$ é o subgrafo formado pelos vértices de $V(G_i)$ que são adjacentes a algum vértice de $C(G_i)$. Observe que $|C(G_i)| = 1 \Rightarrow L(G_i) = \emptyset$;
- (4) $G_{i+1} = G_i - (C(G_i) \cup L(G_i))$, $i = 0 \dots p$. Isto é, o grafo G_{i+1} é obtido removendo de G_i os vértices pertencentes à $C(G_i)$ e toda vizinhança de $C(G_i)$ (que são os vértices em $L(G_i)$). Perceba que assim, entre todos os vértices de $C(G_a)$ e $C(G_b)$, com $a \neq b$, nunca existem arestas (as cliques da seqüência são independentes entre si).

- $\mathcal{SI}(H) = (I(H_0), L(H_0), I(H_1), L(H_1), \dots, I(H_q), L(H_q))$, onde:

- (1) $H_0 = H$;
- (2) $I(H_j)$ é IS maximal em H_j , $j = 0 \dots q$;
- (3) $L(H_j) = \{v \in H_j : v \notin I(H_j) \text{ e } \exists u \in I(H_j) \text{ com } (u, v) \notin E(H_j)\}^2$, $j = 0 \dots q$.
Ou seja, $L(H_j)$ é o subgrafo formado pelos vértices de $V(H_j)$ que são não-adjacentes a algum vértice de $I(H_j)$. Observe que $|I(H_j)| = 1 \Rightarrow L(H_j) = \emptyset$;
- (4) $H_{j+1} = H_j - (I(H_j) \cup L(H_j))$, $j = 0 \dots q$. Isto é, o grafo H_{j+1} é obtido removendo de H_j os vértices pertencentes à $I(H_j)$ e toda não-vizinhança de $I(H_j)$ (que são os vértices em $L(H_j)$). Perceba que assim, entre todos os vértices de $I(H_a)$ e $I(H_b)$, com $a \neq b$, sempre existem arestas (os ISs da seqüência são totalmente conectados entre si).

¹Logicamente, existe também $u' \in C(G_i)$ com $(u', v) \notin E(G_i)$, caso contrário $C(G_i)$ não seria maximal.

²Neste caso, é lógico que existe também $u' \in I(H_j)$ com $(u', v) \in E(H_j)$, caso contrário $I(H_j)$ não seria maximal.

Aqui também faz-se conveniente definir os seguintes conjuntos:

- $R^G = G_{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)}$. Sendo que $R^G \neq \emptyset \Rightarrow R^H = \emptyset$;
- $R^H = H_{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)}$. Sendo que $R^H \neq \emptyset \Rightarrow R^G = \emptyset$.

Como já mencionamos, a desigualdade SeqClique-SeqIS é um *lifting* da desigualdade Clique-IS. Na desigualdade (3.15) o *lifting* é feito da seguinte forma: no grafo G , após determinar uma clique maximal, determina-se um conjunto de vértices que são adjacentes a algum (ou alguns) dos vértices da clique, em seguida, consideramos o subgrafo obtido excluindo-se esse conjunto de vértices e a clique e então obtemos uma nova clique maximal. Este processo é repetido, até que não haja mais nenhum subgrafo a considerar. No grafo H , fazemos o mesmo procedimento, só que ao invés de cliques consideramos ISs maximais e ao invés de vértices adjacentes aos vértices do IS, consideramos vértices não-adjacentes. Desta forma, obtemos uma seqüência de cliques no grafo G e uma seqüência de ISs no grafo H , então atribuímos coeficiente 1 (um) às variáveis relativas ao mapeamento dos vértices da primeira clique no primeiro IS (essas são as variáveis com coeficiente 1 (um) na desigualdade Clique-IS), depois às relativas ao mapeamento dos vértices da segunda clique no segundo IS e assim sucessivamente até que não haja mais cliques ou ISs. Na desigualdade (3.16) o *lifting* é feito de forma equivalente.

Perceba que nesta formulação, diferentes seqüências $\mathcal{SC}(G)$, $\mathcal{SC}(H)$, $\mathcal{SI}(G)$ e $\mathcal{SI}(H)$ dão origem a diferentes desigualdades dos tipos (3.15) e (3.16). Até mesmo a ordem em que cada componente destes conjuntos é considerada, influencia na forma das desigualdades. Assim, é fácil perceber que o número de desigualdades nesta formulação é enorme, no pior caso, crescendo exponencialmente com o número de vértices. Entretanto, algumas das desigualdades deste modelo representam facetas de $\text{conv}(P)$, sob certas condições, o que provaremos a seguir. No entanto, primeiramente faz-se necessário mostrar que as desigualdades são válidas para o poliedro do **MSIC**.

Proposição 3.8 *A desigualdade (3.15) é válida para o poliedro do MSIC.*

Prova: Assuma por contradição que (3.15) não é válida para o poliedro do **MSIC**. Logo, deve existir algum ponto naquele poliedro que não satisfaz a desigualdade. Existem dois casos a considerar.

Caso (1): Duas variáveis assumem valor 1 (um) com um mesmo valor de h . Se isso acontecesse, então, ou um mesmo vértice de G seria mapeado em mais de um vértice de H , ou mais de um vértice de G seria mapeado em um único vértice de H , ou ainda, um par de vértices adjacentes de G (vértices de $C(G_h)$) seria mapeado em um par de vértices não adjacentes de H (vértices de $I(H_h)$). Contudo isso não é possível no **MSIC**.

Caso (2): Duas variáveis assumem valor 1 (um) com valores de h diferentes. Se isso acontecesse, então um par de vértices não adjacentes de G (por construção, as cliques da seqüência são independentes entre si) se mapeariam em um par de vértices adjacentes de H (por construção, os ISs são totalmente conectados entre si). Todavia, tal situação não pode acontecer no **MSIC**.

Portanto, não existe nenhum ponto no poliedro do **MSIC** que não satisfaça à desigualdade, logo, ela é válida para o poliedro em questão. \square

Agora, considere as seguintes condições:

$$\forall C(G_h) \in \mathcal{SC}(G), |C(G_h)| > 1 \quad (3.17)$$

$$\forall I(H_h) \in \mathcal{SI}(H), |I(H_h)| > 1 \quad (3.18)$$

Proposição 3.9 *Se as condições (3.17) e (3.18) são satisfeitas, a restrição (3.15) representa uma faceta $F = \{x \in \text{conv}(P) : \sum_{h=0}^{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)-1} \sum_{i \in C(G_h)} \sum_{j \in I(H_h)} y_{ij} = 1\}$ de $\text{conv}(P)$.*

Prova: Consideraremos nesta prova que $|\mathcal{SC}(G)| < |\mathcal{SI}(H)|$. Para o outro caso a prova é similar. A face definida pela desigualdade (3.15) é própria, pois o vetor nulo não pertence à face e pertence a $\text{conv}(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto na posição $ij : i \in C(G_h)$ e $j \in I(H_h)$ pertence à face.

Seja $F' = \{x \in \text{conv}(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.15) representa uma faceta de $\text{conv}(P)$, então deve existir uma constante $\alpha \in \mathbb{R}$ e positiva que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 1)$ onde $\gamma_{ij} = 1$ se $i \in C(G_h)$ e $j \in I(H_h)$, $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$ e $\gamma_{ij} = 0$, caso contrário. Ou seja, devemos mostrar que $\pi_0 = \alpha$ e $\pi_{ij} = \alpha = \pi_0$ se $i \in C(G_h)$ e $j \in I(H_h)$, $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$ e $\pi_{ij} = 0$, caso contrário.

Para facilitar o acompanhamento e, por conseguinte, o entendimento da prova, a Figura 3.1 apresenta um diagrama com os tipos de mapeamentos contemplados por cada caso da prova. Assim, uma reta ligando a clique $C(G_h)$ ao IS $I(H_h)$, por exemplo representa um mapeamento de um vértice daquela clique em um vértice daquele IS. Os diferentes tipos de retas utilizados representam os diferentes casos da prova de acordo com a legenda na figura. A fim de evitar que a imagem ficasse muito carregada, foram apresentados na figura mapeamentos apenas para o conjunto $C(G_h)$ e $L(G_h)$, no entanto,

pode-se facilmente verificar que os demais mapeamentos são contemplados pelos casos da prova

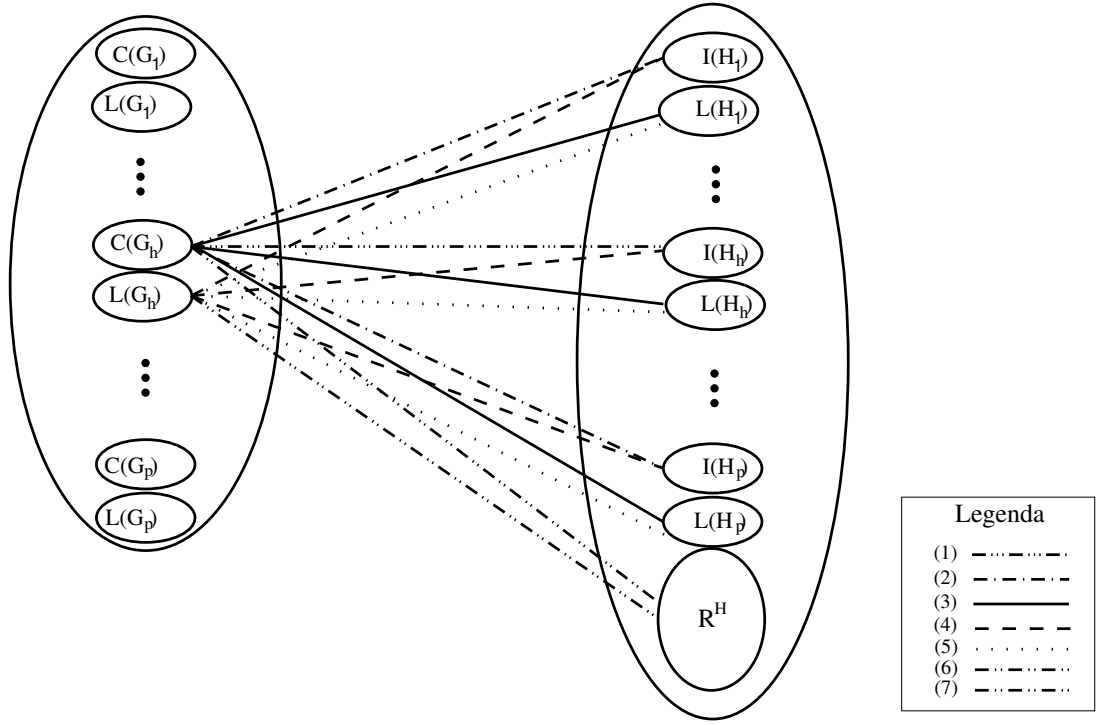


Figura 3.1: Diagrama da prova da proposição 3.9.

Se a desigualdade (3.15) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$, assim, temos:

- (1) $(\dots, 0, 1_{ij}, 0, \dots) \in F$, onde $i \in C(G_h)$ e $j \in I(H_h)$,
 $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1 \Rightarrow (\dots, 0, 1_{ij}, 0, \dots) \in F' \Rightarrow \pi_{ij} = \pi_0$.
- (2) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h)$, $l \in I(H_r)$,
 $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$ e $h \neq r$, então, i é escolhido em $C(G_r)$ e j é escolhido em $I(H_r)$ (o que sempre é possível graças a (3.18)) e, portanto, $(i, k) \notin E(G)$ e $(j, l) \notin E(H) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (2), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h), l \in I(H_r)$,
 $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, h \neq r$.
- (3) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h)$, $l \in L(H_r)$,
 $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então, se $r \geq h$, j é escolhido em $I(H_h)$ de tal forma que $(j, l) \in E(H)$, o que sempre é possível pois $I(H_h)$ é maximal, e i é escolhido em $C(G_h)$ (o que é possível graças a (3.17)) e, portanto, $(i, k) \in E(G)$, pois

os dois vértices pertencem à mesma clique. Se $r < h$, então j é escolhido em $I(H_r)$ de tal forma que $(j, l) \notin E(H)$, o que sempre é possível, pela definição de $L(H_r)$, e i é escolhido em $C(G_r)$ e, portanto, $(i, k) \notin E(G) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (3), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h), l \in L(H_r), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.

(4) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_r), l \in I(H_h)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então, se $r \geq h$, i é escolhido em $C(G_h)$ de tal forma que $(i, k) \notin E(G)$, o que sempre é possível pois $C(G_h)$ é maximal, e j é escolhido em $I(H_h)$ (o que é possível graças a (3.18)) e, portanto, $(j, l) \notin E(H)$, pois os dois vértices pertencem ao mesmo IS. Se $r < h$, então i é escolhido em $C(G_r)$ de tal forma que $(i, k) \in E(G)$, o que sempre é possível, pela definição de $L(G_r)$, e j é escolhido em $I(H_r)$ e, portanto, $(j, l) \in E(H) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (4), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_r), l \in I(H_h), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.

(5) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_h), l \in L(H_r)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então, se $h \leq r$, i é escolhido em $C(G_h)$ de tal forma que $(i, k) \in E(G)$, o que sempre é possível, de outra forma, k faria parte de alguma das cliques da seqüência e j é escolhido em $I(H_h)$ de forma que $(j, l) \in E(H)$, o que sempre é possível porque $I(H_h)$ é maximal. Se $h > r$, então j é escolhido em $I(H_r)$ de tal forma que $(j, l) \notin E(H)$, o que sempre é possível, pela definição de $L(G_h)$, e i é escolhido de $C(G_r)$ de forma que $(i, k) \notin E(G)$, o que sempre é possível, pois $C(G_r)$ é maximal $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (5), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_h), l \in L(H_r), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.

(6) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h)$, $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, $l \in R^H$, então i é escolhido em $C(G_h)$ e, portanto, $(i, k) \in E(G)$ e j é escolhido em $I(H_h)$ e, portanto, $(j, l) \in E(H)$, caso contrário, l pertenceria a $I(H_h)$ ou a $L(H_h) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (6), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H$.

(7) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_h)$, $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, $l \in R^H$, então i é escolhido em $C(G_h)$ de tal forma que $(i, k) \in E(G)$ e j é escolhido em $I(H_h)$ e, portanto, $(j, l) \in E(H)$, caso contrário, l pertenceria a $I(H_h)$ ou a $L(H_h) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (7), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_h), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H$.

□

Para a desigualdade (3.16) a idéia da prova é exatamente a mesma.

Como as desigualdades (3.15) e (3.16) necessitam de condições para que representem facetas, seria interessante encontrar desigualdades que representem facetas quando aquelas condições não são satisfeitas. Seguindo esse raciocínio e aplicando procedimentos de *lifting* sobre as variáveis que aparecem quando as condições não são satisfeitas, obtivemos restrições ainda mais fortes que representam facetas de $conv(P)$. Estas novas restrições que chamamos de LiftSeqClique-SeqIS são apresentadas a seguir:

$$\begin{aligned}
& \sum_{h=0}^{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)-1} \sum_{i \in C(G_h)} \sum_{j \in I(H_h)} y_{ij} + \sum_{k \in MC(G)} \sum_{l \in MI(H)} y_{kl} + \sum_{k \in MC(G)} \sum_{l \in CI(R^H)} y_{kl} + \\
& + \sum_{k \in Is(R^G)} \sum_{l \in MI(H)} y_{kl} + \sum_{h \in mci(G) : \{k\}=C(G_h)} \sum_{l \in F(H_h)} y_{kl} + \sum_{h \in mii(H) : \{l\}=I(H_h)} \sum_{k \in F(G_h)} y_{kl} \leq 1, \\
& \forall \mathcal{SC}(G), \forall \mathcal{SI}(H),
\end{aligned} \tag{3.19}$$

$$\begin{aligned}
& \sum_{h=0}^{\min(|\mathcal{SI}(G)|, |\mathcal{SC}(H)|)-1} \sum_{i \in I(G_h)} \sum_{j \in C(H_h)} y_{ij} + \sum_{k \in MI(G)} \sum_{l \in MC(H)} y_{kl} + \sum_{k \in MI(G)} \sum_{l \in Is(R^H)} y_{kl} + \\
& + \sum_{k \in CI(R^G)} \sum_{l \in MC(H)} y_{kl} + \sum_{h \in mii(G) : \{k\}=I(G_h)} \sum_{l \in F(H_h)} y_{kl} + \sum_{h \in mci(H) : \{l\}=C(H_h)} \sum_{k \in F(G_h)} y_{kl} \leq 1, \\
& \forall \mathcal{SC}(H), \forall \mathcal{SI}(G)
\end{aligned} \tag{3.20}$$

onde:

- $MC(G)$ é o conjunto dos vértices das cliques da sequencia que têm tamanho 1 (um) no grafo G .
- $MI(H)$ é o conjunto dos vértices dos ISs da sequencia que têm tamanho 1 (um) no grafo H .
- $mci(G) = \{k \in \mathbb{N} : k < \min(|\mathcal{SC}|, |\mathcal{SI}|) \wedge |C(G_k)| = 1\}$, isto é, o conjunto dos índices da seqüência cujas cliques têm tamanho 1 (um) no grafo G .
- $mii(H) = \{k \in \mathbb{N} : k < \min(|\mathcal{SC}|, |\mathcal{SI}|) \wedge |I(H_k)| = 1\}$, isto é, o conjunto dos índices da seqüência cujos ISs têm tamanho 1 (um) no grafo H .

- $F(G_r) \subseteq L(G_r)$, $r \in mii(H)$. $F(G_r)$ é maximal e satisfaz as seguintes propriedades:
 - $\forall (u, v)$, $u \in F(G_r)$, $v \in C(G_s)$, $(u, v) \notin E(G)$, $r \in mii(H)$, $s = r + 1 \dots \min(|\mathcal{SC}|, |\mathcal{SI}|) - 1$. Isto é, os vértices pertencentes a $F(G_r)$ são não-adjacentes a todos os vértices das cliques da seqüência em G com índice maior que r ;
 - $\forall (u, v)$, $u \in F(G_r)$, $v \in F(G_s)$, $r \neq s$, $(u, v) \notin E(G)$, $r, s \in mii(H)$. Ou seja, os vértices de $F(G_r)$ e $F(G_s)$ para quaisquer r e s distintos, são não-adjacentes. Perceba que nada é dito sobre os vértices dentro de um determinado $F(G_r)$; e
 - $Is(R^G)$ é um conjunto independente de vértices em R^G t.q. $\forall (u, v)$, $u \in F(G_r)$, $\forall r \in mii(H)$, $v \in Is(R^G)$, $(u, v) \notin E(G)$ e $Is(R^G)$ é maximal satisfazendo as propriedades anteriores. Ou seja, existe um conjunto independente de vértices em R^G tal que todos os vértices de $F(G_r)$, para todo r , são não-adjacentes a todos os vértices deste conjunto. Além disso, esse conjunto é maximal com respeito a esta última propriedade.
- $F(H_r) \subseteq L(H_r)$, $r \in mci(G)$. $F(H_r)$ é maximal e satisfaz as seguintes propriedades:
 - $\forall (u, v)$, $u \in F(H_r)$, $v \in I(H_s)$, $(u, v) \in E(H)$, $r \in mci(G)$, $s = r + 1 \dots \min(|\mathcal{SC}|, |\mathcal{SI}|) - 1$. Isto é, os vértices pertencentes a $F(H_r)$ são adjacentes a todos os vértices dos ISs da seqüência em H com índice maior que r ;
 - $\forall (u, v)$, $u \in F(H_r)$, $v \in F(H_s)$, $r \neq s$, $(u, v) \in E(H)$, $r, s \in mci(G)$. Ou seja, os vértices de $F(H_r)$ e $F(H_s)$ para quaisquer r e s distintos, são adjacentes. Perceba que nada é dito sobre os vértices dentro de um determinado $F(H_r)$; e
 - $Cl(R^H)$ é uma clique em R^H t.q. $\forall (u, v)$, $u \in F(H_r)$, $\forall r \in mci(G)$, $v \in Cl(R^H)$, $(u, v) \in E(H)$ e $Cl(R^H)$ é maximal satisfazendo as propriedades anteriores. Ou seja, existe uma clique em R^H tal que todos os vértices de $F(H_r)$, para todo r , são adjacentes a todos os vértices deste conjunto. Além disso, esse conjunto é maximal com respeito a esta última propriedade.

Os demais conjuntos e símbolos são definidos como anteriormente para as desigualdades (3.15) e (3.16).

Proposição 3.10 *A desigualdade (3.19) é válida para o poliedro do MSIC.*

Prova: Primeiramente devemos notar que o terceiro e o quarto termo do somatório são mutuamente excludentes ($R^H \neq \emptyset \Rightarrow R^G = \emptyset$ e vice-versa). Considere que o número de ISs da seqüência é maior que o número de cliques, portanto, o quarto termo do somatório

é vazio. Agora assuma por contradição que (3.19) não é válida para o poliedro do **MSIC**. Logo, deve existir algum ponto naquele poliedro que não satisfaz à desigualdade. Iremos desconsiderar os casos em que alguma restrição de mapeamento único poderia ser violada:

Caso (1): Duas variáveis assumem valor 1 (um) com valores de h diferentes no primeiro somatório. Se isso acontecesse, então um par de vértices não-adjacentes de G (por construção, as cliques da seqüência são independentes entre si) se mapeariam em um par de vértices adjacentes de H (por construção, os ISs são totalmente conectados entre si). Todavia, tal situação não pode acontecer no **MSIC**.

Caso (2): Duas variáveis assumem valor 1 (um) com valores de h diferentes no quinto ou sexto somatório. Se isso acontecesse, assim como no primeiro caso, então um par de vértices não adjacentes de G (as cliques de tamanho 1 (um) são não-adjacentes, assim como os vértices de dois conjuntos $F(G_r)$ e $F(G_s)$ com $r \neq s$) se mapeariam em um par de vértices adjacentes de H (os ISs de tamanho 1 (um) são adjacentes, assim como os vértices de dois conjuntos $F(H_r)$ e $F(H_s)$ com $r \neq s$). Logo, essa situação não pode ocorrer pois viola a definição do **MSIC**.

Caso (3): Duas variáveis assumem valor 1 (um) em algum dos demais somatórios. O conjunto $MC(G)$ forma um IS e o conjunto $MI(H)$ forma uma clique e em todos os somatórios são considerados os mapeamentos desses conjuntos em conjuntos que são totalmente conectados entre si ou totalmente desconectados entre si, respectivamente, portanto, um ponto do poliedro do **MSIC** nas condições especificadas teria uma aresta mapeada em uma não-aresta, mas pela definição do **MSIC** esse ponto não pertence ao poliedro, o que é uma contradição.

Caso (4): Mais de uma variável assume valor 1 (um) em somatórios diferentes. Quaisquer duas variáveis pertencentes a somatórios diferentes são relativas a vértices não-adjacentes de G e a vértices adjacentes de H ³, portanto, qualquer ponto nas condições especificadas mapeia uma aresta em uma não-aresta e, portanto, não faz parte do poliedro do **MSIC**.

Portanto, não existe nenhum ponto no poliedro do **MSIC** que não satisfaça a desigualdade, logo, ela é válida para o poliedro em questão. \square

Agora, considere as seguintes condições:

³Exceto, possivelmente, variáveis relativas a vértices de algum $F(G_r)$ e $C(G_r)$ que podem ser adjacentes. No entanto, essas variáveis só têm coeficiente 1 (um) quando $|I(H_r)| = 1$ e, neste caso, se as duas variáveis assumissem valor 1 (um) simultaneamente, haveriam dois vértices de G mapeados em um único vértice de H . Uma situação similar pode ocorrer com algum $F(H_r)$ e $I(H_r)$, porém, neste caso, os vértices seriam não-adjacentes.

$$|MC(G)| > 1 \quad (3.21)$$

$$|MI(H)| > 1 \quad (3.22)$$

Proposição 3.11 *Se as condições (3.21) e (3.22) são satisfeitas, a restrição (3.19) representa uma faceta $F = \{x \in \text{conv}(P) : \sum_{h=0}^{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)-1} \sum_{h \notin mci(G) \wedge h \notin mii(H)} \sum_{i \in C(G_h)} \sum_{j \in I(H_h)} y_{ij} + \sum_{k \in MC(G)} \sum_{l \in MI(H)} y_{kl} + \sum_{k \in MC(G)} \sum_{l \in CI(R^H)} y_{kl} + \sum_{k \in Is(R^G)} \sum_{l \in MI(H)} y_{kl} + \sum_{h \in mci(G) : \{k\}=C(G_h)} \sum_{l \in F(H_h)} y_{kl} + \sum_{h \in mii(H) : \{l\}=I(H_h)} \sum_{k \in F(G_h)} y_{kl} = 1\}$ de $\text{conv}(P)$.*

Prova: Consideraremos nesta prova que $|\mathcal{SC}(G)| < |\mathcal{SI}(H)|$ (Neste caso o quarto somatório desaparece). Para o outro caso a prova é similar. A face definida pela desigualdade (3.19) é própria, pois o vetor nulo não pertence à face e pertence a $\text{conv}(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto na posição $ij : i \in C(G_h)$ e $j \in I(H_h)$, para algum $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$ pertencem à face, portanto, a face é própria.

Seja $F' = \{x \in \text{conv}(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.19) representa uma faceta de $\text{conv}(P)$, então deve existir uma constante $\alpha \in \mathbb{R}$ e positiva que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 1)$ onde $\gamma_{ij} = 1$ se a variável y_{ij} possui coeficiente 1 (um) em F e $\gamma_{ij} = 0$, caso contrário. Ou seja, devemos mostrar que $\pi_0 = \alpha$ e $\pi_{ij} = \alpha = \pi_0$ se a variável y_{ij} possui coeficiente 1 (um) em F e $\pi_{ij} = 0$, caso contrário.

Para facilitar o acompanhamento e, por conseguinte, o entendimento da prova, as Figura 3.2 e Figura 3.3 apresentam diagramas com os tipos de mapeamentos contemplados por cada caso da prova. Assim, como na prova da Proposição 3.9 uma reta ligando a clique $C(G_h)$ ao IS $I(H_h)$, por exemplo, representa um mapeamento de um vértice daquela clique em um vértice daquele IS. Também como na prova anterior, os diferentes tipos de retas utilizados representam os diferentes casos da prova de acordo com a legenda na figura. Para esta prova, a figura foi dividida em duas para facilitar a visualização e mostrar todos os casos. A Figura 3.2 mostra casos de mapeamentos de vértices de uma clique com mais de um vértice nos conjuntos da seqüência do outro grafo, enquanto a Figura 3.3 mostra casos de mapeamentos para uma clique com um único vértice.

Se a desigualdade (3.19) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$, assim, temos:

- (1) $(\dots, 0, 1_{ij}, 0, \dots) \in F$, onde ij corresponde a alguma posição em que a variável y_{ij} possui coeficiente 1 (um) em $F \Rightarrow (\dots, 0, 1_{ij}, 0, \dots) \in F' \Rightarrow \pi_{ij} = \pi_0$.

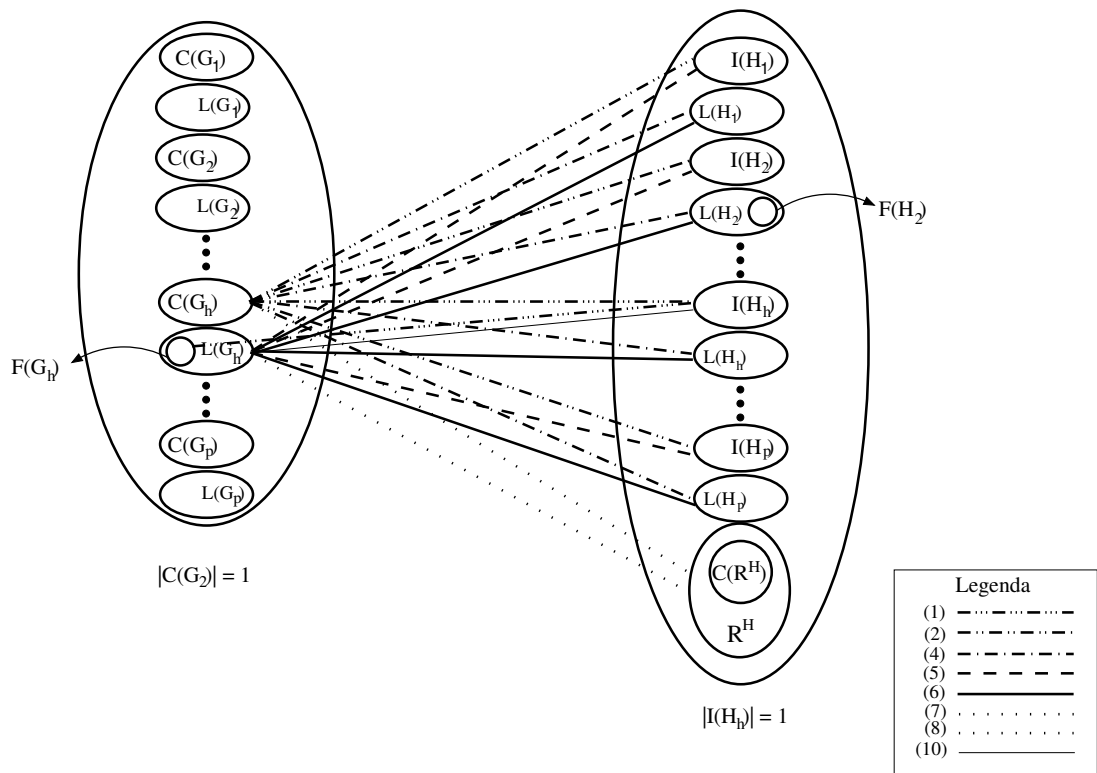


Figura 3.2: Diagrama parcial da prova da Proposição 3.11.

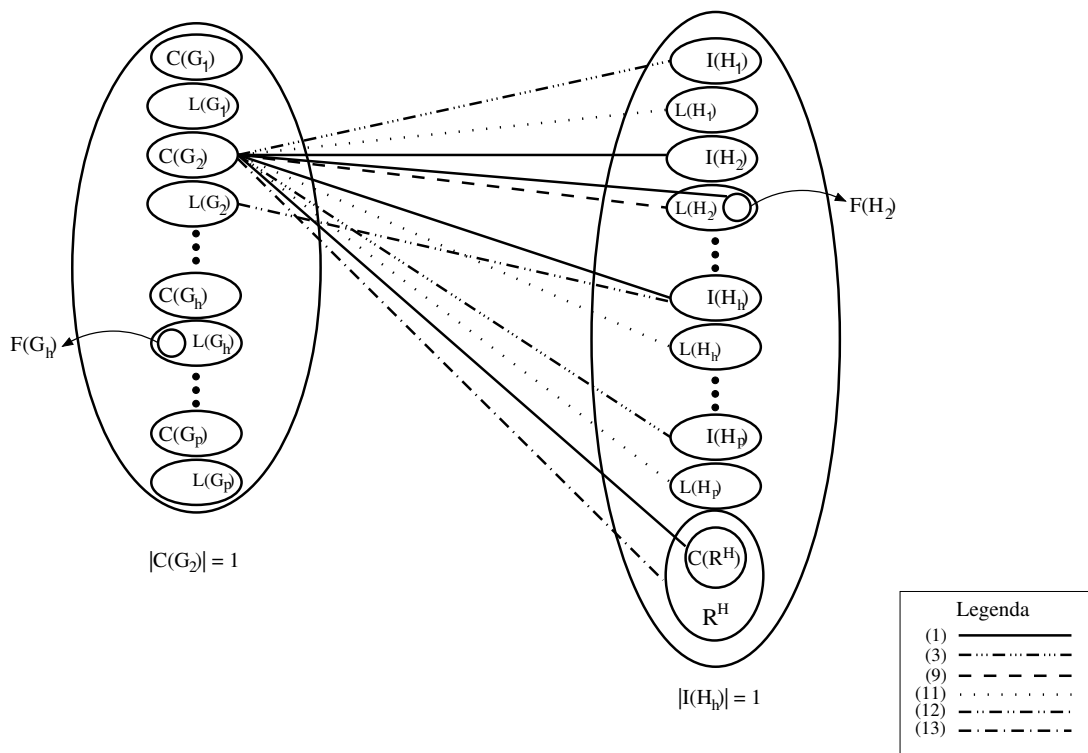


Figura 3.3: Diagrama parcial da prova da Proposição 3.11.

- (2) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h)$, $k \notin MC(G)$, $l \in I(H_r)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$ e $h \neq r$, então, i é escolhido em $C(G_h) \setminus \{k\}$ e j é escolhido em $I(H_h)$ e, portanto, $(i, k) \in E(G)$ e $(j, l) \in E(H) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (2), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h)$, $k \notin MC(G)$, $l \in I(H_r)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, $h \neq r$.
- (3) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h)$, $l \in I(H_r)$, $l \notin MI(H)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$ e $h \neq r$, então, i é escolhido em $C(G_r)$ e j é escolhido em $I(H_r) \setminus \{l\}$ e, portanto, $(i, k) \notin E(G)$ e $(j, l) \notin E(H) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (3), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h)$, $l \in I(H_r)$, $l \notin MI(H)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, $h \neq r$.
- (4) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h)$, $k \notin MC(G)$, $l \in L(H_r)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então, se $r \geq h$, j é escolhido em $I(H_h)$ de tal forma que $(j, l) \in E(H)$, o que sempre é possível pois $I(H_h)$ é maximal, e i é escolhido em $C(G_h) \setminus \{k\}$ e, portanto, $(i, k) \in E(G)$, pois os dois vértices pertencem à mesma clique. Se $r < h$, então j é escolhido em $I(H_r)$ de tal forma que $(j, l) \notin E(H)$, o que sempre é possível, pois, pela definição de $L(H_r)$, tal j deve existir, e i é escolhido em $C(G_r)$ e, portanto, $(i, k) \notin E(G) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (4), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h)$, $k \notin MC(G)$, $l \in L(H_r)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.
- (5) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_r)$, $l \in I(H_h)$, $l \notin MI(H)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então, se $r \geq h$, i é escolhido em $C(G_h)$ de tal forma que $(i, k) \notin E(G)$, o que sempre é possível pois $C(G_h)$ é maximal, e j é escolhido em $I(H_h) \setminus \{l\}$ e, portanto, $(j, l) \notin E(H)$, pois os dois vértices pertencem ao mesmo IS. Se $r < h$, então i é escolhido em $C(G_r)$ de tal forma que $(i, k) \in E(G)$, o que sempre é possível, pois, pela definição de $L(G_r)$, tal i deve existir, e j é escolhido em $I(H_r)$ e, portanto, $(j, l) \in E(H) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (5), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_r)$, $l \in I(H_h)$, $l \notin MI(H)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.
- (6) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_h)$, $l \in L(H_r)$, $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$. Se $h \leq r$, i é escolhido em $C(G_h)$ de tal forma que $(i, k) \in E(G)$, o que sempre é possível, pois, pela definição de $L(G_h)$, tal i sempre existe e j é escolhido em $I(H_h)$ de forma que $(j, l) \in E(H)$, o que sempre é possível porque $I(H_h)$ é maximal. Se $h > r$, então j é escolhido em $I(H_r)$ de tal forma que $(j, l) \notin E(H)$, o que sempre é possível, pois, pela definição de $L(H_r)$, tal j sempre

existe, e i é escolhido de $C(G_r)$ de forma que $(i, k) \notin E(G)$, o que sempre é possível, pois $C(G_r)$ é maximal $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (6), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_h), l \in L(H_r), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.

- (7) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h), k \notin MC(G)$,
 $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H$, então, i é escolhido em $C(G_h) \setminus \{k\}$
e, portanto, $(i, k) \in E(G)$ e j é escolhido em $I(H_h)$ e, portanto, $(j, l) \in E(H)$, caso
contrário, l pertenceria a $I(H_h)$ ou a $L(H_h) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in$
 $F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (8), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} =$
 $0 \forall k \in C(G_h), k \notin MC(G), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in C(R^H)$.
- (8) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_h)$,
 $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H$, então i é escolhido em $C(G_h)$ de tal
forma que $(i, k) \in E(G)$ e j é escolhido em $I(H_h)$ e, portanto, $(j, l) \in E(H)$, caso
contrário, l pertenceria a $I(H_h)$ ou a $L(H_h) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in$
 $F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (8), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} =$
 $0 \forall k \in L(G_h), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H$
- (9) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h), k \in MC(G), l \in L(H_h) \setminus$
 $F(H_h), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então j é escolhido em $I(H_s), s > h$,
ou em $F(H_t), t \in mci(G)$ e $t < h$ de tal forma que $(j, l) \notin E(H)$ e i é escolhido
em $C(G_s)$ ou em $C(G_t)$ (dependendo da escolha do j) e, portanto, $(i, k) \notin E(G)$
 $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (9), temos:
 $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h), k \in MC(G), l \in L(H_h),$
 $l \notin F(H_h), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.
- (10) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_h \setminus F(G_h)), l \in I(H_h), l \in$
 $MI(H), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$, então i é escolhido em $C(G_s), s > h$,
ou em $F(H_t), t \in mii(H)$ e $t < h$ de tal forma que $(i, k) \in E(G)$ e j é escolhido
em $I(H_s)$ ou em $I(H_t)$ (dependendo da escolha do i) e, portanto, $(j, l) \in E(H)$
 $\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (10), temos:
 $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_h), k \notin F(G_h), l \in I(H_h), l \in MI(H),$
 $h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1$.
- (11) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h), k \in MC(G), l \in L(H_r),$
 $h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, h \neq r$, então j é escolhido em $I(H_r)$ de
tal forma que $(j, l) \notin E(H)$, o que sempre é possível, pois, pela definição de
 $L(H_r)$, sempre existe tal j , e i é escolhido em $C(G_r)$ e, portanto, $(i, k) \notin E(G)$

$\Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (11), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h), k \in MC(G), l \in L(H_r), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, h \neq r$.

(12) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in L(G_r), l \in I(H_h), l \in MI(H), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, h \neq r$, então i é escolhido em $C(G_r)$ de tal forma que $(i, k) \in E(G)$, o que sempre é possível, pois, pela definição de $L(G_r)$, tal i sempre existe, e j é escolhido em $I(H_r)$ e, portanto, $(j, l) \in E(H) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (12), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in L(G_r), l \in I(H_h), l \in MI(H), h, r = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, h \neq r$.

(13) $(\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F$, onde $k \in C(G_h), k \in MC(G), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H \setminus Cl(R^H)$, então, j é escolhido em $Cl(R^H)$ ou $F(H_r)$ de tal forma que $(j, l) \notin E(H)$ e i é escolhido em $C(G_r), r \in mci(G)$ (o que sempre é possível graças a (3.21)) e, portanto, $(i, k) \notin E(G) \Rightarrow (\dots, 0, 1_{ij}, \dots, 0, \dots, 1_{kl}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{kl} = \pi_0$, mas de (1) e (13), temos: $\pi_{ij} = \pi_0$ e $\pi_{ij} + \pi_{kl} = \pi_0 \Rightarrow \pi_{kl} = 0 \forall k \in C(G_h), k \in MC(G), h = 0 \dots \min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|) - 1, l \in R^H \setminus Cl(R^H)$.

□

Para a desigualdade (3.20) a idéia da prova é exatamente a mesma.

Quando temos $|MC(G)| = 1, |MI(H)| = 1, |MI(G)| = 1$ ou $|MC(H)| = 1$, as desigualdades (3.19) e (3.20) são modificadas de modo que o terceiro e quarto termos do somatório sejam substituídos adequadamente por:

$$\sum_{l \in R^H : MC(G)=\{k\}} y_{kl} \quad (3.23)$$

$$\sum_{k \in R^G : MI(H)=\{l\}} y_{kl} \quad (3.24)$$

$$\sum_{l \in R^H : MI(G)=\{k\}} y_{kl} \quad (3.25)$$

$$\sum_{k \in R^G : MC(H)=\{l\}} y_{kl} \quad (3.26)$$

A prova da validade dessas desigualdades é idêntica à prova da Proposição 3.10 e a prova de que essas desigualdades representam facetas é muito similar a prova da Proposição 3.11, exceto pelo caso (13) que deixaria de existir porque ele seria contemplado pelo caso (1).

3.3 Outras desigualdades válidas fortes

Além das desigualdades apresentadas, fomos capazes de encontrar outras desigualdades válidas para o poliedro do **MSIC**. Estas outras desigualdades são obtidas a partir de estruturas conhecidas como *Odd-Holes* e *Odd-Antiholes* (ver Capítulo 2). A seguir apresentamos as desigualdades:

$$\sum_{i \in C(G)} \sum_{j \in OH(H)} y_{ij} \leq 2, \quad \forall C(G), |C(G)| \geq 3, \forall OH(H), |OH(H)| = 2k + 1, k > 1 \quad (3.27)$$

$$\sum_{i \in OH(G)} \sum_{j \in C(H)} y_{ij} \leq 2, \quad \forall OH(G), |OH(G)| = 2k + 1, k > 1, \forall C(H), |C(H)| \geq 3 \quad (3.28)$$

$$\sum_{i \in I(G)} \sum_{j \in OA(H)} y_{ij} \leq 2, \quad \forall I(G), |I(G)| \geq 3, \forall OA(H), |OA(H)| = 2k + 1, k > 1 \quad (3.29)$$

$$\sum_{i \in OA(G)} \sum_{j \in I(H)} y_{ij} \leq 2, \quad \forall OA(G), |OA(G)| = 2k + 1, k > 1, \forall I(H), |I(H)| \geq 3 \quad (3.30)$$

$$\sum_{i \in I(G)} \sum_{j \in OH(H)} y_{ij} \leq k, \quad \forall I(G), |I(G)| > k, \forall OH(H), |OH(H)| = 2k + 1, k > 1 \quad (3.31)$$

$$\sum_{i \in OH(G)} \sum_{j \in I(H)} y_{ij} \leq k, \quad \forall OH(G), |OH(G)| = 2k + 1, k > 1, \forall I(H), |I(H)| > k \quad (3.32)$$

$$\sum_{i \in C(G)} \sum_{j \in OA(H)} y_{ij} \leq k, \quad \forall C(G), |C(G)| > k, \forall OA(H), |OA(H)| = 2k + 1, k > 1 \quad (3.33)$$

$$\sum_{i \in OA(G)} \sum_{j \in C(H)} y_{ij} \leq k, \quad \forall OA(G), |OA(G)| = 2k + 1, k > 1, \forall C(H), |C(H)| > k \quad (3.34)$$

onde:

- $C(G)$ é uma clique maximal no grafo G ;
- $I(G)$ é um IS maximal no grafo G ;

- $OH(G)$ é um *Odd-Hole* no grafo G ; e
- $OA(G)$ é um *Odd-Antihole* no grafo G .

A prova que a desigualdade (3.27) é válida para o poliedro do **MSIC** é trivial, pois a maior clique que existe em um *Odd-Hole* tem tamanho 2. Para as desigualdades (3.28), (3.29) e (3.30) as provas seguem a mesma idéia. Perceba que a desigualdade (3.27) foi definida para $|C(G)| \geq 3$. Embora essa desigualdade seja válida para $|C(G)| \geq 2$, se definida dessa forma, ela não representaria uma faceta de $\text{conv}(P) \cap \{x \in \mathbb{B}^{|V(G) \times V(H)|} : x_{ij} = 0 \forall i \notin C(G) \vee j \notin OH(H)\}$, pois, poderia ser obtida a partir de uma combinação linear das desigualdades $\sum_{j \in OH(H)} y_{ij} \leq 1, \forall i \in C(G)$. Efeito semelhante ocorre nas demais desigualdades definidas acima.

Proposição 3.12 *A desigualdade (3.27) representa uma faceta $F = \{x \in \text{conv}(P) : \sum_{i \in C(G)} \sum_{j \in OH(H)} y_{ij} = 2\}$ de $\text{conv}(P) \cap \{x \in \mathbb{B}^{|V(G) \times V(H)|} : x_{ij} = 0 \forall i \notin C(G) \vee j \notin OH(H)\}$.*

Prova: A face definida pela desigualdade (3.27) é própria, pois o vetor nulo não pertence à face e pertence a $\text{conv}(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto nas posições ij e $uv : i, u \in C(G)$ e $j, v \in OH(H), (j, v) \in E(H)$ pertence à face.

Seja $F' = \{x \in \text{conv}(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.27) representa uma faceta de $\text{conv}(P)$, então deve existir uma constante positiva $\alpha \in \mathbb{R}$ que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, 2)$ onde $\gamma_{ij} = 1 \forall i \in C(G)$ e $j \in OH(H)$. Ou seja, devemos mostrar que $\pi_0 = 2\alpha$ e $\pi_{ij} = \alpha = \pi_0/2 \forall i \in C(G)$ e $j \in OH(H)$.

Se a desigualdade (3.27) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$, assim, temos:

- (1) $(\dots, 0, 1_{ij}, 0, \dots, 0, 1_{uv}, 0, \dots) \in F$, onde $i, u \in C(G)$, $j, v \in OH(H)$ e $(j, v) \in E(H) \Rightarrow (\dots, 0, 1_{ij}, 0, \dots, 0, 1_{uv}, 0, \dots) \in F' \Rightarrow \pi_{ij} + \pi_{uv} = \pi_0 \forall i, u \in C(G), j, v \in OH(H)$ e $(j, v) \in E(H)$.

Suponhamos uma ordem para os vértices de $OH(H)$ de forma que o primeiro vértice seja escolhido arbitrariamente e o vértice na posição p deve ser adjacente aos vértices nas posições $(p-1) \bmod (2k+1)$ e $(p+1) \bmod (2k+1)$.

Chamemos de posição ip do vetor que representa um ponto de P , a posição correspondente à variável que representa o mapeamento do vértice $i \in V(G)$ no vértice de $OH(H)$ que recebeu a posição p na ordem de $OH(H)$. Sejam i, k e $h \in C(G)$

Assim, de (1) sabemos que:

$$\begin{aligned}\pi_{i0} + \pi_{u1} = \pi_0 \wedge \pi_{u1} + \pi_{i2} = \pi_0 &\Rightarrow \pi_{i0} = \pi_{i2} \\ \pi_{i1} + \pi_{u2} = \pi_0 \wedge \pi_{u2} + \pi_{i3} = \pi_0 &\Rightarrow \pi_{i1} = \pi_{i3} \\ \pi_{i2k-1} + \pi_{u2k} = \pi_0 \wedge \pi_{u2k} + \pi_{i0} = \pi_0 &\Rightarrow \pi_{i2k-1} = \pi_{i0} \Rightarrow \\ \pi_{i0} = \pi_{i1} = \pi_{i2} = \dots = \pi_{i2k} = \pi_{i0}\end{aligned}$$

Sendo $w \in C(G) \setminus \{i, u\}$, temos também que:

$$\begin{aligned}\pi_{i0} + \pi_{u1} = \pi_0 \wedge \pi_{u1} + \pi_{w2} = \pi_0 &\Rightarrow \pi_{i0} = \pi_{w2} \\ \pi_{i0} + \pi_{w1} = \pi_0 \wedge \pi_{w1} + \pi_{u2} = \pi_0 &\Rightarrow \pi_{i0} = \pi_{u2}\end{aligned}$$

Como podemos obter o mesmo resultado que obtivemos para os π_{ip} , com $p = 0 \dots 2k$ para os π_{up} , com $p = 0 \dots 2k$ e para os π_{wp} , com $p = 0 \dots 2k$, temos que $\pi_{ip} = \pi_{uq} = \pi_{wr}$, com $p, q, r = 0 \dots 2k$. Ou seja, $\pi_{ij} = \alpha \forall i \in C(G)$ e $j \in OH(H)$ e $\pi_0 = 2\alpha$. \square

Neste trabalho não investigamos quais seriam as condições necessárias e suficientes para que (3.27) defina uma faceta de $conv(P)$ no caso geral. Possivelmente, operações de *lifting* poderão ser aplicadas mostrando que a face definida pela desigualdade (3.27) ainda tem dimensão muito baixa.

Provas e conclusões semelhantes podem ser obtidas para as desigualdades (3.28), (3.29) e (3.30).

Proposição 3.13 *A desigualdade (3.31) é válida para o poliedro do MSIC.*

Prova: Suponha para o propósito de se obter uma contradição que (3.31) não é válida para o poliedro do MSIC. Para isso acontecer é preciso que exista um IS maior que k em $OH(H)$. Logo, existe um IS de tamanho $k + 1$ em $OH(H)$. Como $OH(H)$ possui $2k + 1$ vértices então existem k vértices que não fazem parte do IS, mas sabemos que todos os vértices de $OH(H)$ tem grau 2 e que $OH(H)$ possui $2k + 1$ arestas, pois é um *Odd-Hole*. Sabemos também que os vértices do IS não possuem nenhuma aresta entre si e cada vértice que não faz parte do IS é incidido por duas arestas que podem fazê-los adjacentes a vértices do IS ou a outros vértices que não estão no IS. Se todas as arestas conectam vértices que estão no IS a vértices que não estão no IS, então o grafo tem $2k$ arestas. Se alguma das arestas conecta dois vértices que não estão no IS, então o grafo tem menos que $2k$ arestas. Como um *Odd-hole* tem $2k + 1$ arestas, significa que o grafo não é um *Odd-hole* e, portanto, temos uma contradição. Logo, não pode existir um IS maior que k em $OH(H)$ e a desigualdade é válida. \square

Proposição 3.14 *A desigualdade (3.31) representa uma faceta $F = \{x \in \text{conv}(P) : \sum_{i \in I(G)} \sum_{j \in OH(H)} y_{ij} = k\}$ de $\text{conv}(P) \cap \{x \in \mathbb{B}^{|V(G) \times V(H)|} : x_{ij} = 0 \forall i \notin I(G) \vee j \notin OH(H)\}$.*

Prova: A face definida pela desigualdade (3.31) é própria, pois o vetor nulo não pertence à face e pertence a $\text{conv}(P)$, e qualquer vetor que possua 0 (zero) em todas as posições exceto nas posições $i_1 j_1, i_2 j_2, \dots, i_k j_k : i_r \in I(G), r = 1 \dots k$ e $j_s \in OH(H), s = 1 \dots k, (j_p, j_q) \notin E(H) \forall p, q \in \{1 \dots k\}$ pertence à face.

Seja $F' = \{x \in \text{conv}(P) : \pi x = \pi_0\}$ e $F \subseteq F'$. Agora, pelo teorema de caracterização de facetas, se a desigualdade (3.31) representa uma faceta de $\text{conv}(P)$, então deve existir uma constante $\alpha \in \mathbb{R}$ e positiva que satisfaz:

$(\pi, \pi_0) = \alpha(\gamma, k)$ onde $\gamma_{ij} = 1 \forall i \in I(G)$ e $j \in OH(H)$. Ou seja, devemos mostrar que $\pi_0 = k\alpha$ e $\pi_{ij} = \alpha = \pi_0/k \forall i \in I(G)$ e $j \in OH(H)$.

Se a desigualdade (3.31) representa uma faceta, então deve ser possível chegar às relações acima utilizando o fato de que $F \subseteq F'$, assim, temos:

$$(1) \quad (\dots, 0, 1_{i_1 j_1}, 0, \dots, 0, 1_{i_2 j_2}, 0, \dots, 0, 1_{i_k j_k}, 0, \dots) \in F, \text{ onde } i_r \in I(G), r = 1 \dots k, j_s \in OH(H), s = 1 \dots k, (j_p, j_q) \notin E(H) \forall p, q \in \{1 \dots k\} \Rightarrow$$

$$(\dots, 0, 1_{i_1 j_1}, 0, \dots, 0, 1_{i_2 j_2}, 0, \dots, 0, 1_{i_k j_k}, 0, \dots) \in F' \Rightarrow \pi_{i_1 j_1} + \pi_{i_2 j_2} + \dots + \pi_{i_k j_k} = \pi_0 \forall i_r \in I(G), r = 1 \dots k, j_s \in OH(H), s = 1 \dots k, (j_p, j_q) \notin E(H).$$

Suponhamos uma ordem para os vértices de $OH(H)$ de forma que o primeiro vértice seja escolhido arbitrariamente e o vértice na posição p deve ser adjacente aos vértices nas posições $(p-1) \bmod (2k+1)$ e $(p+1) \bmod (2k+1)$.

Chamemos de posição ip do vetor que representa um ponto de P , a posição correspondente à variável que representa o mapeamento do vértice $i \in V(G)$ no vértice de $OH(H)$ que recebeu a posição p na ordem de $OH(H)$.

Os vértices de $I(G)$ serão chamados arbitrariamente de $i_0, i_1, i_2 \dots i_{|I(G)|-1}$.

Assim, de (1) sabemos que:

$$\pi_{i_0 j_0} + \pi_{i_1 j_3} + \dots + \pi_{i_{k-1} j_{2k-1}} = \pi_0 \wedge \pi_{i_0 j_1} + \pi_{i_1 j_3} + \dots + \pi_{i_{k-1} j_{2k-1}} = \pi_0 \Rightarrow \pi_{i_0 j_0} = \pi_{i_0 j_1}$$

$$\pi_{i_0 j_1} + \pi_{i_1 j_4} + \dots + \pi_{i_{k-1} j_{2k}} = \pi_0 \wedge \pi_{i_0 j_2} + \pi_{i_1 j_4} + \dots + \pi_{i_{k-1} j_{2k}} = \pi_0 \Rightarrow \pi_{i_0 j_1} = \pi_{i_0 j_2}$$

$$\pi_{i_0 j_{2k-1}} + \pi_{i_1 j_1} + \dots + \pi_{i_{k-1} j_{2k-3}} = \pi_0 \wedge \pi_{i_0 j_{2k}} + \pi_{i_1 j_1} + \dots + \pi_{i_{k-1} j_{2k-3}} = \pi_0 \Rightarrow$$

$$\pi_{i_0 j_{2k-1}} = \pi_{i_0 j_{2k}} \Rightarrow \pi_{i_0 j_0} = \pi_{i_0 j_1} = \pi_{i_0 j_2} = \dots = \pi_{i_0 j_{2k-1}} = \pi_{i_0 j_{2k}}$$

Também temos que:

$$\begin{aligned} \pi_{i_0j_1} + \pi_{i_1j_3} + \dots + \pi_{i_{k-1}j_{2k-1}} = \pi_0 \wedge \pi_{i_kj_1} + \pi_{i_1j_3} + \dots + \pi_{i_{k-1}j_{2k-1}} = \pi_0 &\Rightarrow \pi_{i_0j_1} = \pi_{i_kj_1} \\ \pi_{i_0j_1} + \pi_{i_1j_3} + \dots + \pi_{i_{k-1}j_{2k-1}} = \pi_0 \wedge \pi_{i_0j_1} + \pi_{i_kj_3} + \dots + \pi_{i_{k-1}j_{2k-1}} = \pi_0 &\Rightarrow \pi_{i_1j_3} = \pi_{i_kj_3} \end{aligned}$$

Como podemos obter o mesmo resultado que obtivemos para os $\pi_{i_0j_p}$, com $p = 0 \dots 2k$ para os $\pi_{i_rj_p}$, com $p = 0 \dots 2k$, $r = 1 \dots |I(G)| - 1$, temos que $\pi_{i_0j_p} = \pi_{i_1j_q} = \dots = \pi_{i_{|I(G)|-1}j_r} = \pi_0/k$, com $p, q, r = 0 \dots 2k$ e $i_0, i_1, \dots, i_{|I(G)|-1} \in I(G)$. \square

Assim como no caso da desigualdade (3.27), neste trabalho não investigamos quais seriam as condições necessárias e suficientes para que (3.31) defina uma faceta de $\text{conv}(P)$ no caso geral. Possivelmente, operações de *lifting* poderão ser aplicadas mostrando que a face definida por esta desigualdade ainda tem dimensão muito baixa.

Podemos chegar a provas e conclusões similares para as desigualdades (3.32), (3.33), (3.34).

Capítulo 4

O Problema da Clique Máxima

No Capítulo 2 foi dito que o **MSIC** é comumente resolvido através de sua redução para o problema da Clique Máxima (de agora em diante será referido apenas como o problema da Clique ou, simplesmente, Clique, quando o contexto tornar claro que estamos nos referindo ao problema e não à estrutura). Neste capítulo, definiremos o problema da clique e mostraremos um modelo **PLI** deste problema. Em seguida, faremos uma comparação entre os modelos para o problema da clique e o modelo do **MSIC**, demonstraremos que estes modelos são equivalentes e que algumas das desigualdades apresentadas no Capítulo 3 são equivalentes a desigualdades conhecidas no modelo para o problema da Clique.

O problema da Clique pode ser definido da seguinte maneira:

Entrada: Um grafo G .
Pergunta: encontre um subgrafo G' em G que seja completo e cujo número de vértices seja máximo.

Este capítulo está organizado da seguinte forma: na Seção 4.1 apresentaremos um modelo **PLI** bem conhecido para este problema. Na Seção 4.2 mostraremos como o **MSIC** pode ser reduzido ao problema da Clique e mostraremos as relações existentes entre as desigualdades do modelo do **MSIC** e as desigualdades do modelo da Clique e, então, mostraremos que os dois modelos são de fato equivalentes. Por fim, na Seção 4.3 falaremos sobre o algoritmo utilizado em um programa chamado Cliquer que resolve o problema da Clique.

4.1 Modelo PLI

Em [25] são apresentadas algumas formulações para o problema do IS máximo, que é o equivalente ao problema da Clique no grafo complementar. Aqui, reproduzimos apenas a mais forte delas, adicionada de desigualdades *Odd-Antihole* e fazendo as modificações

necessárias de modo que a formulação aqui apresentada modele o problema da Clique. Note-se ainda que, a formulação apresentada em [25] contempla o caso mais geral em que os vértices possuem pesos, enquanto que aqui consideramos que todos os vértices têm peso 1 (um). Outros tipos de desigualdades para o problema do IS máximo e, portanto, da Clique podem ser encontradas em [3].

A formulação para o problema da Clique é apresentada a seguir:

$$\max \quad \sum_{i \in V(G)} y_i \quad (4.1)$$

$$\sum_{i \in I(G)} y_i \leq 1, \quad (4.2)$$

\forall conjunto independente $I(G) \in G$

$$\sum_{i \in OA(G)} y_i + \sum_{i \notin OA(G)} \alpha_i y_i \leq k, \quad (4.3)$$

\forall *Odd-Antihole* $OA(G) \in G, |OA(G)| = 2k + 1$

$$\sum_{i \in OH(G)} y_i + \sum_{i \notin OH(G)} \alpha_i y_i \leq 2, \quad (4.4)$$

\forall *Odd-Hole* $OH(G) \in G$

$$y_i \in \{0, 1\} \quad \forall i \in V(G) \quad (4.5)$$

Neste modelo a variável y_i assume valor 1 (um) se o vértice i faz parte da clique máxima dada como solução e valor 0 (zero), caso contrário. Como o objetivo é encontrar a maior clique em G , a função objetivo determina que a soma das variáveis deve ser maximizada.

A resolução deste problema deve dar como resposta uma clique máxima para o grafo G , portanto, todos os vértices que fazem parte da solução são adjacentes, ou seja, não podem haver dois vértices não-adjacentes na solução do problema. Traduzindo isso em termos das variáveis do modelo apresentado, significa que a soma de duas variáveis correspondentes a vértices não-adjacentes deve ser menor ou igual a 1 (um). Generalizando essa idéia, temos que a soma das variáveis correspondentes a vértices que fazem parte do mesmo IS deve ser menor ou igual a 1 (um). A desigualdade (4.2) determina exatamente isso. É fácil provar que aquela desigualdade define uma faceta se, e somente se, $I(G)$ é um IS maximal.

Nas desigualdades (4.3) e (4.4) existem somatórios de termos da forma $\alpha_i y_i$. Estes somatórios representam *liftings* sobre as desigualdades $\sum_{i \in OA(G)} y_i \leq k$ e $\sum_{i \in OH(G)} y_i \leq 2$, onde os α_i representam os coeficientes atribuídos às variáveis pelo *lifting*. O termo correspondente ao *lifting* dessas desigualdades é apresentado no modelo porque, no caso geral, é fácil mostrar que aquelas desigualdades não definem facetas sem ele.

A desigualdade $\sum_{i \in OA(G)} y_i \leq k$ define que a soma das variáveis correspondentes aos vértices que fazem parte de um *Odd-Antihole* de tamanho $2k + 1$ deve ser menor ou igual a k , ou seja, em um *Odd-Antihole* de tamanho $2k + 1$ a maior clique possui tamanho

k . Este fato pode ser provado de maneira semelhante à prova dada para a validade da desigualdade (3.31). Portanto, como a desigualdade (4.3) representa um *lifting* da desigualdade $\sum_{i \in OA(G)} y_i \leq k$, ela também é válida.

Se tivermos um *Odd-Hole* em G , é obvio que a maior clique que pode existir nessa estrutura tem tamanho 2 e é exatamente isso que diz a desigualdade $\sum_{i \in OH(G)} y_i \leq 2$. Portanto, essa desigualdade é, trivialmente, válida. Como a desigualdade (4.4) descreve um *lifting* daquela desigualdade, esta última também é válida. A forma como os coeficientes α_i são computados nos procedimentos de *lifting* não será discutida neste trabalho. Uma discussão sobre como isso é feito pode ser encontrada em [25].

Além dessas desigualdades utilizadas na formulação acima, existem diversas outras que podem ser encontradas em [3], no entanto, as desigualdades aqui apresentadas são as que interessam aos propósitos deste trabalho por possuírem uma relação com as desigualdades apresentadas para o MSIC no Capítulo 3.

4.2 MSIC versus Clique

Como sabemos, qualquer problema em \mathcal{NP} pode ser reduzido em tempo polinomial a um problema \mathcal{NP} -difícil. No caso do MSIC essa estratégia é bastante utilizada para resolvê-lo. Em especial ele é reduzido ao problema da Clique através da construção de um grafo conhecido como **Grafo Associação**.

Um grafo associação $A = (V(A), E(A))$ construído a partir dos grafos $G = (V(G), E(G))$ e $H = (V(H), E(H))$ onde $V(A) = (V(G) \times V(H))$ e $((i, j), (k, l)) \in E(A)$ se $(i, k) \in E(G)$ e $(j, l) \in E(H)$ ou, $(i, k) \notin E(G)$ e $(j, l) \notin E(H)$. Ou seja, o conjunto de vértices do grafo associação é um conjunto de pares de vértices onde um dos vértices pertence ao grafo G e o outro ao grafo H . O significado dessa construção é que cada vértice do grafo associação corresponde ao mapeamento de um vértice de G em um vértice de H . O conjunto de arestas do grafo associação é construído de forma que uma aresta só existe entre dois vértices de A se os ambos os vértices originais do grafo G e os vértices originais do grafo H forem adjacentes, ou se ambos forem não-adjacentes. Colocando de outra forma, dois vértices de A são adjacentes se os dois mapeamentos dos vértices de G nos vértices de H puderem ocorrer ao mesmo tempo de acordo com a definição do MSIC. A Figura 4.1 mostra um exemplo de grafo associação construído a partir dos dois grafos G e H dados na figura. A prova de que a solução do problema da Clique para o grafo associação resulta em uma solução para o MSIC e, portanto, é uma redução válida, pode ser encontrada em [8] *apud* [11] e aqui apresentamos uma prova independente.

Proposição 4.1 *Dados dois grafos G e H e um grafo associação A , construído a partir destes como descrito acima, existe uma correspondência entre cliques de A e pares de*

subgrafos induzidos isomorfos de G e H em que os subgrafos possuem o mesmo tamanho das cliques. Isto é, para toda clique C em A tal que $|V(C)| = n$, existe um par de subgrafos induzidos isomorfos G' e H' , respectivamente de G e H , associado a ela tal que $|V(G')| = |V(H')| = n$. Também, para todo par de subgrafos induzidos isomorfos G' e H' , respectivamente de G e H , tal que $|V(G')| = |V(H')| = n$, existe uma clique C em A associada a ele, com $|V(C)| = n$.

Prova:

Seja C uma clique em A com $|V(C)| = n$. Então, pela construção do grafo associação, sabemos que cada vértice de A está associado a um par de vértices de G e H e também que se dois vértices (i, j) e (k, l) são adjacentes, então, $i \neq j$ e $k \neq l$. Portanto, cada vértice de C está associado a um vértice distinto de G e da mesma forma, cada vértice de C está associado a um vértice distinto de H . Logo, C está associada a dois subgrafos induzidos G' e H' , respectivamente, de G e H ambos contendo n vértices.

Agora, suponha por contradição que G' e H' não são isomorfos, logo, não é possível encontrar um mapeamento válido entre os vértices destes dois grafos. Considere o mapeamento sugerido pelos vértices de C , isto é, se $(i, j) \in V(C)$, então, considere o mapeamento do vértice i de G' no vértice j de H' . Se G' e H' não são isomorfos, então, para algum par de mapeamentos (i, j) e (k, l) , devemos ter que $(i, k) \in E(G)$ e $(j, l) \notin E(H)$ ou que $(i, k) \notin E(G)$ e $(j, l) \in E(H)$. Contudo, em qualquer uma dessas duas situações, os vértices (i, j) e (k, l) de A não seriam adjacentes. Porém, como esses vértices fazem parte de C , sabemos que eles são adjacentes e, portanto, chegamos a um absurdo.

Sejam G' e H' dois subgrafos induzidos isomorfos, respectivamente de G e H , tais que $|V(G')| = |V(H')| = n$. Então, existe um mapeamento entre os vértices de G' e H' no qual um par de vértices i e k de G' é mapeado em um vértice j e l de H' se $(i, k) \in E(G')$ e $(j, l) \in E(H')$ ou $(i, k) \notin E(G')$ e $(j, l) \notin E(H')$. Na construção do grafo associação, um vértice (i, j) é adjacente a um vértice (k, l) se $(i, k) \in E(G)$ e $(j, l) \in E(H)$ ou $(i, k) \notin E(G)$ e $(j, l) \notin E(H)$. Assim, é fácil perceber que os vértices de A correspondentes ao mapeamento dos vértices de G' nos vértices de H' definem uma clique C com $|V(C)| = n$.

□

Corolário 4.1 *Dados G e H , entrada do MSIC, e A , o grafo associação obtido a partir daqueles dois grafos, a solução ótima do problema da Clique Máxima tendo A como entrada dá uma solução ótima para o problema do MSIC.*

Prova: Pela Proposição 4.1, sabemos que toda clique em A possui um par de subgrafos induzidos isomorfos de G e H com o mesmo número de vértices associado. Logo, a solução do problema da Clique dá uma solução para o MSIC e esta solução é ótima pois, pela

mesma proposição, todo par de subgrafos induzidos isomorfos de G e H está associado a uma clique com o mesmo número de vértices em A . Portanto, se a solução não for ótima para o **MSIC**, então, a solução do problema da Clique também não pode ser ótima. \square

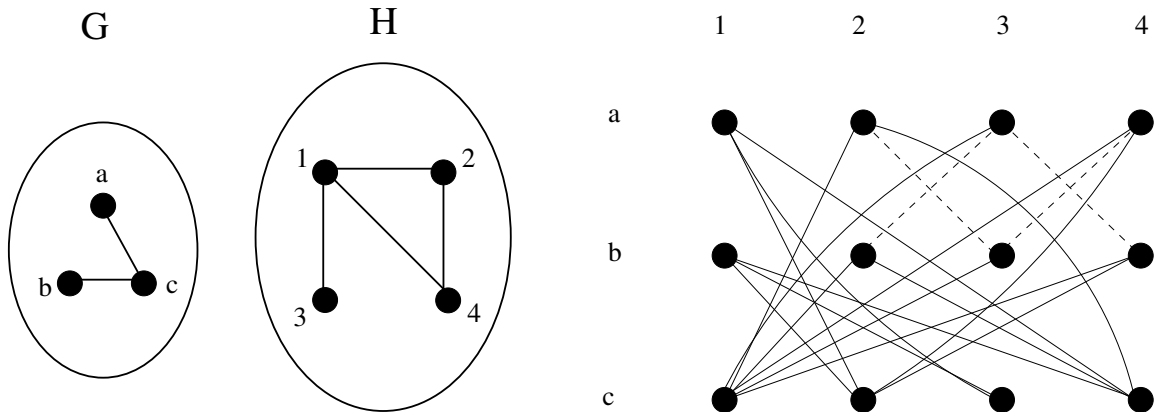


Figura 4.1: Grafo associação construído a partir de dois grafos. As arestas segmentadas no grafo associação representam arestas obtidas pelo mapeamento de pares de vértices não-adjacentes dos dois grafos.

Podemos notar que o modelo **PLI** apresentado para o problema da Clique, quando aplicado a um grafo associação possui $|V(G)| \times |V(H)|$ variáveis e, portanto estamos trabalhando em um espaço de mesma dimensão que no modelo para o **MSIC**. Além disso, o significado implícito de cada variável nos dois casos é o mesmo, pois o grafo associação foi construído dessa forma. Em outras palavras, estamos dizendo que em ambos os modelos a variável y_{ij} recebe valor 1 se o vértice $i \in G$ é mapeado no vértice $j \in H$ e, como sabemos que a redução é válida, podemos chegar à conclusão que os modelos são equivalentes (no sentido que estão definidos exatamente sobre o mesmo conjunto de variáveis).

Agora que sabemos que os dois modelos são equivalentes, podemos comparar as desigualdades que apresentamos para ambos e, com isso, chegamos à conclusão que algumas dessas desigualdades estão relacionadas como provamos a seguir:

Proposição 4.2 *O conjunto das desigualdades obtidas a partir da inequação (3.3) (página 15) é um subconjunto das desigualdades obtidas a partir da equação (4.2).*

Prova: Para provar que (3.3) define um subconjunto de (4.2), precisamos mostrar que qualquer subgrafo do grafo associação obtido a partir dos subgrafos definidos pelas variáveis que possuem coeficiente 1 (um) em (3.3) é um IS.

Assuma com o propósito de obter uma contradição que os vértices do grafo associação, A , correspondentes às variáveis que possuem coeficiente 1 (um) na desigualdade (3.3) não formam um IS. Portanto, existem dois vértices adjacentes em A dentre os vértices

considerados, chamemos esses vértices de (i, j) e (k, l) (lembre-se que um vértice do grafo associação é um par de vértices que representa o mapeamento de um vértice de um dos grafos em um vértice do outro grafo), ou seja, a aresta $((i, j), (k, l))$ existe no grafo associação. Porém, pela forma que o grafo associação é construído, esses vértices só podem ser adjacentes se i é adjacente a k em G e j é adjacente a l em H , ou se i é não-adjacente a k em G e j é não-adjacente a l em H . Mas, pela definição do **MSIC** se essa situação ocorre, então, ambas as variáveis y_{ij} e y_{kl} podem assumir valor 1 (um) simultaneamente. Todavia, se isso é verdade, então a desigualdade dada por (3.3) não é válida e, portanto, chegamos a uma contradição. Logo, o grafo associação, obtido a partir dos vértices correspondentes às variáveis que possuem coeficiente 1 (um) em (3.3) é um IS e a proposição está provada. \square

Note que, como a desigualdade (3.3) é um *lifting* de (2.2) (página 5), se as variáveis que possuem coeficiente 1 (um) em (3.3) formam um IS no grafo associação, então as variáveis que possuem coeficiente 1 (um) em (2.2) também formam um IS naquele grafo. Portanto, (2.2) também é uma desigualdade do tipo (4.2).

Proposição 4.3 *O conjunto das desigualdades obtidas a partir da inequação (3.19) (página 29) é um subconjunto das desigualdades obtidas a partir da equação (4.2).*

Prova: Para provar que (3.19) define um subconjunto de (4.2), precisamos mostrar que qualquer subgrafo grafo associação obtido a partir dos subgrafos definidos pelas variáveis que possuem coeficiente 1 (um) em (3.19) é um IS.

Assuma para o propósito de obter uma contradição que os vértices do grafo associação, A , correspondentes às variáveis que possuem coeficiente 1 (um) na desigualdade (3.19) não formam um IS. Portanto, existem dois vértices adjacentes em A dentre os vértices considerados. Chamemos esses vértices de (i, j) e (k, l) e, assim, a aresta $((i, j), (k, l))$ existe no grafo associação. Porém, pela forma que o grafo associação é construído, esses vértices só podem ser adjacentes se i é adjacente a k em G e j é adjacente a l em H , ou se i é não-adjacente a k em G e j é não-adjacente a l em H . Mas, pela definição do **MSIC**, se essa situação ocorre, então, ambas as variáveis y_{ij} e y_{kl} podem assumir valor 1 (um) simultaneamente. Todavia, se isso é verdade, então a desigualdade dada por (3.19) não é válida e, deste modo, chegamos a uma contradição. Logo, o grafo associação, obtido a partir dos vértices correspondentes às variáveis que possuem coeficiente 1 (um) nessa desigualdade, é um IS e a proposição está provada. \square

Assim como no caso anterior envolvendo as desigualdades (3.3) e (2.2), a desigualdade (3.19) é um *lifting* das desigualdades (3.15) (página 24) e (3.11) (página 22). Deste modo, estas duas últimas são também desigualdades do tipo (4.2).

Proposição 4.4 *O conjunto das desigualdades obtidas a partir da inequação (3.31) (página 38) é um subconjunto das desigualdades obtidas a partir da equação (4.3).*

Prova: Para provar que a desigualdade (3.31) define um subconjunto de (4.3), precisamos mostrar que o subgrafo do grafo associação obtido a partir dos subgrafos induzidos de G e H cujos vértices correspondentes às variáveis que possuem coeficiente 1 (um) naquela desigualdade, possui um *Odd-Antihole* de tamanho $2k + 1$, sendo que este também é o tamanho do *Odd-Hole* em (3.31).

A prova é dada por construção. Em (3.31), sejam, respectivamente, c_0, c_1, \dots, c_k os vértices de $I(G)$ e h_0, h_1, \dots, h_{2k} os vértices de $OH(H)$ na ordem do *Odd-Hole*, ou seja, h_i é adjacente apenas a h_{i-1} e $h_{(i+1) \bmod (2k+1)}$ para todo $i = 1 \dots 2k$.

Para facilitar o entendimento da prova, exibe-se a Figura 4.2 que mostra um *Odd-Antihole* no grafo associação construído a partir de dois grafos que contém, respectivamente, um IS e um *Odd-Hole*.

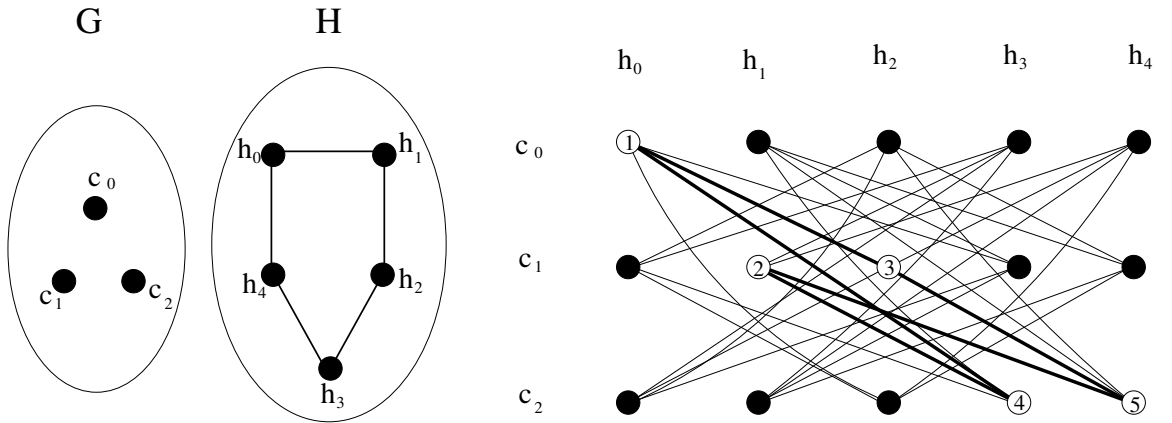


Figura 4.2: Grafo associação que contém *Odd-Antihole*. Os números dentro dos vértices do grafo associação mostram a ordem do *Odd-Antihole*. As arestas em negrito são as arestas do *Odd-Antihole*.

Então, construindo o grafo associação para $I(G)$ e $OH(H)$, temos que os vértices (c_i, h_{2i}) são não-adjacentes aos vértices (c_i, h_{2i-1}) e $(c_{(i+1) \bmod k}, h_{(2i+1) \bmod (2k+1)})$ para $i = 1 \dots k$ (na Figura 4.2 o vértice (c_1, h_2) é não-adjacente a (c_1, h_1) e (c_2, h_3) , o vértice (c_2, h_4) é não-adjacente a (c_2, h_5) e (c_0, h_0)) e (c_0, h_0) não é adjacente a (c_1, h_1) e (c_k, h_{2k}) (na Figura 4.2 o vértice (c_0, h_0) não é adjacente a (c_1, h_1) e (c_2, h_4)). Além disso, os vértices (c_i, h_{2i}) são adjacentes aos vértices (c_j, h_l) onde: $i = 1 \dots k, j = 0 \dots k, l = 0 \dots k$, e $((j \neq i) \vee (l \neq 2i - 1)) \wedge ((j \neq i + 1) \vee (l \neq (2i + 1) \bmod (2k + 1)))$ (na Figura 4.2 o vértice (c_1, h_2) é adjacente a (c_0, h_0) e (c_2, h_4) e o vértice (c_2, h_4) é adjacente a (c_1, h_1) e (c_1, h_2)) e os vértices (c_i, h_{2i-1}) são adjacentes aos vértices (c_j, h_l) para $i = 1 \dots k, j = 0 \dots k$,

$l = 0 \dots k$ e $((j \neq i - 1) \vee (l \neq 2i - 2)) \wedge ((j \neq i) \vee (l \neq 2i))$ (na Figura 4.2 o vértice (c_1, h_1) é adjacente a (c_2, h_3) e (c_2, h_4) e o vértice (c_2, h_3) é adjacente a (c_0, h_0) e (c_1, h_1)).

Assim, os vértices considerados acima formam um *Odd-Antihole* de tamanho $2k + 1$ e, adicionalmente, existem outras variáveis além das variáveis do *Odd-Antihole* que possuem coeficiente 1 (um) (correspondendo ao somatório de $\alpha_i y_i$ em (4.3)). Portanto, a desigualdade é do tipo (4.3). \square

Proposição 4.5 *O conjunto das desigualdades obtidas a partir da inequação (3.27) (página 38) é um subconjunto das desigualdades obtidas a partir da equação (4.4).*

Prova: Para provar que a desigualdade (3.27) define um subconjunto de (4.4), precisamos mostrar que o grafo associação obtido a partir dos subgrafos induzidos de G e H cujos vértices correspondentes às variáveis que possuem coeficiente 1 (um) naquela desigualdade, possui um *Odd-Hole* de tamanho $2k + 1$ sendo que este também é o tamanho do *Odd-Hole* em (3.27).

A prova é dada por construção. Em (3.27), sejam c_0, c_1, \dots, c_p , $p \geq 2$, os vértices de $C(G)$ e h_0, h_1, \dots, h_{2k} os vértices de $OH(H)$ na ordem do *Odd-Hole*, ou seja, h_i é adjacente apenas a h_{i-1} e $h_{(i+1) \bmod (2k+1)}$ para todo $i = 1 \dots 2k$.

Para facilitar o entendimento da prova, exhibe-se a Figura 4.3 que mostra um *Odd-Hole* no grafo associação construído a partir de dois grafos que contém respectivamente uma Clique e um *Odd-Hole*.

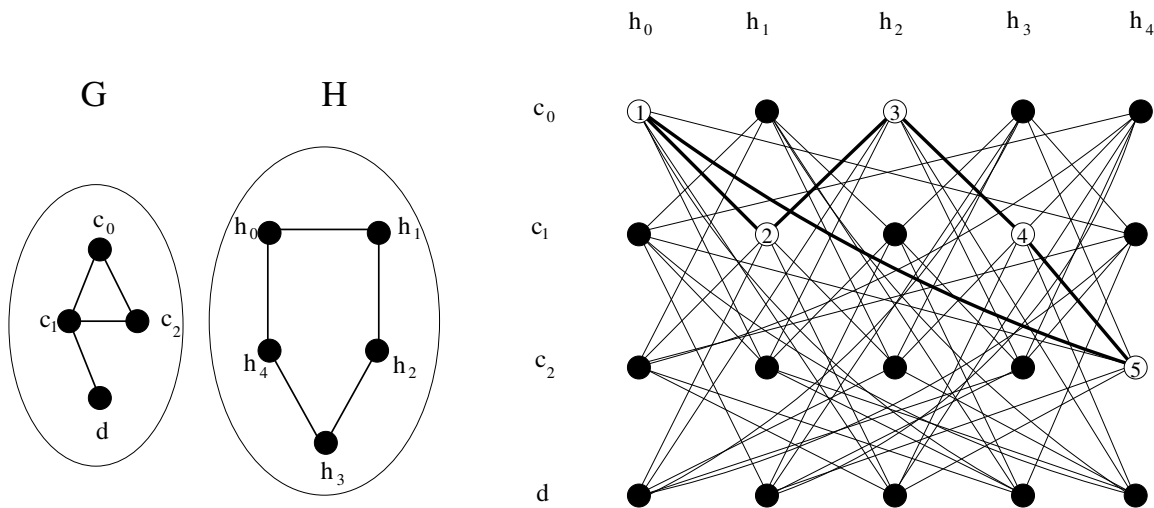


Figura 4.3: Grafo associação que contém *Odd-Hole*. Os números dentro dos vértices do grafo associação mostram a ordem do *Odd-Hole*. As arestas em negrito são as arestas do *Odd-Hole*.

Então, construindo o grafo associação para $C(G)$ e $OH(H)$, temos que os vértices (c_0, h_{2i-1}) são adjacentes aos vértices (c_1, h_{2i-2}) e (c_1, h_{2i}) para $i = 2 \dots k$ (na Figura 4.3 o vértice (c_0, h_2) é adjacente a (c_1, h_1) e (c_1, h_3)) e (c_0, h_0) é adjacente a (c_1, h_1) e (c_2, h_{2k}) (na Figura 4.3 o vértice (c_0, h_0) é adjacente a (c_1, h_1) e (c_2, h_4)), sendo este último também adjacente (c_1, h_{2k-1}) (na Figura 4.3 o vértice (c_2, h_4) é adjacente a (c_1, h_3)). Além disso, todas as demais possíveis adjacências entre estes vértices são inexistentes.

Assim, os vértices considerados acima formam um *Odd-Hole* de tamanho $2k + 1$ e adicionalmente, existem outras variáveis além das variáveis do *Odd-Hole* que possuem coeficiente 1 (um) (correspondendo ao somatório de $\alpha_i y_i$ em (4.4)). Portanto, a desigualdade é do tipo (4.4). \square

4.3 Cliquer

Existem vários algoritmos para resolver o problema da Clique, dentre estes algoritmos há o algoritmo utilizado no programa chamado Cliquer [40]. Este algoritmo é simples, possui implementação disponível *on-line* e se mostrou bastante rápido nos experimentos realizados com os grafos associação correspondentes às instâncias de teste geradas para o **MSIC**. Estas características nos levaram a estudar o algoritmo do Cliquer mais detalhadamente. Deste momento em diante, faremos um abuso de linguagem e não mais distinguiremos entre o algoritmo utilizado pelo Cliquer e o programa propriamente dito.

O Cliquer é baseado em um algoritmo simples de *backtrack* em que a cada nó da árvore de recursão tenta-se aumentar o tamanho da clique atual adicionando-se a ela algum dos vértices não presentes na mesma, porém, adjacente a todos os seus vértices (os chamados vértices candidatos). O algoritmo prevê que, no início, os vértices estejam ordenados de alguma forma. O resultado é a construção de uma árvore de enumeração em que a cada nó é feita a escolha de adicionar ou não adicionar o vértice em questão à clique atual.

Obviamente, o algoritmo de *backtrack* realiza podas na árvore de enumeração evitando a necessidade de explorá-la completamente, pois, de outra forma, ele seria completamente inviável, exceto para instâncias extremamente pequenas. O algoritmo original poda uma subárvore toda vez que o tamanho da clique encontrada naquela subárvore adicionado do número de vértices candidatos for menor ou igual ao tamanho da maior clique já encontrada, isto é, quando se tem certeza que aquela subárvore não tem possibilidade de gerar uma solução melhor que a já encontrada.

Segundo [40], apesar de bastante simples, o algoritmo que acabamos de descrever era, na época em que o artigo foi escrito, o melhor algoritmo conhecido para grafos esparsos.

Como dissemos anteriormente, o Cliquer se baseia neste algoritmo, no entanto, ele

divide o problema original em subproblemas, da seguinte forma. Suponha um grafo com n vértices para o qual se deseja encontrar uma clique de tamanho máximo. Ordenamos os vértices por algum critério e consideramos o problema de encontrar uma clique máxima no subgrafo contendo apenas o n -ésimo vértice da ordem (trivialmente, a solução é uma clique de tamanho 1 (um)). Em seguida consideramos o problema de encontrar uma clique máxima em um subgrafo contendo apenas o último e o penúltimo vértices da ordem, então, sabemos que a clique máxima terá tamanho 1 (um) ou 2 (dois). Repetindo este processo temos que quando considerarmos o subproblema que contém o vértice j , a clique máxima deste subproblema terá tamanho $c(j + 1)$ ou $c(j + 1) + 1$, onde, $c(j + 1)$ é o tamanho da clique máxima no subproblema que contém os vértices $j + 1$ até n . O processo continua até que os n subproblemas sejam resolvidos, sendo que o n -ésimo subproblema é igual ao problema original.

Se o Cliquer fosse executado da forma como foi descrito acima o resultado seria, obviamente, um algoritmo mais lento que o original. No entanto, é fácil perceber que a informação sobre o tamanho da clique no subproblema anterior ($c(j + 1)$) pode ser usada para fazer uma nova poda na árvore de enumeração e, de fato, é isso que o Cliquer faz. Assim, uma poda é realizada quando a condição de poda do algoritmo original for satisfeita, ou quando o tamanho da clique encontrada em uma subárvore adicionado de $c(i)$ for menor que o tamanho da maior clique já encontrada, onde i é o menor índice na ordem dos vértices dentre os vértices candidatos na subárvore.

Na descrição dos algoritmos falamos algumas vezes em ordenação dos vértices mas, até o momento, não tínhamos discutido como essa ordenação é feita. Contudo, o desempenho dos algoritmos dependem da escolha de um bom critério para ordenação dos vértices. No algoritmo original, os vértices são ordenados por ordem crescente do grau no subgrafo induzido pelos vértices candidatos. Já o Cliquer, provê diferentes formas de ordenação: nenhuma ordenação, ordem reversa a que os vértices aparecem no arquivo de entrada, ordenação por grau, ordenação por cores e ordenação randômica. Por padrão a ordenação por cores é utilizada. Nesta ordenação um conjunto independente maximal é encontrado considerando-se primeiramente os vértices de maior grau e a este conjunto de vértices é atribuída uma cor. O procedimento é repetido no subgrafo induzido pelos vértices não-coloridos até que não restem mais vértices não-coloridos. Os vértices são considerados pelo Cliquer na ordem reversa em que são coloridos.

Nos experimentos computacionais apresentados em [40] que comparam o Cliquer com outros cinco algoritmos conhecidos na literatura utilizando grafos randômicos de diferentes tamanhos e densidades e alguns grafos de *benchmark* do DIMACS, o Cliquer se mostra competitivo, apresentando, em particular, um desempenho melhor que o algoritmo de *backtrack* original em quase todos os casos testados. Além disso, seu desempenho foi muito bom para grafos randômicos esparsos.

Capítulo 5

Implementação e Resultados Computacionais

Nos dois capítulos anteriores apresentamos um estudo teórico sobre o problema do **MSIC** e uma comparação com o problema da Clique Máxima. Neste capítulo mostraremos como as informações obtidas naqueles capítulos foram utilizadas para o desenvolvimento de um algoritmo de **PLI** para resolver o problema do **MSIC** e os resultados que foram alcançados. O presente capítulo será dividido em duas partes. Na primeira discutiremos a implementação do algoritmo enquanto na segunda mostraremos e compararemos os resultados computacionais obtidos por ele com diversos parâmetros diferentes e também faremos uma comparação com o Cliquer.

5.1 Implementação

Nesta seção discutiremos a implementação das principais partes do algoritmo de **B&B** e **B&C** para o **MSIC**. Iniciaremos com a implementação das formulações; a seguir passaremos para as estratégias de *lifting*; rotinas de separação; estratégias de seleção de variáveis; estratégia de ramificações múltiplas e, por fim, estratégias de ramificação ao estilo do que é feito no Cliquer.

Como ao longo desta seção serão feitas muitas referências à desigualdades apresentadas nos Capítulos 2 e 3, por conveniência, re-apresentamos a seguir estas desigualdades:

$$\sum_{i \in V(G)} y_{ij} \leq 1, \quad \forall j \in V(H) \quad [2.2]$$

$$\sum_{j \in V(H)} y_{ij} \leq 1, \quad \forall i \in V(G) \quad [2.3]$$

$$y_{ij} + y_{kj} + y_{il} + y_{kl} \leq 1, \quad \forall (i, k) \in E(G), \forall (l, j) \notin E(H) \quad [2.4]$$

$$y_{ij} + y_{kj} + y_{il} + y_{kl} \leq 1, \quad \forall (i, k) \notin E(G), \forall (l, j) \in E(H) \quad [2.5]$$

$$\begin{aligned} \sum_{i \in V(G)} y_{ij} + \sum_{l \in C(j) \setminus \{j\}} \sum_{k \in D_G} y_{kl} + \sum_{l \in I(j) \setminus \{j\}} \sum_{k \in U_G} y_{kl} \leq 1, \\ \forall j \in V(H), \forall C(j), I(j) \in H \end{aligned} \quad [3.3]$$

$$\begin{aligned} \sum_{j \in V(H)} y_{ij} + \sum_{l \in D_H} \sum_{k \in C(i) \setminus \{i\}} y_{kl} + \sum_{l \in U_H} \sum_{k \in I(i) \setminus \{i\}} y_{kl} \leq 1, \\ \forall i \in V(G), \forall C(i), I(i) \in G \end{aligned} \quad [3.4]$$

$$\sum_{i \in V(G)} y_{ij} + \sum_{l \in Adj(j)} y_{kl} \leq 1,$$

$$\forall j \in V(H), \forall D_G, \text{ com } D_G = \{k\} \quad [3.7]$$

$$\sum_{i \in V(G)} y_{ij} + \sum_{l \notin Adj(j)} y_{kl} \leq 1,$$

$$\forall j \in V(H), \forall U_G, \text{ com } U_G = \{k\} \quad [3.8]$$

$$\sum_{j \in V(H)} y_{ij} + \sum_{k \in Adj(i)} y_{kl} \leq 1,$$

$$\forall i \in V(G), \forall D_H, \text{ com } D_H = \{l\} \quad [3.9]$$

$$\sum_{j \in V(H)} y_{ij} + \sum_{k \notin Adj(i)} y_{kl} \leq 1,$$

$$\forall i \in V(G), \forall U_H, \text{ com } U_H = \{l\} \quad [3.10]$$

$$\sum_{i \in C} \sum_{j \in I} y_{ij} \leq 1, \quad \forall \text{ clique maximal } C \in G, \forall \text{ IS maximal } I \in H \quad [3.11]$$

$$\sum_{i \in I} \sum_{j \in C} y_{ij} \leq 1, \quad \forall \text{ IS maximal } I \in G, \forall \text{ clique maximal } C \in H \quad [3.12]$$

$$\begin{aligned} \sum_{h=0}^{\min(|SC(G)|, |SI(H)|)-1} \sum_{i \in C(G_h)} \sum_{j \in I(H_h)} y_{ij} \leq 1, \\ \forall SC(G), \forall SI(H) \end{aligned} \quad [3.15]$$

$$\sum_{h=0}^{\min(|\mathcal{SC}(H)|, |\mathcal{SI}(G)|)-1} \sum_{i \in I(G_h)} \sum_{j \in C(H_h)} y_{ij} \leq 1, \quad \forall \mathcal{SI}(G), \forall \mathcal{SC}(H) \quad [3.16]$$

$$\begin{aligned} & \sum_{h=0}^{\min(|\mathcal{SC}(G)|, |\mathcal{SI}(H)|)-1} \sum_{i \in C(G_h)} \sum_{j \in I(H_h)} y_{ij} + \sum_{k \in MC(G)} \sum_{l \in MI(H)} y_{kl} + \sum_{k \in MC(G)} \sum_{l \in CI(R^H)} y_{kl} + \\ & + \sum_{k \in Is(R^G)} \sum_{l \in MI(H)} y_{kl} + \sum_{h \in mci(G) : \{k\}=C(G_h)} \sum_{l \in F(H_h)} y_{kl} + \sum_{h \in mii(H) : \{l\}=I(H_h)} \sum_{k \in F(G_h)} y_{kl} \leq 1, \\ & \forall \mathcal{SC}(G), \forall \mathcal{SI}(H) \quad [3.19] \end{aligned}$$

$$\begin{aligned} & \sum_{h=0}^{\min(|\mathcal{SI}(G)|, |\mathcal{SC}(H)|)-1} \sum_{i \in I(G_h)} \sum_{j \in C(H_h)} y_{ij} + \sum_{k \in MI(G)} \sum_{l \in MC(H)} y_{kl} + \sum_{k \in MI(G)} \sum_{l \in Is(R^H)} y_{kl} + \\ & + \sum_{k \in CI(R^G)} \sum_{l \in MC(H)} y_{kl} + \sum_{h \in mii(G) : \{k\}=I(G_h)} \sum_{l \in F(H_h)} y_{kl} + \sum_{h \in mci(H) : \{l\}=C(H_h)} \sum_{k \in F(G_h)} y_{kl} \leq 1, \\ & \forall \mathcal{SC}(H), \forall \mathcal{SI}(G) \quad [3.20] \end{aligned}$$

$$\sum_{i \in C(G)} \sum_{j \in OH(H)} y_{ij} \leq 2,$$

$$\forall C(G), |C(G)| \geq 3, \forall OH(H), |OH(H)| = 2k + 1, k > 1 \quad [3.27]$$

$$\sum_{i \in OH(G)} \sum_{j \in C(H)} y_{ij} \leq 2,$$

$$\forall OH(G), |OH(G)| = 2k + 1, k > 1, \forall C(H), |C(H)| \geq 3 \quad [3.28]$$

$$\sum_{i \in I(G)} \sum_{j \in OA(H)} y_{ij} \leq 2,$$

$$\forall I(G), |I(G)| \geq 3, \forall OA(H), |OA(H)| = 2k + 1, k > 1 \quad [3.29]$$

$$\sum_{i \in OA(G)} \sum_{j \in I(H)} y_{ij} \leq 2,$$

$$\forall OA(G), |OA(G)| = 2k + 1, k > 1, \forall I(H), |I(H)| \geq 3 \quad [3.30]$$

$$\sum_{i \in I(G)} \sum_{j \in OH(H)} y_{ij} \leq k,$$

$$\forall I(G), |I(G)| > k, \forall OH(H), |OH(H)| = 2k + 1, k > 1 \quad [3.31]$$

$$\sum_{i \in OH(G)} \sum_{j \in I(H)} y_{ij} \leq k,$$

$$\forall OH(G), |OH(G)| = 2k + 1, k > 1, \forall I(H), |I(H)| > k \quad [3.32]$$

$$\sum_{i \in C(G)} \sum_{j \in OA(H)} y_{ij} \leq k,$$

$$\forall C(G), |C(G)| > k, \forall OA(H), |OA(H)| = 2k + 1, k > 1 \quad [3.33]$$

$$\sum_{i \in OA(G)} \sum_{j \in C(H)} y_{ij} \leq k,$$

$$\forall OA(G), |OA(G)| = 2k + 1, k > 1, \forall C(H), |C(H)| > k \quad [3.34]$$

$$\sum_{i \in I(G)} y_i \leq 1,$$

$$\forall \text{conjunto independente } I(G) \in G \quad [4.2]$$

$$\sum_{i \in OA(G)} y_i + \sum_{i \notin OA(G)} \alpha_i y_i \leq k,$$

$$\forall \text{Odd-Antihole } OA(G) \in G, |OA(G)| = 2k + 1 \quad [4.3]$$

$$\sum_{i \in OH(G)} y_i + \sum_{i \notin OH(G)} \alpha_i y_i \leq 2,$$

$$\forall \text{Odd-Hole } OH(G) \in G \quad [4.4]$$

5.1.1 Formulações

Visando determinar quais formulações produziam os melhores resultados computacionais, implementamos algoritmos de **B&B** utilizando algumas formulações diferentes compostas pelas restrições apresentadas nos Capítulos 2 e 3.

A primeira formulação implementada foi aquela apresentada no Capítulo 2, a qual chamamos de formulação **inicial**. A implementação desta formulação é demasiadamente simples, não necessitando de maiores detalhes sobre a forma como foi realizada. Ela é composta por $nG + nH$ desigualdades de mapeamento único e $mGm\overline{H} + m\overline{G}mH$ desigualdades de mapeamento de arestas, onde: nG é o número de vértices do grafo G , nH é o número de vértices do grafo H , mG o número de arestas do grafo G , $m\overline{G}$ o número de não-arestas no grafo G , mH o número de arestas no grafo H e $m\overline{H}$ o número de não-arestas no grafo H . Deste momento em diante assumiremos estes significados para os símbolos nG , nH , mG , $m\overline{G}$, mH e $m\overline{H}$ em todo o texto.

A segunda formulação implementada foi a que chamamos de **Clique-IS**. Esta formulação é obtida substituindo-se as restrições dadas pelas inequações (2.4) e (2.5) pelas restrições representadas pelas inequações (3.11) e (3.12). Resultando em:

$$\begin{aligned}
\max \quad & \sum_{i \in V(G)} \sum_{j \in V(H)} y_{ij} \\
& \sum_{i \in V(G)} y_{ij} \leq 1, \quad \forall j \in V(H) \\
& \sum_{j \in V(H)} y_{ij} \leq 1, \quad \forall i \in V(G) \\
& \sum_{i \in C} \sum_{j \in I} y_{ij} \leq 1, \quad \forall \text{ clique maximal } C \in G, \forall \text{ IS maximal } I \in H \\
& \sum_{i \in I} \sum_{j \in C} y_{ij} \leq 1, \quad \forall \text{ IS maximal } I \in G, \forall \text{ clique maximal } C \in H \\
& y_{ij} \in \{0, 1\} \quad \forall i \in V(G), j \in V(H)
\end{aligned}$$

Como já dissemos quando apresentamos as desigualdades (3.11) e (3.12), estas restrições são mais fortes que aquelas utilizadas na formulação inicial. Assim, esta nova formulação deve produzir melhores limitantes duais que aquela.

Como os conjuntos de todas as cliques e IS maximais de um grafo possuem um número exponencial de elementos, não é possível gerar uma formulação que contenha todas as desigualdades da formulação Clique-IS. Ao invés disso, para cada um dos grafos, obtemos um conjunto de cliques maximais que cobre todas as arestas e um conjunto de IS maximais que cobre todas as não-arestas. Num modelo obtido desta forma o número de desigualdades é menor do que aquele necessário ao se utilizar as desigualdades do tipo (2.4) e (2.5). Agora, precisamos garantir que a formulação obtida é válida para o **MSIC**.

Proposição 5.1 *O modelo Clique-IS obtido como descrito acima é uma formulação para o problema do MSIC.*

Prova: Suponha para o propósito de se obter uma contradição que aquele modelo não define uma formulação para o **MSIC**. Logo, deve existir um vetor binário y que representa uma solução viável para o **MSIC** e que não atende a pelo menos uma restrição do modelo, ou então, existe um vetor binário y que não representa uma solução viável para o **MSIC** e que satisfaz o modelo.

Primeiramente, note que como foi provado que todas as desigualdades do modelo são válidas, não pode existir uma solução viável y que não satisfaça alguma das restrições. Portanto, para fins de contradição, deve-se supor que existe uma solução inviável para o **MSIC** cujo vetor característico y satisfaz o modelo.

Uma solução inviável para o **MSIC** mapeia um vértice de um dos grafos de entrada em mais de um vértice do outro grafo, ou mapeia dois vértices adjacentes de um dos grafos em dois vértices não-adjacentes do outro grafo. Suponha que exista uma solução y que mapeia um vértice i de G em mais de um vértice de H . Considere a restrição de

mapeamento único do vértice i e sejam k e k' dois vértices de H para os quais i esteja sendo mapeado na solução. Note que as variáveis que representam os mapeamentos de i em k e k' , ou seja, y_{ik} e $y_{i,k'}$, respectivamente, valem 1 e também têm coeficiente 1 na restrição de mapeamento único do vértice i . Por outro lado, esta restrição é do tipo menor ou igual (\leq) e tem lado direito 1 (um). Assim, ela tem que estar violada pelo vetor y , o que é uma contradição.

Desta forma, se existir uma solução inviável para o **MSIC** cujo vetor característico satisfaz o modelo, esta solução deve mapear um par de vértices adjacentes de um dos grafos de entrada em um par de vértices não-adjacentes no outro grafo. Suponha sem perda de generalidade que o par de vértices adjacentes, (i, j) , pertence ao primeiro grafo, G , da entrada e o par de vértices não-adjacentes, (k, l) , ao segundo grafo, H . Pela definição do modelo Clique-IS, toda aresta de G faz parte de alguma das cliques consideradas e toda não-aresta de H faz parte de algum IS considerado. Logo, existe uma clique C em G que contém (i, j) e um IS I em H que contém (k, l) tais que C e I foram usados para montar uma desigualdade do tipo Clique-IS que tem do lado esquerdo coeficiente 1 (um) em todas as variáveis correspondentes a mapeamentos dos vértices de C em vértices de I e coeficiente 0 (zero) em todas as demais. Esta restrição é do tipo menor ou igual (\leq) e tem lado direito igual a 1 (um). Portanto, a solução inviável aqui descrita teria as variáveis que mapeiam simultaneamente i em k e j em l , ou i em l e j em k , com valor 1 (um) e, desta forma, a restrição Clique-IS formada pelos mapeamentos dos vértices de C nos vértices de I seria violada. Com isto, conclui-se que o vetor associado a tal solução não pode satisfazer o modelo, chegando-se então a um absurdo. \square

As cliques e os IS maximais são obtidos pelo mesmo algoritmo aplicado respectivamente aos grafos originais e aos grafos complementares. Antes de mostrar o algoritmo, devemos falar sobre as estruturas de dados usadas para representar os grafos. Não entraremos em detalhes específicos de como essas estruturas são implementadas, basta dizer, para a compreensão do restante do texto, que foram utilizadas: listas de incidência, listas de adjacência e matrizes de adjacência. Além disso, os graus dos vértices podem ser obtidos em tempo constante. A justificativa para a utilização de várias estruturas que representam os mesmos grafos de forma redundante é que, no caso de problemas deste tipo, o tempo é um recurso muito mais escasso que a memória. Desta forma, cada estrutura foi utilizada quando se fez mais conveniente para tornar o algoritmo mais rápido.

O Algoritmo 5.1 encontra uma seqüência de cliques (de acordo com a definição da Seção 3.2) em um grafo P dado como entrada. Além de P , o algoritmo também recebe como entrada uma fila z que determina uma ordem dos vértices de P . Esta ordem é aquela em que os vértices serão considerados para entrar em uma clique ou para iniciar uma nova clique na seqüência. As funções: `CriaFila()`, `InsererFila()`, `RemoveFila()`, `FilaVazia()`, `CriaClique()`, `InsererClique()` e `Adj()`, têm os seus significados intuitivos, isto é,

Entrada: $P = (V, E)$; z ;

/ Onde: $P = (V, E)$ é o grafo que contém a sequência de cliques; V é o conjunto de vértices e E o conjunto de arestas de P e z é uma fila com uma ordem dos vértices em V . */*

/ Numa iteração qualquer tem-se que: z contém os vértices não adjacentes a qualquer clique da sequência atual; nz é a lista dos vértices vizinhos de algum vértice da clique corrente e, se i é um vértice em nz , $conta[i]$ é o número de vértices da clique corrente adjacentes a i ; */*

Saída: sequência de cliques maximais (ou uma única clique maximal);

```

1 para  $i \leftarrow 1$  até  $|V|$  faça
2    $conta[i] \leftarrow 0$ ;
3  $l \leftarrow 1$ ;                               /* índice da clique corrente */
4 CriaFila( $nz$ );
5 enquanto  $\neg$  FilaVazia( $z$ ) faça
6    $u \leftarrow$  PrimeiroFila( $z$ );             /* retorna vértice e remove da fila */
7   CriaClique( $l$ );
8   InsereClique( $u, l$ );
9   para todo  $v \in Adj(u)$  com  $conta[v] = 0$  e considerados na ordem em que
    aparecem em  $z$  faça
10     $conta[v] \leftarrow 1$ ;
11    RemoveFila( $v, z$ );
12    InsereFila( $v, nz$ );
13     $grauNecessario \leftarrow 1$ ;
14    enquanto  $\neg$  FilaVazia( $nz$ ) faça
15       $v \leftarrow$  PrimeiroFila( $nz$ );
16      se  $conta[v] = grauNecessario$  então
17        InsereClique( $v, l$ );
18        para todo  $w \in Adj(v)$  faça
19          se  $conta[w] = 0$  então
20            RemoveFila( $w, z$ );
21             $conta[w] ++$ ;
22             $grauNecessario ++$ ;
23     $l ++$ ;                                     /* índice da próxima clique */

```

Algoritmo 5.1: Algoritmo para obtenção de uma sequência de cliques em um grafo.

respectivamente: cria uma nova fila, insere um elemento em uma fila, remove um elemento de uma fila, testa se a fila é vazia, cria uma nova clique, insere um vértice em uma clique e, retorna a lista de adjacência de um vértice. Vale observar que todas estas operações foram implementadas de modo a ter complexidade $O(1)$. Note-se ainda que as listas de adjacência passadas na entrada do algoritmo devem estar ordenadas de acordo com a ordem imposta pela fila z . Em breve veremos o porquê desta última exigência.

Continuando com a descrição do algoritmo, a variável l mantém o índice da clique que está sendo construída atualmente. Caso se deseje obter apenas uma clique maximal ao invés de uma seqüência, basta modificar a linha 4 de modo que a condição se torne: $\neg \text{FilaVazia}(z) \wedge l = 1$. O vetor $\text{conta}[i]$ mantém um registro do número de vértices adjacentes a i que foram colocados em alguma clique da seqüência e grauNecessario determina o número de vizinhos que um vértice deve ter na clique corrente para que seja adicionado a ela. Este algoritmo tem complexidade de tempo de ordem $O(|V| + |E|)$, sendo V e E os conjuntos de vértices e arestas de P , respectivamente.

Para obter uma cobertura das arestas de P por cliques maximais, utilizamos o Algoritmo 5.1 repetidamente, construindo uma clique de cada vez e, marcando de forma adequada as arestas cobertas pelas cliques encontradas. A cada nova execução a fila z passada na entrada do algoritmo é montada de forma que os vértices das arestas ainda não marcadas apareçam primeiro. Como dito anteriormente, isso força que as listas de adjacência sejam reordenadas de acordo com z , o que pode ser feito em $O(|V| + |E| \log |E|)$. Este procedimento iterativo gera uma cobertura de arestas contendo no máximo $|E|$ cliques maximais. Deste modo, para o **MSIC** tendo os grafos G e H como entrada, uma formulação Clique-IS possui no máximo $mGm\bar{H} + mHm\bar{G}$ desigualdades de mapeamento de arestas e exatamente $nG + nH$ desigualdades de mapeamento único.

A próxima formulação implementada foi chamada de **SeqClique-SeqIS**. Ela é obtida fazendo-se um *lifting* (Seção 5.1.2) das restrições da formulação Clique-IS de modo que as restrições (2.2) são substituídas pelas restrições (3.3), (3.7) e (3.8), as restrições (2.3) são substituídas pelas restrições (3.4), (3.9) e (3.10), as restrições (3.11) são substituídas pelas restrições (3.15) e as restrições (3.12) são substituídas por (3.16). A formulação obtida desta forma é mais forte que a formulação Clique-IS e, por isso, deve gerar limitantes duais não piores que esta.

Como a formulação SeqClique-SeqIS é obtida através de um *lifting* da formulação Clique-IS sem adição de nenhuma outra desigualdade, o número de desigualdades das duas formulações é exatamente o mesmo. No entanto, o número de elementos não-nulos na primeira é maior que na segunda. Os algoritmos utilizados para realizar os procedimentos de *lifting* usados na formulação SeqClique-SeqIS serão discutidos em detalhes na Seção 5.1.2.

Na formulação SeqClique-SeqIS, as restrições de mapeamento único definem facetas,

como mostramos no Capítulo 3, porém as restrições de mapeamento de arestas não são. Logo, podemos obter uma formulação ainda mais forte fazendo o *lifting* dessas restrições para as restrições de mapeamento de arestas que definem facetas mostradas no Capítulo 3. No entanto, a caracterização destas últimas é bastante complicada e, como mostramos no Capítulo 4 que as mesmas são equivalentes às restrições (4.2) do modelo da Clique, podemos fazer um *lifting* baseado nestas restrições. Assim, obtemos a formulação a qual chamamos de MaxIS, na qual é feito o *lifting* das desigualdades de mapeamento de arestas da formulação SeqClique-SeqIS de modo que elas passem a representar conjuntos independentes maximais no grafo associação.

Além das formulações já apresentadas, implementamos também, a título de comparação, uma formulação totalmente baseada no problema da Clique. Esta formulação foi chamada de formulação MaxIS Light por ser uma formulação onde as desigualdades são baseadas em IS maximais que formam uma cobertura das não-arestas do grafo associação para o problema da Clique gerado a partir dos grafos de entrada do MSIC. O termo “Light” vem do fato de que esta formulação geralmente possui menos desigualdades que a formulação MaxIS já apresentada.

5.1.2 Estratégias de Lifting

Uma forma bem conhecida na literatura para obter melhores limitantes duais é aplicar procedimentos de *lifting* sobre as desigualdades que fazem parte da formulação. Podemos pensar em diversas formas para realizar este procedimento. Uma forma possível é realizá-lo antes mesmo de resolver a primeira relaxação linear, o que na verdade dá origem a uma nova formulação (o que foi mostrado na Seção 5.1.1), que chamaremos de *lifting a priori*. Outra forma é realizá-lo após a otimização da relaxação linear aplicando o *lifting* apenas às restrições ativas na solução da relaxação linear, que chamaremos de *lifting a posteriori*.

Para estas duas formas de realizar o procedimento de *lifting* existem vantagens e desvantagens. As vantagens de usar o *lifting a priori* são que o procedimento de *lifting* é realizado uma única vez e a formulação resultante é mais forte que a original. No entanto, a desvantagem é que o número de coeficientes não-nulos nas restrições aumenta, podendo levar a tempos maiores para encontrar as soluções ótimas das relaxações lineares, sendo que várias das restrições reforçadas podem estar inativas nestas soluções. Quanto à utilização do *lifting a posteriori* a vantagem é que as restrições são reforçadas apenas quando necessário, evitando-se um aumento no tempo de otimização das relaxações lineares. Contudo, esta técnica apresenta a desvantagem de que os procedimentos de *lifting* para cada uma das restrições são repetidos diversas vezes ao longo da árvore de **B&B**.

Procedimentos de *lifting* foram implementados para as desigualdades (2.2), (2.3), (3.11), (3.12), (3.15) e (3.16). Estes procedimentos foram utilizados tanto *a priori*, dando

origem às formulações já discutidas na Seção 5.1.1, quanto *a posteriori*. Neste último caso, duas estratégias diferentes foram implementadas: na primeira estratégia, a cada nó da árvore de **B&B**, o *lifting* de uma única restrição ativa é executado, e a nova relaxação é reotimizada. O processo se repete até que não hajam mais restrições ativas para aplicar o *lifting*, o que pode acontecer porque todas as restrições ativas já são facetas ou porque um procedimento de *lifting* não foi implementado para algumas das restrições ativas. A segunda estratégia é similar, porém, ao invés de realizar o *lifting* de uma única restrição ativa por iteração, o *lifting* de todas as restrições ativas é realizado.

O procedimento de *lifting* para as desigualdades do tipo (2.2) pode ser dividido em quatro casos. No primeiro caso, o grafo G possui mais de um vértice isolado (desigualdade (3.3)). O segundo caso ocorre quando o grafo G possui mais de um vértice universal (desigualdade (3.3)). O terceiro caso se dá quando o grafo G possui um único vértice isolado (desigualdade (3.7)) e o quarto e último caso é quando o grafo G possui um único vértice universal (desigualdade (3.8)). O *lifting* das desigualdades do tipo (2.3) é equivalente ao das desigualdades do tipo (2.2).

Para o primeiro caso, o algoritmo de *lifting* pode ser descrito da seguinte maneira: inicialmente determine uma clique maximal que contém o vértice j (fixado na desigualdade) de H , em seguida atribua o valor 1 (um) aos coeficientes das variáveis que correspondem ao mapeamento dos vértices pertencentes à clique maximal contendo j nos vértices isolados de G , os demais coeficientes da desigualdade devem ser mantidos. Os custos dominantes neste algoritmo são o de determinar uma clique maximal contendo j que tem custo $O(nH + mH)$ e o de modificar os coeficientes das variáveis na desigualdade que, no pior caso, tem custo $O(nGnH)$. O segundo caso é similar, porém, ao invés de determinar uma clique maximal, determinamos um IS maximal contendo j - o que tem custo $O(nH + m\bar{H})$ - e modificamos os coeficientes das variáveis correspondentes aos mapeamentos dos vértices que fazem parte do IS em vértices universais de G .

O algoritmo de *lifting* para o terceiro caso é ainda mais simples que para os dois primeiros casos: atribua o valor 1 (um) aos coeficientes das variáveis que correspondem ao mapeamento dos vértices adjacentes a j de H no vértice isolado de G , os demais coeficientes permanecem inalterados. O custo deste algoritmo é de ordem $O(nG + nH)$. O quarto caso é semelhante e possui a mesma complexidade.

Para realizar o *lifting* das desigualdades do tipo (3.11) e obter uma desigualdade (3.15), procedemos da seguinte forma. Inicialmente, determinamos os vértices de G e H correspondentes às variáveis com coeficiente 1 (um), tais vértices formam respectivamente uma clique maximal em G e um IS maximal em H . A seguir, os vértices de G são colocados em uma fila z , de forma que os vértices que formam a clique ocupem as primeiras posições e, então, o Algoritmo 5.1, com G e z passados como entrada, é executado para obter uma seqüência de cliques. O mesmo é feito no grafo \bar{H} para obter uma seqüência de IS. Em

seguida atribuímos o valor 1 (um) às variáveis correspondentes ao mapeamento de vértices pertencentes às cliques do grafo G em vértices pertencentes à IS do grafo H em que a clique e o IS estão na mesma posição de ordem (vértices da primeira clique mapeados em vértices do primeiro IS, da segunda clique no segundo IS e assim sucessivamente). Os demais coeficientes permanecem inalterados. O custo de obter uma sequência de clique como descrito é da ordem de $O(nG + mG \log mG)$, obter uma sequência de IS tem um custo da ordem de $O(nH + m\bar{H} \log m\bar{H})$, enquanto alterar os coeficientes da desigualdade tem custo de ordem $O(nGnH)$. Assim como no caso das desigualdades do tipo (2.2) e (2.3), o *lifting* das desigualdades do tipo (3.12) para o tipo (3.16) é equivalente ao das desigualdades do tipo (3.11) para o tipo (3.15).

O último procedimento de *lifting* implementado foi o procedimento para as desigualdades dos tipos (3.15) e (3.16). O *lifting* natural para estas desigualdades, de acordo com o que foi apresentado no Capítulo 3, seria para as desigualdades (3.19) e (3.20). Porém, como já comentamos na Seção 5.1.1, a caracterização destas últimas é complexa. Desta forma, como no Capítulo 4 provamos a equivalência de (3.19) e (3.20) com (4.2), fazemos o *lifting* de (3.15) e (3.16) para (4.2) usando o grafo associação e, por este fato, o procedimento é exatamente o mesmo para os dois tipos de desigualdades do **MSIC**. Inicialmente criamos uma fila z com os vértices do grafo associação (A) de tal forma que os vértices correspondentes às variáveis que possuem coeficiente 1 (um) na desigualdade em questão apareçam primeiro que os demais. Em seguida, aplicamos o Algoritmo 5.1 dando \bar{A} e z como entrada e, então, atribuímos valor 1 (um) às variáveis correspondentes aos vértices que fazem parte da clique maximal assim obtida. A complexidade deste algoritmo é da ordem $O(nA + m\bar{A} \log m\bar{A})$ onde: $nA = nGnH$ é o número de vértices do grafo associação e $m\bar{A}$ é o número de não-arestas nesse grafo.

5.1.3 Rotinas de Separação

Como já mencionamos no Capítulo 2, um algoritmo de **B&C** consiste em executar algoritmos de planos de corte em cada nó da árvore de **B&B** fazendo assim, uma reformulação automática do problema. Para realizar essa reformulação é preciso resolver um problema conhecido como problema da separação, que consiste em encontrar uma desigualdade válida para o problema e que é violada pela solução ótima da relaxação linear. Este processo reforça a formulação, mas, assim como no caso do *lifting*, existe a desvantagem de a formulação resultante ser mais pesada, fazendo com que a resolução da relaxação linear seja mais lenta.

Para os propósitos deste trabalho implementamos rotinas heurísticas de separação para as desigualdades (3.11), (3.12), (4.2) e para as desigualdades (3.27), (3.28), (3.29), (3.30), (3.33), (3.34), (3.31) e (3.32). O algoritmo para os dois primeiros tipos de desigualdades é

idêntico, apenas trocando os grafos usados na primeira por seus complementos. O mesmo acontece com as desigualdades (3.27) a (3.32). Assim, a seguir apresentamos os três procedimentos usados para separação no nosso algoritmo de **B&C**.

A rotina de separação para as desigualdades (3.11) segue o seguinte algoritmo. Ordenamos as variáveis pela proximidade de 0,5, isto é, primeiro as variáveis que mais violam as restrições de integralidade. Essa ordem das variáveis induz uma ordem z sobre os vértices dos grafos de entrada do **MSIC** G e H . Com isto, aplicamos o Algoritmo 5.1 sobre os grafos G e \overline{H} para obter, respectivamente, uma clique maximal e um IS maximal. Em seguida, verificamos se a soma das variáveis correspondentes ao mapeamento dos vértices da clique nos vértices do IS é maior que 1 (um), em caso positivo o corte é adicionado.

Para as desigualdades (4.2) primeiro criamos uma fila z ordenando as variáveis pela proximidade de 0,5 e utilizamos o Algoritmo 5.1 passando o complemento do grafo associação (\overline{A}) e z como entrada. Testamos se a desigualdade formada pelo IS em A assim obtido é violada. Em caso negativo, criamos uma fila z ordenando as variáveis pela proximidade de 1 e executamos todo o procedimento novamente para obter uma nova desigualdade e testar se ela é violada. Esse procedimento é semelhante àquele encontrado em [25].

No procedimento de separação para as desigualdades (3.27) a (3.32), ordenamos as variáveis do problema pela proximidade de 0,5, induzindo assim uma ordem sobre os vértices dos grafos de entrada do **MSIC**, e obtemos uma clique maximal (IS maximal) em um dos grafos da mesma forma que foi feito na separação das desigualdades (3.11). Em seguida, para toda aresta (u, v) do outro grafo atribuímos peso $2 - (\sum_{i \in clique} x_{iu}) - (\sum_{i \in clique} x_{iv})$ (obviamente este peso sempre será maior ou igual a zero devido às restrições de mapeamento único). Então, construímos um grafo bipartido de forma que se existe a aresta (x, y) no grafo, então as arestas (x, y') e (x', y) existem no grafo bipartido e possuem o mesmo peso daquela, além disso, a aresta (x, y) não existe no grafo bipartido. Finalmente, aplicamos o algoritmo de Dijkstra para encontrar o menor caminho entre x e x' para todo vértice x até que um *odd-hole* de tamanho maior ou igual a 5 seja encontrado ou que se verifique a inexistência do mesmo.

O procedimento descrito no parágrafo anterior é baseado naquele apresentado em [25] para separação de desigualdades *odd-hole* no problema do IS máximo. Para a desigualdade (3.27), por exemplo, o caminho encontrado no grafo bipartido terá peso $4k + 2 - 2 \sum_{i \in clique} \sum_{j \in oddHole} x_{ij}$, onde $2k + 1$ é o número de vértices do *odd-hole* e x_{ij} é a variável correspondente ao mapeamento do vértice i , de um dos grafos, no vértice j , do outro grafo. Assim, como o lado esquerdo da desigualdade é dado por $\sum_{i \in clique} \sum_{j \in oddHole} x_{ij}$, para uma determinada clique maximal em um dos grafos, o *odd-hole* no outro grafo que maximiza o lado esquerdo da desigualdade é dado pelo caminho mínimo no grafo bipar-

tido.

5.1.4 Estratégias de Seleção de Variáveis

Uma questão importante em um algoritmo de **B&B** que surge após a otimização de uma relaxação linear em que a solução ótima não é inteira é: qual das variáveis fracionárias deve ser selecionada para ramificação? Isto é, qual variável será usada para dividir o problema em questão em um determinado nó em dois novos subproblemas correspondentes ao arredondamento da variável para cima e para baixo? Embora o **XPRESS-MP** implemente estratégias para realizar essa seleção, implementamos alguns critérios baseados no valor das variáveis, características dos grafos de entrada do **MSIC** e características do grafo associação.

O primeiro critério implementado foi a seleção da variável “mais fracionária”, no nosso caso, a variável mais próxima de 0,5. Assim, após a otimização de uma relaxação linear, quando obtemos uma solução fracionária, dividimos o problema em dois subproblemas fixando a variável com valor mais próximo de 0,5 em 1 (um) em um deles e em 0 (zero) no outro.

Outro critério para seleção de variáveis implementado foi pré-determinar uma ordem de seleção das variáveis baseada no grau dos vértices do grafo associação. Tal ordem é dada selecionando o vértice de menor grau no grafo associação e colocando-o na primeira posição livre de um vetor de ordem, em seguida o vértice é removido do grafo associação e o procedimento se repete sobre o grafo resultante até que todos os vértices estejam ordenados. Como cada vértice do grafo associação está relacionado a uma variável do problema, essa ordem dos vértices determina também uma ordem sobre as variáveis. Desta forma, após a otimização de uma relaxação linear, a variável selecionada para ramificação será a primeira variável com valor fracionário na ordem pré-definida.

A ordenação dos vértices apresentada no parágrafo anterior é comumente usada em algoritmos para o problema da Clique. Outra forma de ordenar os vértices, também utilizada naquele problema, é a ordenação por cores, ambas são apresentadas em [40] e usadas pelo Cliquer. A ordenação por cores é feita determinando uma classe de cor por vez (dois vértices podem pertencer à mesma classe de cor se forem não-adjacentes) e em uma mesma classe de cor, vértices com maior grau são ordenados primeiro. Após a ordenação de toda uma classe de cor, seus vértices são removidos e o processo continua sobre o grafo induzido pelos vértices ainda não ordenados até que este se torne vazio. Por último, a ordem obtida da forma aqui descrita é invertida. Esta ordem dos vértices do grafo associação determina como escolhemos a variável de ramificação da mesma forma que na ordenação por grau.

A última ordenação que implementamos foi a que chamamos ordenação por grau de

mapeamento. Nela, calculamos para cada vértice v dos grafos de entrada do **MSIC**, o grau de mapeamento de v , que é dado pela soma dos graus dos vértices do grafo associação que correspondem a mapeamentos de v nos vértices do outro grafo. Assim, os vértices dos dois grafos são ordenados de modo que o vértice com menor grau de mapeamento é colocado na primeira posição livre de um vetor de ordem. Em seguida, o vértice é removido do seu grafo de origem e o vértice com menor grau de mapeamento no grafo resultante é colocado na próxima posição livre. Esse processo se repete até que todos os vértices estejam ordenados. Com os vértices dos dois grafos ordenados, ordenamos as variáveis de acordo com o procedimento descrito a seguir, no qual supõe-se que o número de vértices do grafo de entrada G é menor ou igual ao número de vértices do grafo de entrada H . Na primeira posição coloca-se a variável correspondente ao mapeamento do primeiro vértice de G no primeiro vértice de H . Depois, a variável correspondente ao mapeamento do primeiro vértice de G no segundo vértice de H e assim sucessivamente até que as variáveis correspondentes a todos os mapeamentos do primeiro vértice de G estejam ordenadas. Em seguida consideramos os mapeamentos do segundo vértice de G e o processo se repete até que todas as variáveis estejam ordenadas.

A motivação por trás da ordenação por grau de mapeamento é, na verdade, realizar uma ordenação por cores utilizando uma coloração previamente conhecida. Tal coloração se baseia em uma propriedade do grafo associação que todo par de vértices (i, j) e (i, k) são não-adjacentes, bem como todo par (i, j) e (k, j) . Desta forma, todo grafo associação tem duas colorações dadas pelas classes: $(i_1, ?), (i_2, ?) \dots (i_{nG}, ?)$ e $(?, j_1), (?, j_2) \dots (?, j_{nH})$, onde $(i_x, ?)$ são os vértices do grafo associação obtidos a partir dos mapeamentos do vértice i_x de um dos grafos em vértices do outro grafo ($(?, j_y)$ tem significado semelhante). Logo, todo grafo associação obtido a partir de dois grafos G e H pode ser colorido com $\min(nG, nH)$ cores. Essas classes de cores são exatamente aquelas que aparecem nas restrições de mapeamento único.

5.1.5 Estratégia de Ramificações Múltiplas

Analisando a árvore de **B&B** percebemos que após a escolha de uma variável para ramificação, quando ela é fixada em 1 (um) várias outras variáveis são fixadas em 0 (zero). Todavia, quando a variável de ramificação é fixada em 0 (zero), nenhuma outra variável é fixada. Este tipo de ramificação tende a criar árvores de enumeração desbalanceadas que, notoriamente, são indesejáveis para algoritmos do tipo divisão e conquista como é o caso do **B&B**. Uma alternativa é fazer ramificações múltiplas sobre grupos de variáveis binárias cuja soma seja limitada a 1 (um). Assim, em cada ramo uma variável do grupo é fixada em 1 (um) e as demais em 0 (zero). Além disso, acrescenta-se um último ramo onde todas as variáveis do grupo são fixadas em 0 (zero). Com isto, espera-se que o número de

nós total da árvore de enumeração seja reduzido.

A escolha dos grupos de variáveis a serem usados para a realização das ramificações múltiplas pode ser feita de diversas maneiras. No entanto, para o problema do **MSIC**, estes grupos devem possuir a característica de que quando uma de suas variáveis assume valor 1 (um), todas as demais devem assumir valor 0 (zero), isto é, duas variáveis de um grupo não podem assumir valor 1 (um) simultaneamente. Dito de outra forma, se um grupo é válido para a realização de ramificações múltiplas no problema do **MSIC**, a desigualdade do tipo menor ou igual em que o lado esquerdo é o somatório das variáveis de um grupo e o lado direito é 1 (um) é uma desigualdade válida para o **MSIC**.

No caso do **MSIC**, tendo em vista a propriedade que os grupos de variáveis para ramificações devem satisfazer, selecionamos os grupos da seguinte forma. Dadas as desigualdades do tipo " ≤ 1 " de alguma formulação do **MSIC**, escolhemos aquela que possua o maior número de variáveis com coeficiente 1 (um) e marcamos essas variáveis como cobertas. A seguir escolhemos dentre as desigualdades restantes aquela que possui o maior número de variáveis ainda não cobertas e repetimos este processo até que todas as variáveis sejam cobertas. As variáveis que possuem coeficiente 1 (um) em cada desigualdade selecionada, removidas aquelas nas interseções com as demais desigualdades selecionadas, definem um dos grupos que serão utilizados nas ramificações múltiplas.

Escolhendo os grupos de variáveis da forma descrita no parágrafo anterior, obtemos um conjunto de grupos que forma uma partição do conjunto de variáveis do problema. Ao longo do algoritmo de **B&B**, o grupo selecionado para ramificação é aquele que contém a variável que seria selecionada se estivesse sendo executada uma ramificação binária.

5.1.6 Estratégias de Ramificação ao Estilo Cliquer

Devido ao bom desempenho apresentado pelo Cliquer, pode-se imaginar que a estratégia usada pelo algoritmo pode ser pelo menos parcialmente implementada em outros algoritmos. Assim, implementamos a idéia utilizada pelo Cliquer de resolver inicialmente um problema bem pequeno, em seguida resolver um problema um pouco maior utilizando o resultado do problema anterior e assim sucessivamente, até que o problema original fosse resolvido.

Essa idéia deu origem a duas abordagens: na primeira delas, as variáveis são ordenadas de acordo com algum critério (pode-se utilizar alguns daqueles já discutidos para seleção de variáveis de ramificação) e, então, o algoritmo de **B&B** ou **B&C** é aplicado ao subproblema em que todas as variáveis são fixadas em 0 (zero), exceto a última. Em seguida resolvemos o problema em que todas as variáveis são fixadas em 0 (zero), exceto as duas últimas, adicionando à formulação a restrição de que a última variável deve ser menor que a solução ótima do subproblema anterior (obviamente a resolução do primeiro sub-

problema e a adição desta restrição são desnecessários e estão aqui descritas para facilitar a compreensão). Depois, resolvemos o problema em que todas as variáveis são fixadas em 0 (zero) exceto as três últimas e adicionamos à formulação, além da restrição adicionada no subproblema anterior, a restrição de que o somatório das duas últimas variáveis deve ser menor que o valor ótimo do subproblema anterior. Esse processo é repetido até que o problema original seja resolvido com algumas restrições adicionais que devem melhorar o limitante dual.

Como a fixação de variáveis e carga do problema geram custos adicionais, podemos tentar reduzir estes custos utilizando a segunda abordagem que é a de utilizar a estratégia do Cliquer adicionando grupos de variáveis nos subproblemas seguintes ao invés de uma única variável. Nesta abordagem, os grupos de variáveis podem ser obtidos através de uma coloração do grafo associação - neste caso, cada classe de cor seria um grupo - ou utilizando um algoritmo de partição das variáveis como o apresentado na Seção 5.1.5. A idéia do algoritmo permanece a mesma. Inicialmente todas as variáveis são fixadas em 0 (zero) exceto as pertencentes ao último grupo (assim como na abordagem já discutida, o primeiro problema não precisaria ser resolvido pois sua solução ótima é obviamente 1 (um)). Em seguida, todas as variáveis são fixadas em zero exceto as dos dois últimos grupos e é adicionada à formulação a restrição de que o somatório das variáveis do último grupo deve ser menor ou igual à solução ótima do subproblema anterior. Esse processo é repetido até que o problema original seja resolvido.

5.2 Resultados Computacionais

Na Seção 5.1 discutimos as diversas técnicas e estratégias que foram implementadas na tentativa de obter um algoritmo com melhor desempenho. Nesta seção analisaremos os resultados computacionais obtidos aplicando diferentes combinações daquelas técnicas com as diferentes formulações já apresentadas.

Para o conjunto de instâncias de teste foram gerados aleatoriamente, de acordo com o modelo de Erdős-Rényi, grafos com 10, 12, 15, 17 e 20 vértices em cinco faixas de densidades: 10%, 30%, 50%, 70% e 90%. Para cada número de vértices e faixa de densidade foram gerados dois grafos, dando um total de 50 grafos. Como a entrada do **MSIC** são dois grafos, todas as combinações de pares de grafos foram utilizados, resultando em um total de 1.225 instâncias de teste. A utilização de diferentes faixas de densidade visa analisar qual faixa traz mais dificuldades para os algoritmos. Para todos os testes realizados, um tempo máximo de 1.800 segundos.

O ambiente computacional utilizado nos experimentos consistiu em um computador com processador Intel Core 2 Quad 2.40GHz, cache de 4096 kB, memória RAM de 4GB e sistema operacional Ubuntu 8.10. A linguagem de programação usada na implementação

foi a linguagem C com compilador gcc (Ubuntu 4.3.2-1ubuntu12) 4.3.2 e todos os programas foram compilados com a *flag* de otimização -O3. Além disso, foi usado o resolvidor de **PLI XPRESS-MP-Optimizer** 64-bit v19.00.00 (xpressmp2008A.2). O **XPRESS-MP** contém bibliotecas especiais para implementar **B&C**, heurísticas primais etc. Os cortes, heurísticas e pré-processamento padrões do **XPRESS-MP** foram desligados.

Neste trabalho, procuramos seguir as recomendações constantes em [16] quanto à forma de realizar os experimentos e apresentar os resultados. Por este motivo, neste texto optamos por não incluir longas tabelas contendo dados das execuções experimentais individualizado por instância, como é comum em textos dessa natureza. Ao invés disso, compilamos os resultados, exibindo-os sob a forma de estatísticas e gráficos. Isso se deve principalmente ao fato da quantidade de resultados ser muito grande e, por isso, apresentá-los por completo seria tedioso e em partes, pouco significativa. Sobre as estatísticas relativas aos tempos de execução, deve-se notar que empates são inexistentes ou insignificantes. Assim, quando dizemos que, por exemplo, os tempos de execução do ExperimentoX são melhores que os do ExperimentoY em 23,00% dos casos, é o mesmo que dizer que eles são piores em 77,00% dos casos. A Tabela 5.1 resume todos os experimentos que são discutidos em detalhe no restante desta seção.

Os primeiros experimentos realizados visaram determinar qual das formulações conduz a um algoritmo mais eficiente. Inicialmente foi realizado um experimento executando cada uma das 1.225 instâncias utilizando **B&B** puro com a formulação inicial (este experimento foi chamado Experimento01). Posteriormente um experimento semelhante foi realizado com a formulação Clique-IS (Experimento02) e os resultados deste último foram comparados com aqueles do Experimento01. Concluída esta comparação, outro experimento semelhante foi executado, desta vez com a formulação SeqClique-SeqIS (Experimento03). Os resultados obtidos foram comparados com os melhores resultados entre os Experimento01 e Experimento02. O mesmo foi feito com as formulações MaxIS (Experimento04) e MaxIS Light (Experimento05).

No Experimento01 99,43% das instâncias foram resolvidas no tempo máximo de 1.800s e 77,47% no tempo de 30s. O tempo de montagem da formulação e carregamento do problema neste experimento é irrisório em relação ao tempo total, assim como nos demais experimentos, sendo menor que 0,192s em todas as instâncias. Neste experimento, as instâncias que se mostraram mais difíceis foram aquelas em que a densidade de um dos grafos estava entre 30% e 70% (fixada a densidade do outro grafo). Este fato se repetiu em todas as demais formulações como ilustra a Figura 5.1 relativa aos tempos do Experimento02 para instâncias em que um dos grafos possui 17 e o outro 20 vértices. Em média, a distância entre o limitante dual na raiz (LD_{raiz}) e o melhor (LD_{melhor}) - calculada por $100 \times ([LD_{raiz}] - [LD_{melhor}]) / [LD_{melhor}]$ - foi de 70,83%. Como 99,43% das instâncias foram resolvidas à otimalidade, LD_{melhor} é bem próximo do ótimo, em média.

Experimento	Descrição	Melhor
01	Formulação inicial com B&B .	
02	Formulação Clique-IS com B&B .	*
03	Formulação SeqClique-SeqIS com B&B .	
04	Formulação MaxIS com B&B .	
05	Formulação MaxIS Light com B&B .	
06	Mesmo que Experimento02 adicionado de <i>lifting a posteriori</i> de uma única restrição ativa por iteração.	
07	Mesmo que Experimento02 adicionado de <i>lifting a posteriori</i> de todas as restrições ativas por iteração.	
08	Formulação Clique-IS com B&C separando desigualdades (4.2).	
09	Formulação Clique-IS com B&C separando desigualdades (3.27) e (3.34).	
10	Formulação Clique-IS com B&C separando desigualdades (3.11) e (3.12)	
11	Mesmo que Experimento02 adicionado de seleção de variáveis por grau.	*
12	Mesmo que Experimento02 adicionado de seleção de variáveis por cores.	
13	Mesmo que Experimento02 adicionado de seleção de variáveis por grau de mapeamento.	*
14	Mesmo que Experimento02 adicionado de seleção por variável mais violada.	
15	Mesmo que Experimento13 adicionado de estratégia de ramificações múltiplas.	
16	Mesmo que Experimento13 adicionado de estratégia do Cliquer com uma única variável adicionada por vez.	
17	Mesmo que Experimento13 adicionado de estratégia do Cliquer com um conjunto de variáveis adicionado por vez.	
18	Cliquer	*

Tabela 5.1: Resumo de todos os experimentos realizados. Na primeira coluna é mostrado o número do experimento, na segunda a descrição do que é feito nele e a terceira coluna identifica o experimento que apresentou o melhor desempenho até o momento.

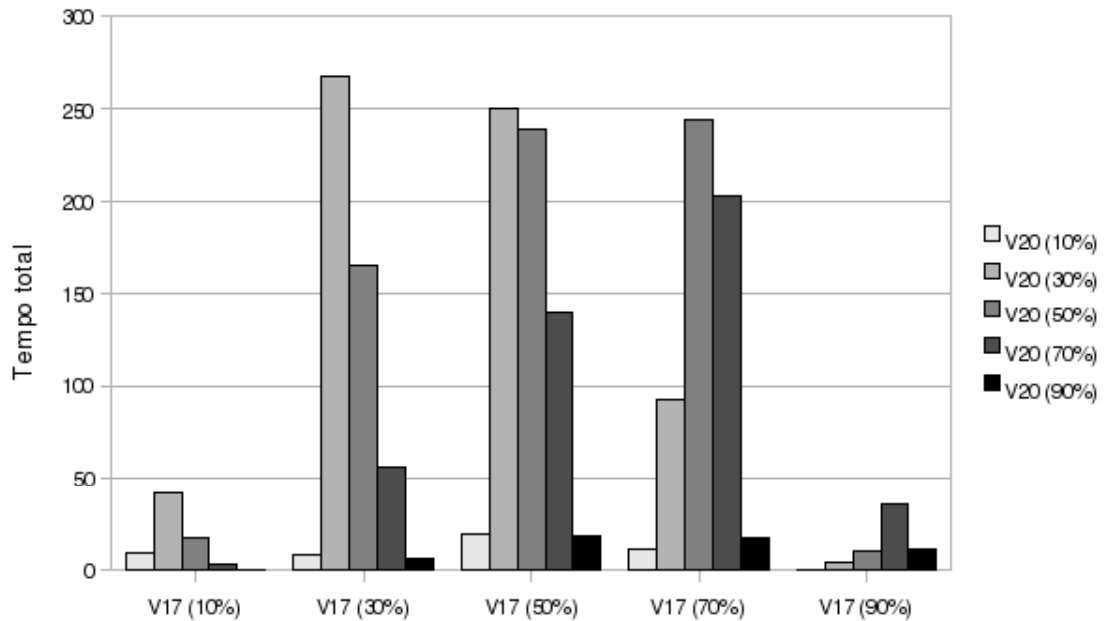


Figura 5.1: Tempo total de execução para diferentes densidades no Experimento02.

Os resultados do Experimento02 foram bem melhores que os do Experimento01, onde, 100,00% das instâncias foram resolvidas em no máximo 1.800s e 89,31% foram resolvidas em 30s. Na comparação entre esses dois experimentos, 95,92% das instâncias apresentam um tempo de resolução melhor no Experimento02 e, em média, essa melhora foi de 68,73% (a melhora do tempo de resolução de uma instância no Experimento02 em relação ao Experimento01, em valores percentuais, é definida como sendo $100 \times (tExp01 - tExp02)/tExp01$, onde $tExpX$ é o tempo da instância no ExperimentoX). Em média, a distância entre o limitante dual na raiz e o melhor limitante dual foi de 20,99% (muito melhor que o obtido com a formulação inicial) e como 100% das instâncias foram resolvidas à otimalidade, em todos os casos o melhor limitante dual era ótimo. Este ganho de tempo no Experimento02 pode ser explicado pelo fato da formulação Clique-IS ser formada por desigualdades mais fortes e, ao mesmo tempo, possuir um número menor de restrições e elementos não-nulos que a formulação inicial (em 100,00% das instâncias), o que pode tornar a resolução das relaxações lineares mais rápida.

Na Figura 5.2 mostramos gráficos comparando o tempo de execução das instâncias no Experimento01 e Experimento02. O eixo das ordenadas corresponde ao tempo gasto para resolver as instâncias e o eixo das abscissas representa as 1.225 instâncias ordenadas por número de vértices e por densidade do primeiro grafo e por número de vértices e densidade do segundo grafo. Ou seja, o valor 1 no eixo x corresponde à instância em que

o primeiro grafo tem 10 vértices e densidade 10% e o segundo grafo tem o mesmo número de vértices e densidade; o valor 2 no eixo x corresponde à instância em que o primeiro grafo tem 10 vértices e densidade 10% e o segundo grafo tem o mesmo número de vértices e densidade 30% e assim sucessivamente. Para uma melhor visualização, o gráfico foi dividido em seis partes correspondendo a um segmento do eixo das abscissas, sendo a primeira parte correspondendo às primeiras 204 instâncias, a segunda às 204 seguintes, assim por diante e a sexta parte corresponde às 205 últimas instâncias. Vale ressaltar que apesar dos gráficos serem mostrados como funções contínuas, na verdade, o eixo x é discreto como se pode perceber pela sua descrição. Contudo, entendemos que um gráfico como uma função discreta teria uma visualização mais difícil. Note que o gráfico do Experimento02, que tem um melhor desempenho, é apresentado com uma linha contínua enquanto o do Experimento01 é apresentado com uma linha pontilhada. Em todos os gráficos apresentados neste trabalho, adotamos esta convenção de que o experimento com melhor desempenho é apresentado com uma linha contínua e o de pior desempenho com uma linha pontilhada.

Como a formulação Clique-IS apresentou melhores resultados que a formulação inicial, que é mais fraca, é razoável pensar que a formulação SeqClique-SeqIS sendo ainda mais forte deveria apresentar resultados de melhor qualidade. Contudo, não foi isso o constatado. No Experimento03, 100,00% das instâncias foram resolvidas em no máximo 1.800s e 88,65% em 30s e em apenas 31,27% dos casos este experimento apresentou tempo inferior aos do Experimento02, sendo que a piora do tempo foi em média de 24,44% (a piora do tempo é simplesmente uma melhora negativa e, portanto, é calculada como já apresentado anteriormente). A piora dos tempos na formulação SeqClique-SeqIS na maioria dos testes pode ser explicada pelo fato desta formulação, apesar de formada por desigualdades mais fortes e possuir o mesmo número de restrições que a formulação Clique-IS, tem um número maior de elementos não-nulos (em 99,10% dos casos) e gera uma melhora do limitante dual na raiz de apenas 0,08% em média, sendo que a distância deste limitante na raiz para o melhor limitante foi de 20,49% em média. De fato, o aumento do número de elementos não-nulos é de 13,43% em média, o que deve contribuir para elevar, o custo de resolução das relaxações lineares. Entretanto, o aumento desse custo não é compensado pela melhora no limitante dual.

Mesmo com o resultado negativo observado no Experimento03 decidimos testar a possibilidade de uma formulação ainda mais forte produzir melhores resultados, o que poderia acontecer porque a melhora do limitante dual com tal formulação poderia compensar o aumento no custo de resolução das relaxações lineares. No Experimento04 essa possibilidade foi testada, sendo que 100,00% das instâncias foram solucionadas à otimalidade no tempo máximo de 1.800s e 88,82% foram resolvidas em 30s. Na comparação com o Experimento02 houve uma melhora de, em média, 0,25% do valor do limitante dual na

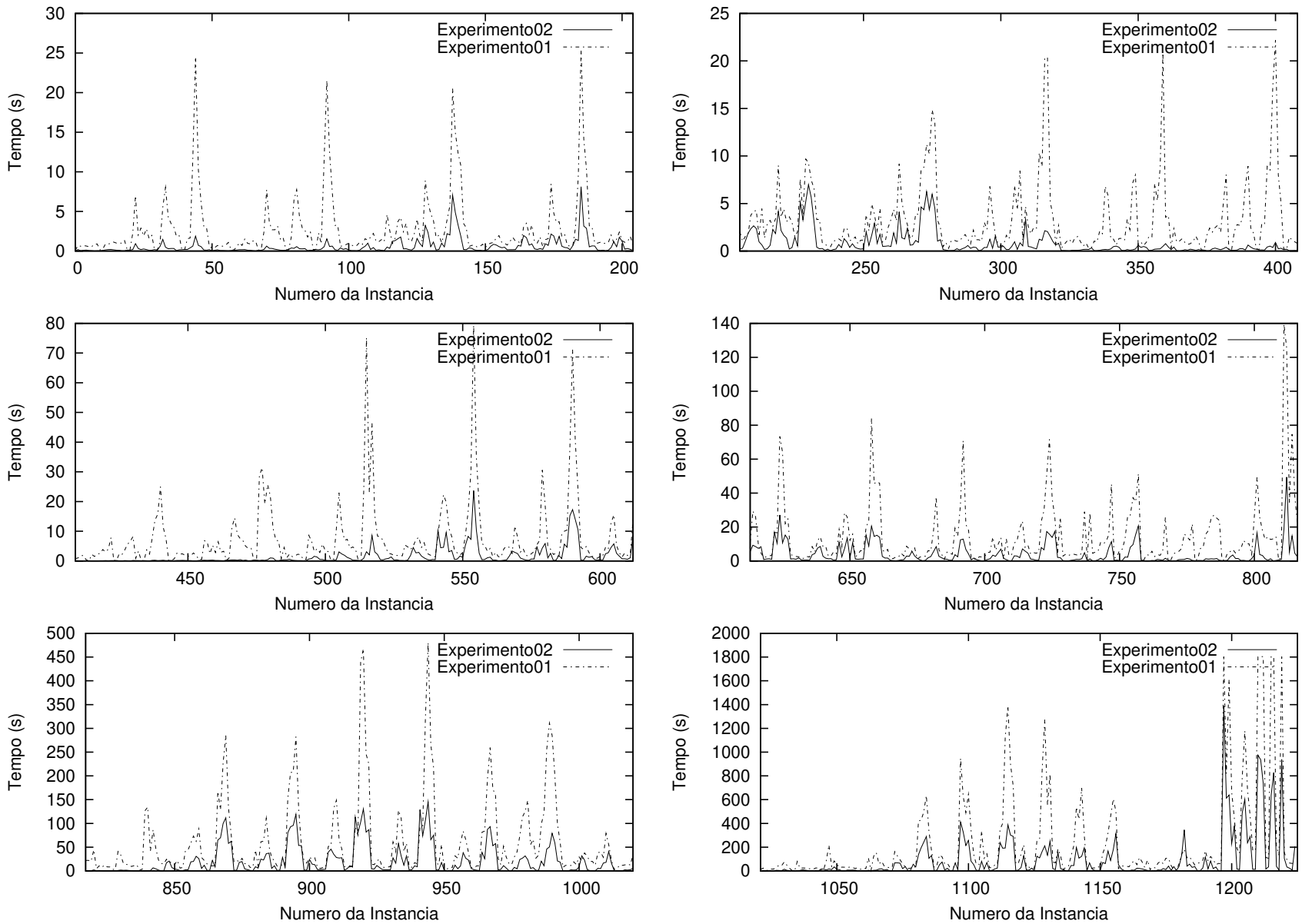


Figura 5.2: Comparação entre os tempos do Experimento02 e Experimento01.

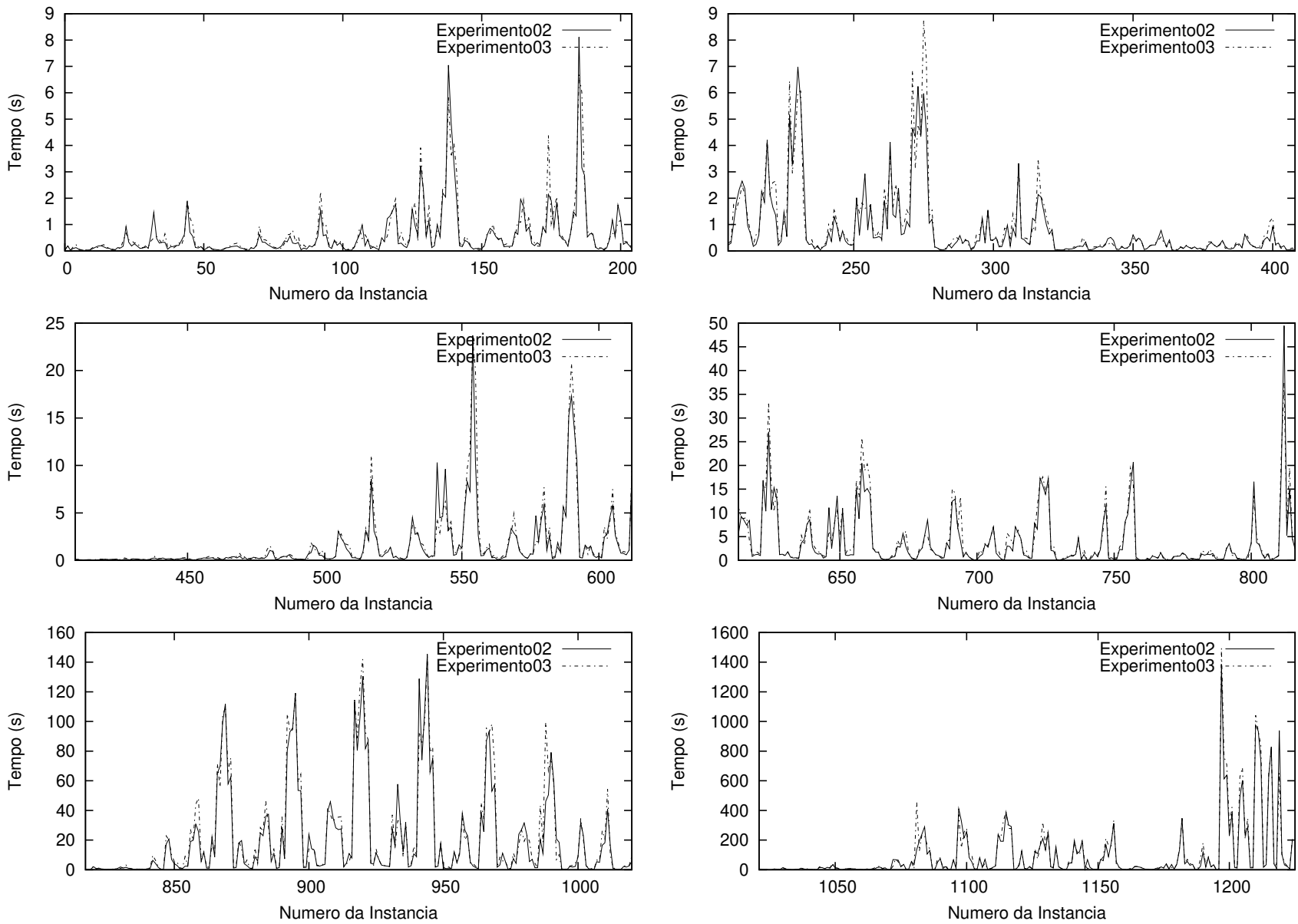


Figura 5.3: Comparação entre os tempos do Experimento03 e Experimento02.

raiz (a distância do limitante dual na raiz para o melhor limitante dual foi de 19,90% em média), contudo, em apenas 25,80% dos casos o tempo de resolução foi menor, havendo uma piora de, em média, 32,26%. Isso nos leva à conclusão de que, também nesta formulação, a melhora do limitante dual não foi suficiente para compensar o aumento no peso da formulação.

Observando a Figura 5.3 podemos perceber claramente que apesar do Experimento03 ter apresentado resultados piores que o Experimento02, a diferença entre os dois foi muito menos expressiva que a diferença entre os Experimento02 e Experimento01. O mesmo se observa na Figura 5.4 entre os Experimento04 e Experimento02.

Aparentemente, o fato da formulação Clique-IS ser mais leve que as formulações SeqClique-SeqIS e MaxIS faz com que ela possa ser resolvida mais eficientemente que as demais. Assim, podemos pensar que uma formulação que fosse ao mesmo tempo mais forte e mais leve poderia dominar a formulação Clique-IS. A formulação MaxIS Light, que foi usada no Experimento05, é formada apenas por desigualdades que definem faces do poliedro da Clique (e, conseqüentemente, também do poliedro do **MSIC**), porém, em geral, possui um número menor de restrições que a formulação Clique-IS (no Experimento05 o número de restrições foi menor que no Experimento02 em 80,73% dos casos). No Experimento05 todas as instâncias foram resolvidas em 1.800s e 82,20% em 30s, o que não parece muito pior que os resultados do Experimento02. No entanto, comparando os tempos obtidos nos dois experimentos, observamos que o Experimento05 foi melhor que o Experimento02 em apenas 8,82% dos casos, apresentando, em média, uma piora de 320,68%. Na Figura 5.5 percebe-se que a piora nos tempos de resolução do Experimento05 em relação ao Experimento02 foi mesmo bastante significativa. Este mau desempenho da formulação MaxIS Light pode ser explicado por dois fatos. Primeiro, apesar do número de restrições ser menor, esta formulação apresentou um número de elementos não-nulos por restrição maior que a Clique-IS em 86,94% dos casos, o que pode piorar o tempo de resolução das relaxações lineares. O segundo fato foi que, apesar de formada por desigualdades mais fortes (todas as desigualdades da formulação são facetas), houve uma degradação, em média, de 16,47% no valor do limitante dual na raiz (a distância entre o limitante dual na raiz e o melhor limitante dual foi de 37,20% em média), o que pode ser explicado pelo fato dessa formulação não levar em consideração as informações que podem ser obtidas da estrutura dos grafos de entrada originais (existência de cliques e IS maximais, *odd-(anti)holes* etc.). Assim, esse experimento apresenta indícios de que é vantajoso utilizar uma formulação baseada diretamente sobre problema do **MSIC** ao invés de outra que se vale da redução ao problema da Clique.

Estes primeiros experimentos apontaram como melhor formulação a Clique-IS, apesar desta ser mais fraca que outras formulações testadas. Nossa hipótese é de que isso ocorreu porque aquela formulação representa um melhor equilíbrio entre força das desigualdades

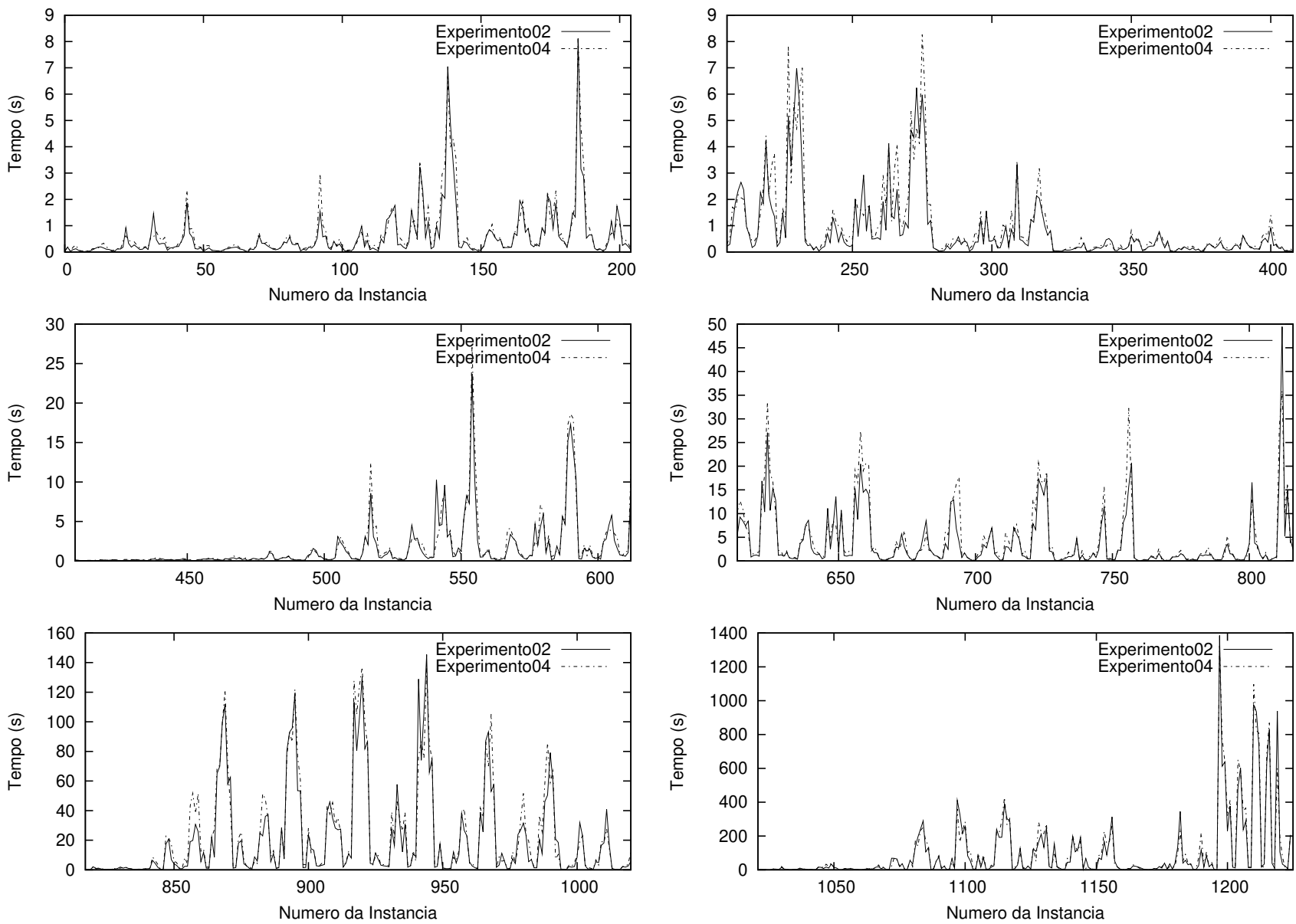


Figura 5.4: Comparação entre os tempos do Experimento04 e Experimento02.

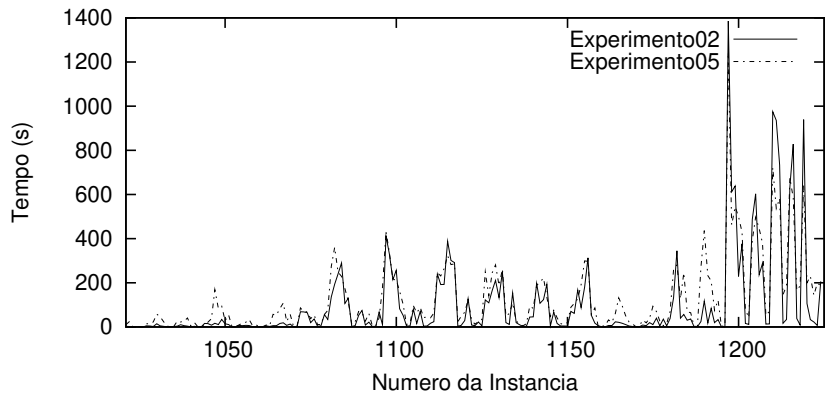
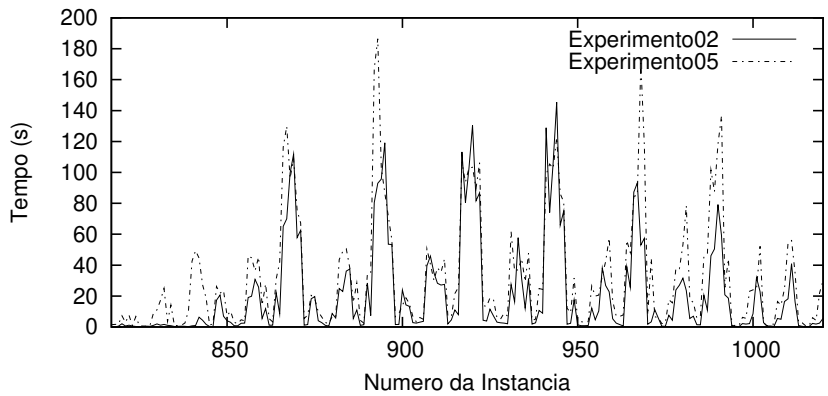
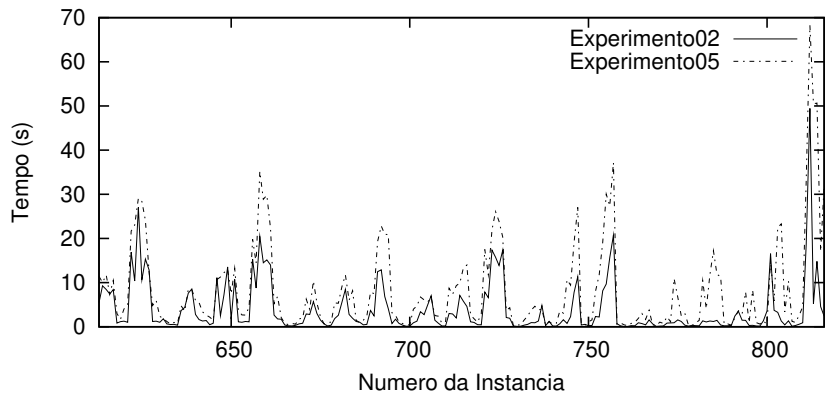
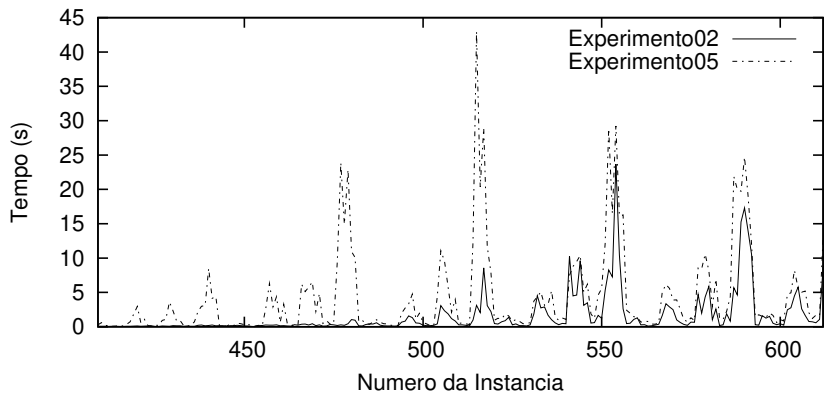
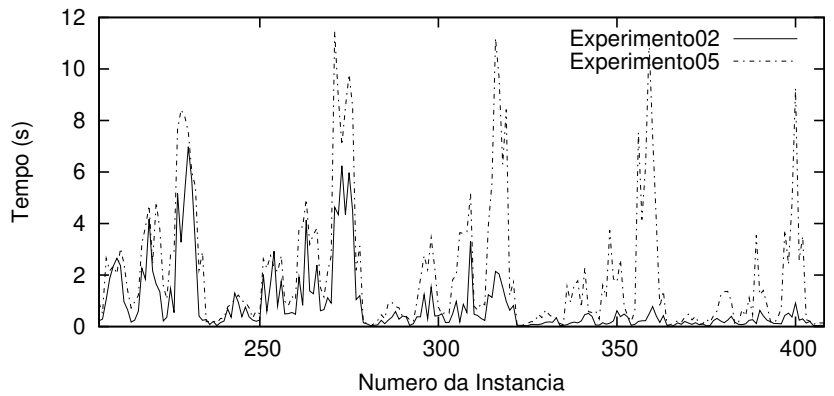
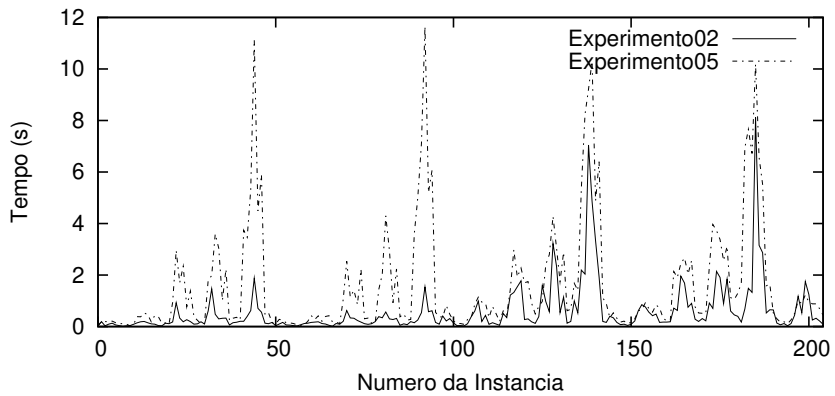


Figura 5.5: Comparação entre os tempos do Experimento05 e Experimento02.

no modelo e leveza (para fins de resolução) das relaxações lineares. Desta forma, conjecturamos que talvez pudéssemos melhorar ainda mais esse equilíbrio começando com uma formulação mais fraca (Clique-IS) e fortalecendo as restrições à medida em que elas ficassem ativas. Para tanto, utilizando os procedimentos de *lifting* já discutidos na Seção 5.1.2 (*lifting a posteriori*).

Para testar se a conjectura era verdadeira, realizamos dois experimentos. No primeiro deles, executamos um algoritmo de **B&B** com a formulação Clique-IS utilizando um procedimento de *lifting a posteriori* de uma única restrição ativa por iteração (Experimento06) e comparamos os resultados com o Experimento02 (que teve o melhor resultado dentre os experimentos anteriores). O segundo experimento foi realizado de forma semelhante, todavia, executou-se o *lifting* de todas as restrições ativas por iteração (Experimento07). Novamente, comparamos os resultados com aqueles obtidos pelo melhor algoritmo de acordo com os Experimento02 e Experimento06.

Os Experimento06 e Experimento07 mostraram que a conjectura estava errada. Ou seja, a utilização de *lifting a posteriori* não melhora o desempenho do algoritmo. Comparando com o Experimento02 os novos experimentos foram melhores, respectivamente, em apenas 21,39% e 20,57% dos casos, sendo que a piora nos tempos foi, em média, de 30,06% e 29,39%. Apesar das desigualdades serem fortalecidas quando necessário e do tempo gasto com o *lifting* ser relativamente baixo (em média 7,71% do tempo total no Experimento06 e 8,01% no Experimento07), esses experimentos apresentam um aumento, em média, de 10,06% e 7,71%, respectivamente, no Experimento06 e no Experimento07, no número de nós explorados. Embora do número de nós, em geral, não ser um fator determinante no tempo de resolução (o número de nós explorados em um experimento pode ser muito maior do que o de outro, e ainda assim ter o tempo total menor se o primeiro gasta pouco tempo em cada nó e o segundo gasta muito tempo por nó), neste caso, essa correlação parece ser verdadeira, o que parece explicar o desempenho superior do **B&B** do Experimento02. Nas Figuras 5.6 e 5.7 pode-se observar as diferenças entre os tempos obtidos no Experimento06 e Experimento02 e entre os tempos obtidos no Experimento07 e Experimento02, respectivamente.

Tendo confirmado que a melhor formulação é a Clique-IS e que a utilização de procedimentos de *lifting* não melhora o desempenho do algoritmo, partimos para os experimentos com os algoritmos de **B&C**. Um fato bem conhecido na literatura é que algoritmos de planos de corte podem ser bastante úteis para melhorar o limitante dual e como os testes anteriores mostraram que grande parte do tempo era gasto para encontrar um limitante dual justo - no Experimento02 o tempo gasto para encontrar o melhor limitante primal foi em média 66,19% do tempo total e esse valor cai para 44,26% quando consideramos apenas as instâncias que tiveram um tempo total superior a 30s - essa abordagem pareceu bastante promissora para melhorar a eficiência do algoritmo. Assim, realizamos três expe-

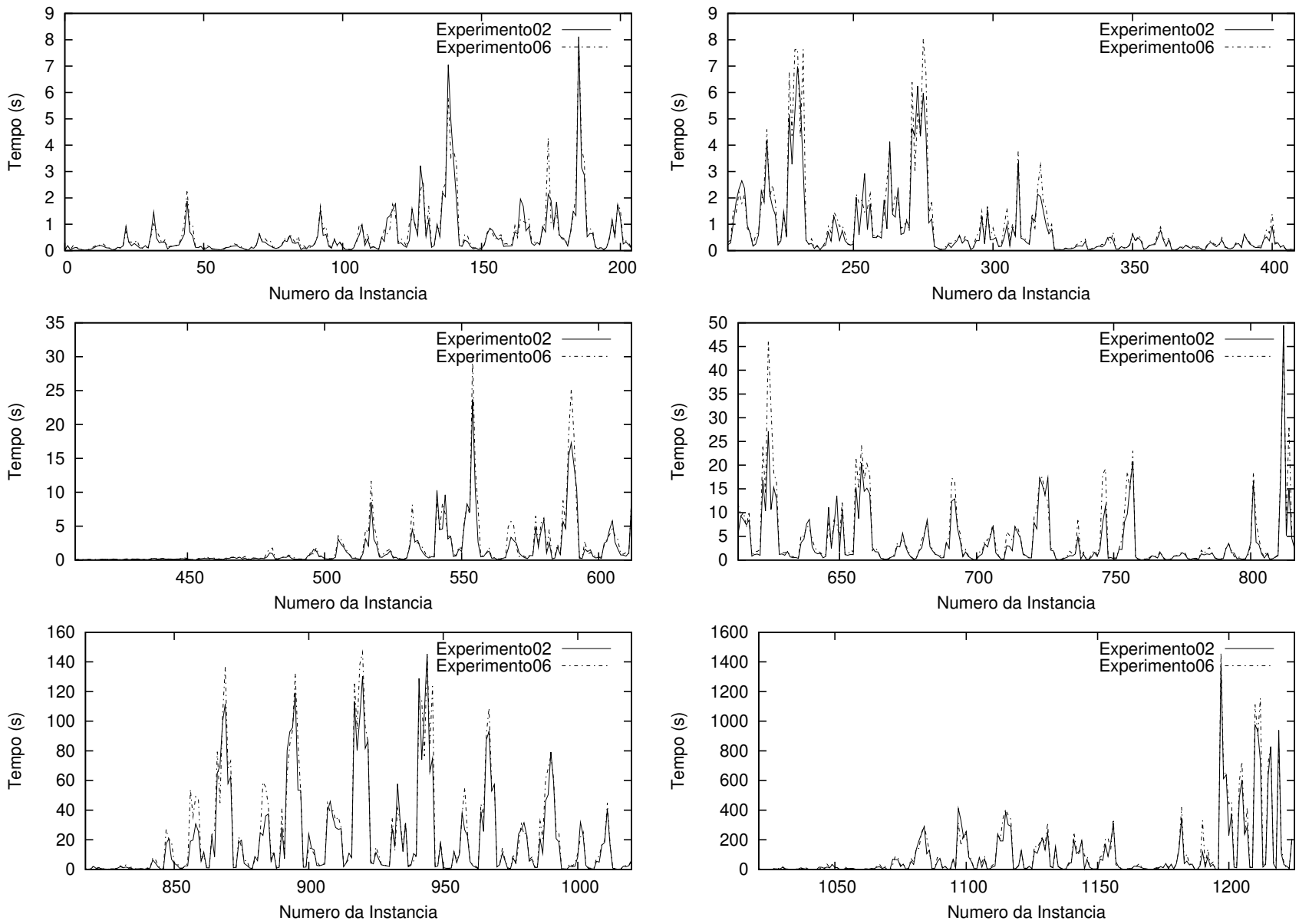


Figura 5.6: Comparação entre os tempos do Experimento06 e Experimento02.

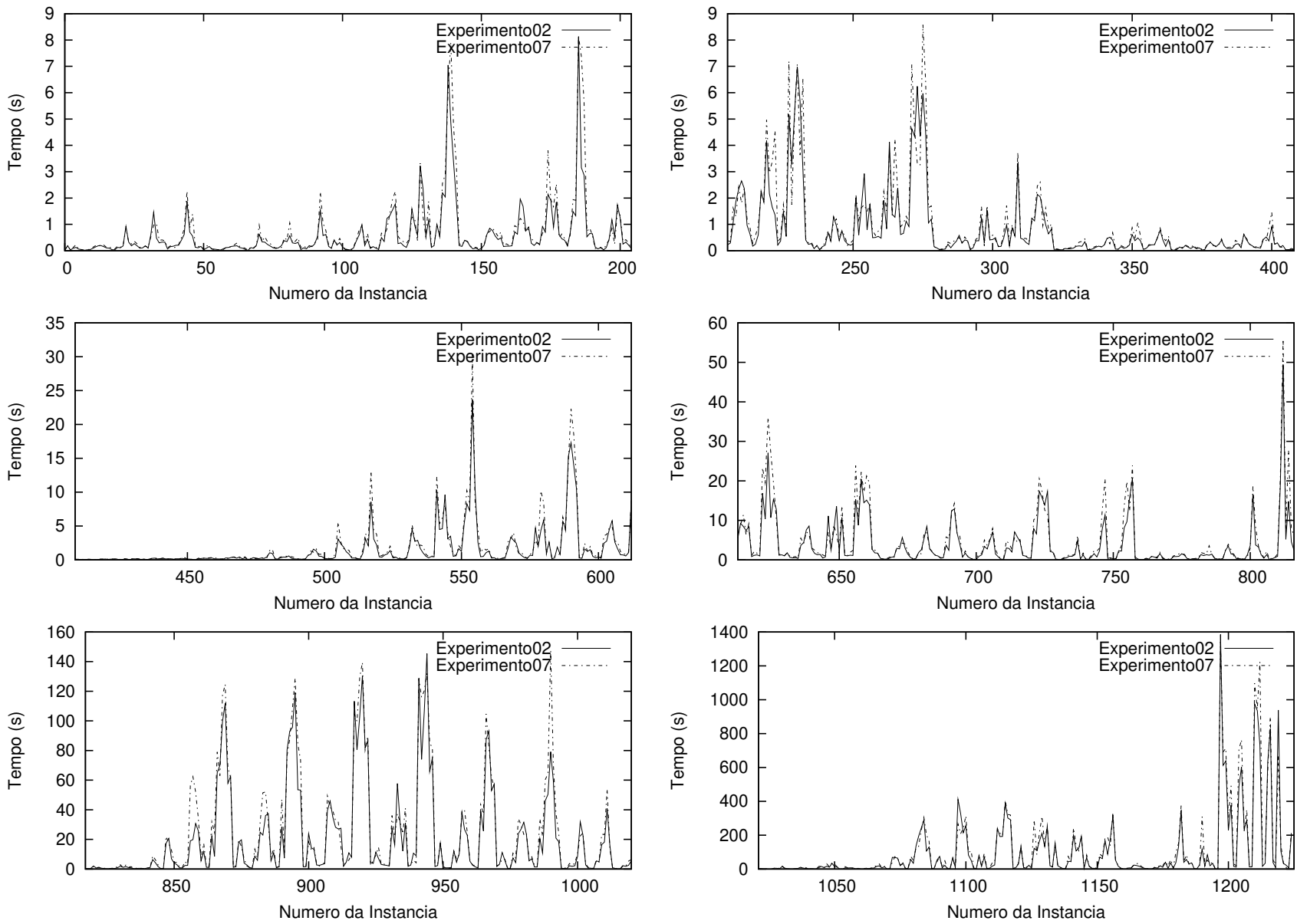


Figura 5.7: Comparação entre os tempos do Experimento07 e Experimento02.

rimentos utilizando diferentes rotinas de separação explicadas na Seção 5.1.3. O primeiro experimento utilizou a rotina de separação para as desigualdades (4.2) (Experimento08), o segundo utilizou as rotinas de separação relacionadas às desigualdades (3.27) a (3.34) (Experimento09) e o terceiro utilizou a rotina de separação para desigualdades (3.11) e (3.12) (Experimento10). Nos três experimentos o número de iterações de separação por nó foi limitado a 10 e o número máximo de cortes inseridos, limitado a $10nGnH$, ou seja, dez vezes o número de variáveis do problema. Da mesma forma que foi feito nos casos anteriores, os resultados de cada experimento foram comparados com aqueles do experimento anterior que apresentou melhor desempenho.

O Experimento08 em relação ao Experimento02 apresentou um tempo melhor em apenas 18,69% dos casos, tendo uma piora, em média, de 54,50%. Esses dados são curiosos, visto que foi utilizada uma rotina de separação para adicionar cortes que definem facetas - o que deveria agilizar a descoberta do limitante dual ótimo - e que consome um tempo relativamente baixo (apenas 1,84% do tempo total, em média), mesmo inserindo uma quantidade razoável de cortes (em média, 126,61% do número de restrições da formulação). Além disso, houve uma redução no número de nós explorados, em média, de 35,14% em relação ao Experimento02. Assim, a piora dos tempos do Experimento08 só pode ser explicada pelo tempo gasto na resolução das relaxações lineares. Uma visão comparativa entre os tempos desses dois experimentos pode ser obtida observando a Figura 5.8.

Já os Experimento09 e Experimento10 também apresentaram resultados piores que o Experimento02: apenas 4,90% dos casos apresentaram tempos melhores, com piora média de 9,69% no Experimento09 e 32,98% dos casos foram melhores no tempo, com uma piora média de 9,85% no Experimento10. Nos dois experimentos, os tempos gastos com as rotinas de separação também foram relativamente baixos, 7,85% e 1,52% dos tempos totais, respectivamente. Em média, o número de cortes inseridos foi, respectivamente, da ordem de 0,17% e 22,38% do número de restrições da formulação para os Experimento09 e Experimento10. Em ambos experimentos houve um pequeno aumento no número de nós explorados em relação ao Experimento02: 1,12% para o Experimento09 e 3,65% para o Experimento10. Este último fato, juntamente com o pequeno número de cortes inseridos, pode ajudar a explicar o aumento nos tempos. Porém, como este último foi mais significativo que o incremento no número de nós e que o tempo gasto com cortes, provavelmente também houve um incremento na complexidade de resolução das relaxações lineares com a inclusão dos cortes. As Figuras 5.9 e 5.10 dão uma idéia da relação entre os tempos destes experimentos com o Experimento02.

Como já comentamos na Seção 5.1.4, uma questão importante para o bom desempenho de um algoritmo de **B&B** é a questão da seleção de variáveis. Desta forma, realizamos experimentos para testar qual critério de seleção é o mais eficiente: primeiramente testamos a seleção de variáveis por grau (Experimento11), em seguida a seleção de variáveis

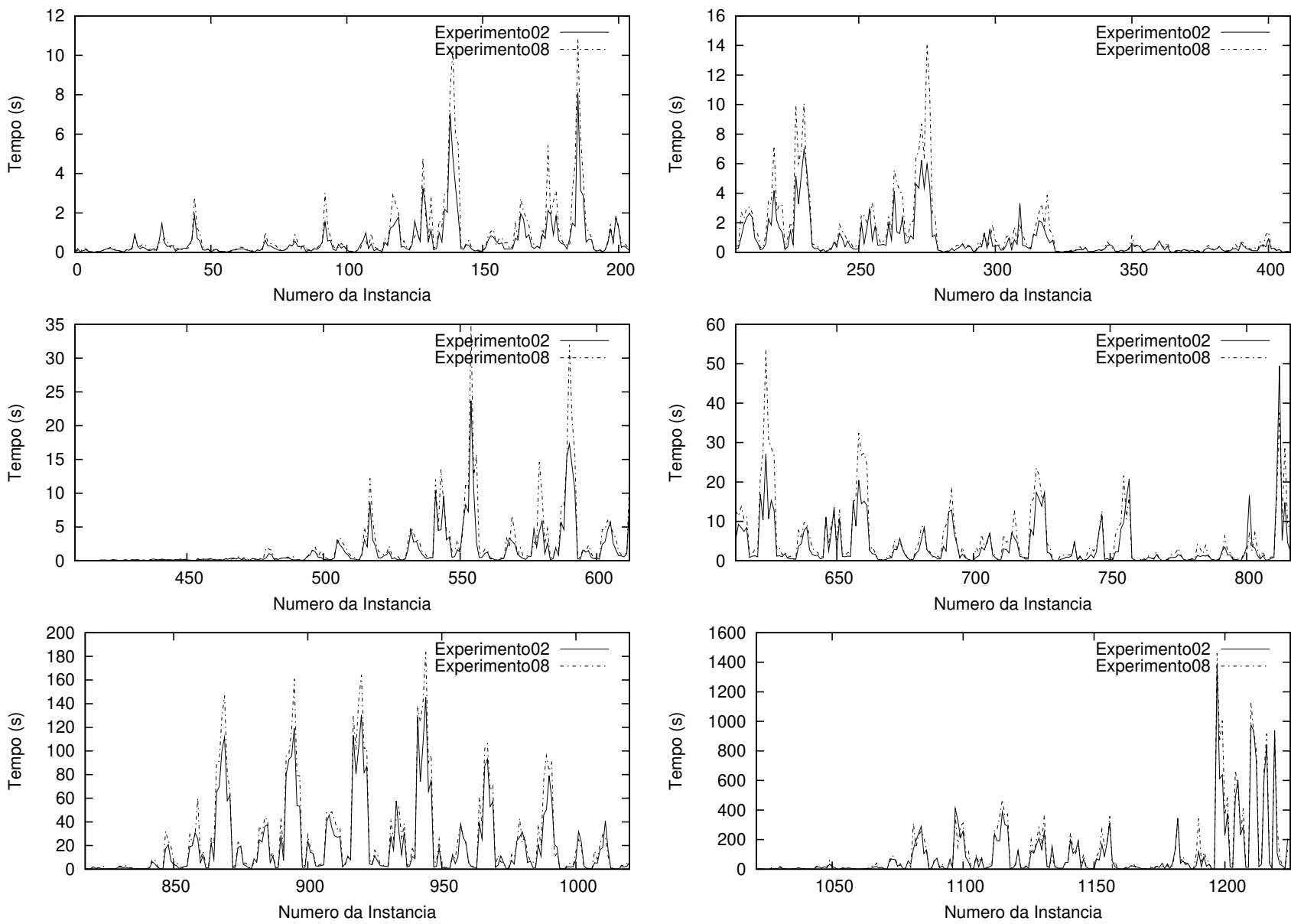


Figura 5.8: Comparação entre os tempos do Experimento08 e Experimento02.

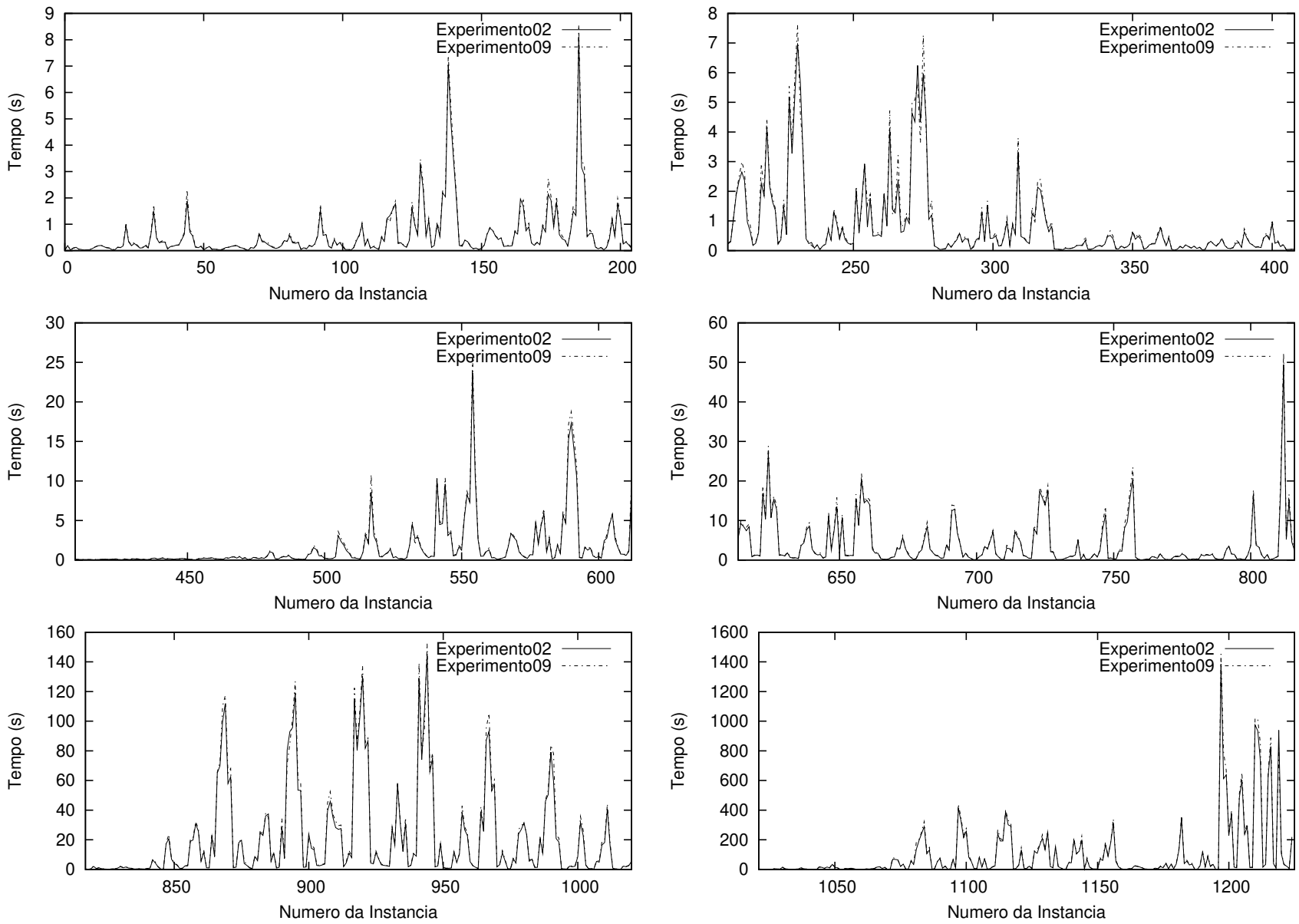


Figura 5.9: Comparação entre os tempos do Experimento09 e Experimento02.

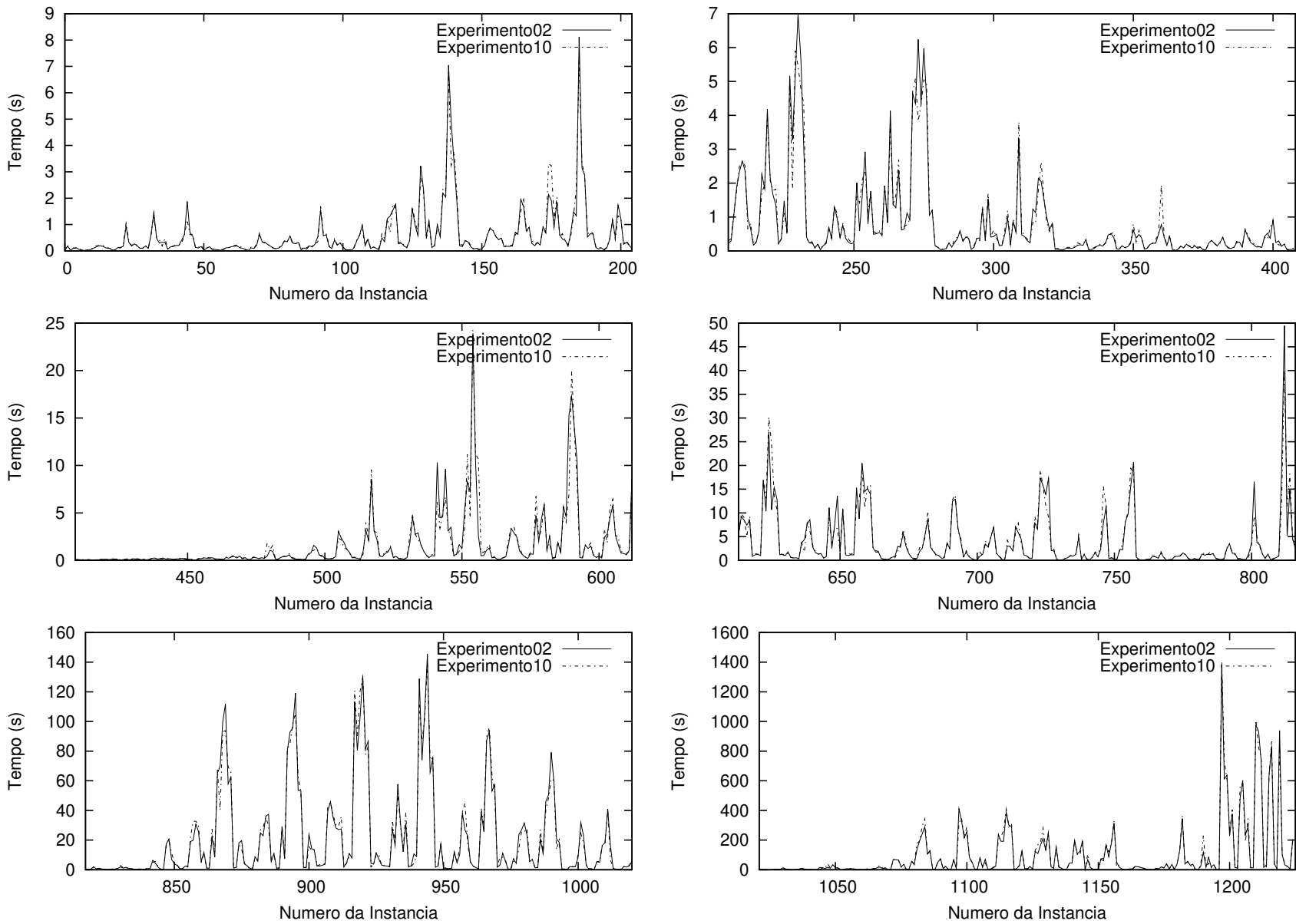


Figura 5.10: Comparação entre os tempos do Experimento10 e Experimento02.

por cores (Experimento12), posteriormente a seleção por grau de mapeamento (Experimento13) e, por último, a seleção por variável mais violada (Experimento14). Nestes experimentos, foi utilizada a mesma configuração do Experimento02, apenas adicionando algum critério de seleção de variáveis. A comparação entre os experimentos foi feita como anteriormente.

Analisando os resultados do Experimento11, constatamos que a seleção de variáveis por grau de fato melhora o desempenho do algoritmo de **B&B**. Neste experimento, o tempo de resolução foi menor em 57,71% dos casos, tendo uma melhora, em média, de 2,06% em relação ao Experimento02. Uma comparação dos tempos dos dois experimentos pode ser vista na Figura 5.11. O Experimento11 também apresentou melhores resultados que o Experimento12, com tempos menores que este em 58,94% dos casos, sendo que, em média, o tempo foi 9,65% melhor. A Figura 5.12 mostra uma comparação dos tempos dos Experimento11 e Experimento12.

A seleção de variáveis por grau de mapeamento testada no Experimento13 se mostrou ainda mais eficiente que a seleção de variáveis por grau. Em 61,47% dos casos o Experimento13 apresentou tempos melhores que o Experimento11, com uma melhora média de 2,01%. O Experimento13 também apresentou tempos melhores que os do Experimento14 em 64,49% dos casos, sendo que no Experimento14, a piora dos tempos em relação ao Experimento13 foi, em média, de 152,73%. Podemos comparar os tempos do Experimento13 com os do Experimento11 observando a Figura 5.13 e com os do Experimento14 observando a Figura 5.14. Não sabemos ao certo o porquê do melhor desempenho do Experimento13 em relação aos demais experimentos com outros métodos de seleção de variáveis.

Na Seção 5.1.5 dissemos que durante o processo de **B&B** com ramificação binária, quando a variável de ramificação é fixada em 1 (um) diversas variáveis são fixadas, porém, quando a variável de ramificação é fixada em 0 (zero), nenhuma outra variável é fixada. Por este motivo pensamos que utilizar uma estratégia de ramificações múltiplas poderia ajudar a melhorar a eficiência do algoritmo e realizamos um experimento para verificar se isso de fato ocorre (Experimento15). Neste experimento além da estratégia de ramificações múltiplas, também foi utilizada a seleção de variáveis por grau de mapeamento, já que esta se mostrou a forma de seleção mais eficiente. Entretanto, comparando os resultados deste experimento com os do Experimento13, observamos uma melhora do tempo em apenas 4,57% dos casos com uma piora, em média, de 109,47%. O aumento do tempo pode ser atribuído ao número de nós adicionais que são explorados devido à estratégia de ramificações múltiplas (houve um aumento de 174,33% no número de nós do Experimento15 em relação ao Experimento13). Os tempos desses dois experimentos podem ser melhor comparados observando a Figura 5.15.

Em testes preliminares, que fizemos, o resolvedor Cliquer (ver Seção 4.3) se mostrou

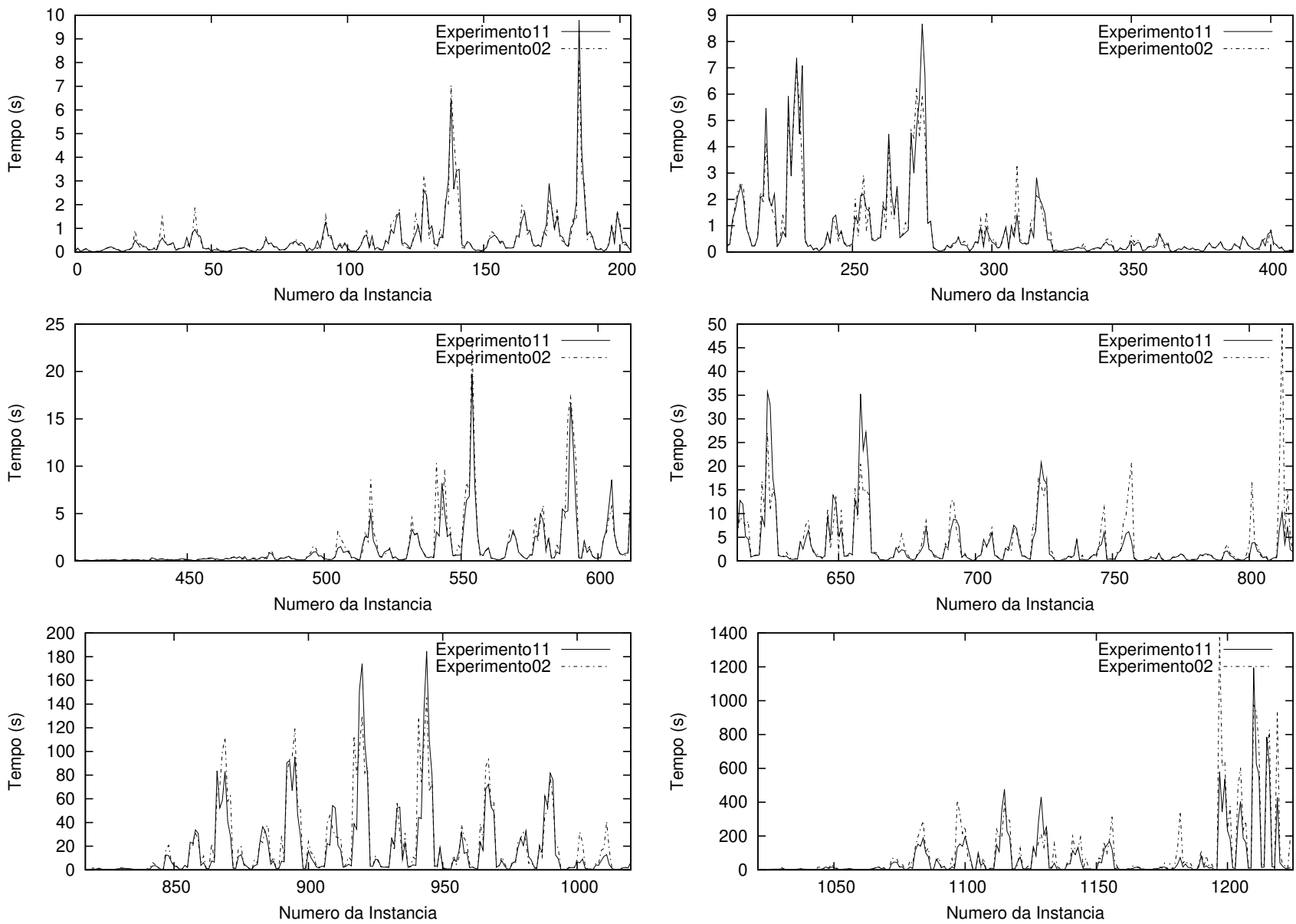


Figura 5.11: Comparação entre os tempos do Experimento11 e Experimento02.

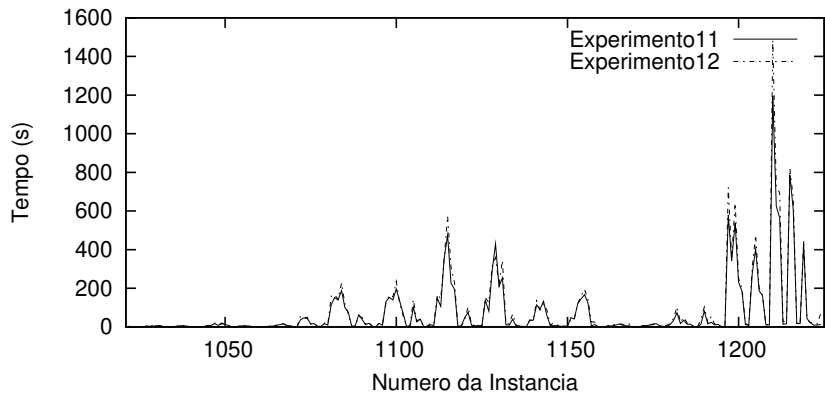
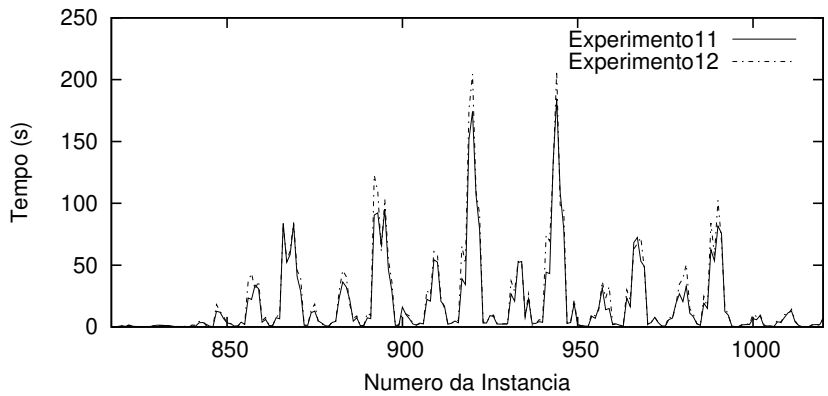
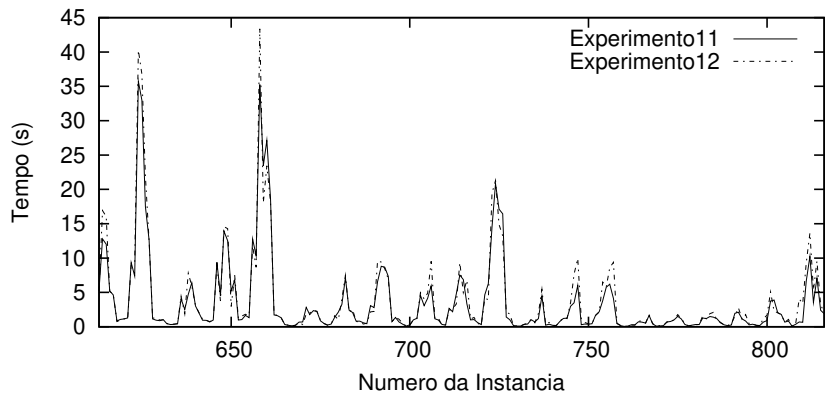
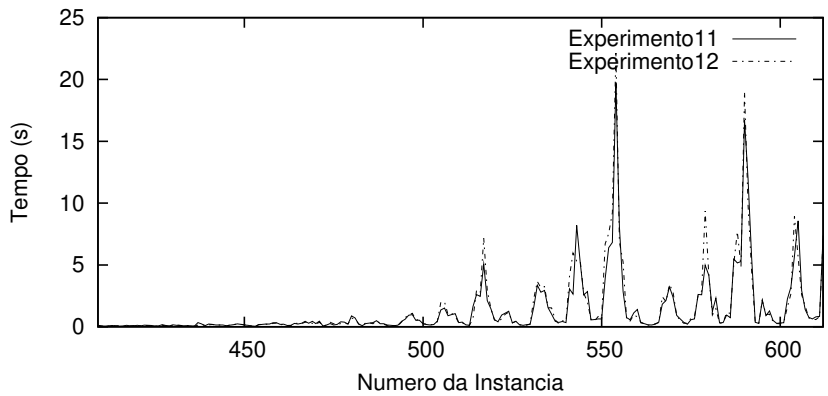
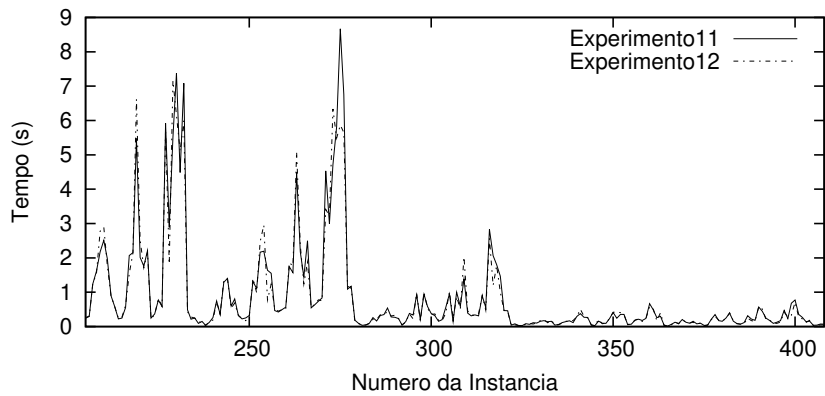
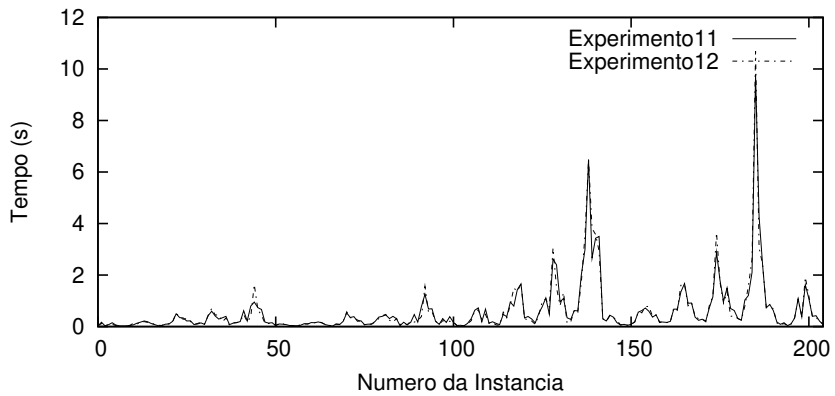


Figura 5.12: Comparação entre os tempos do Experimento12 e Experimento11.

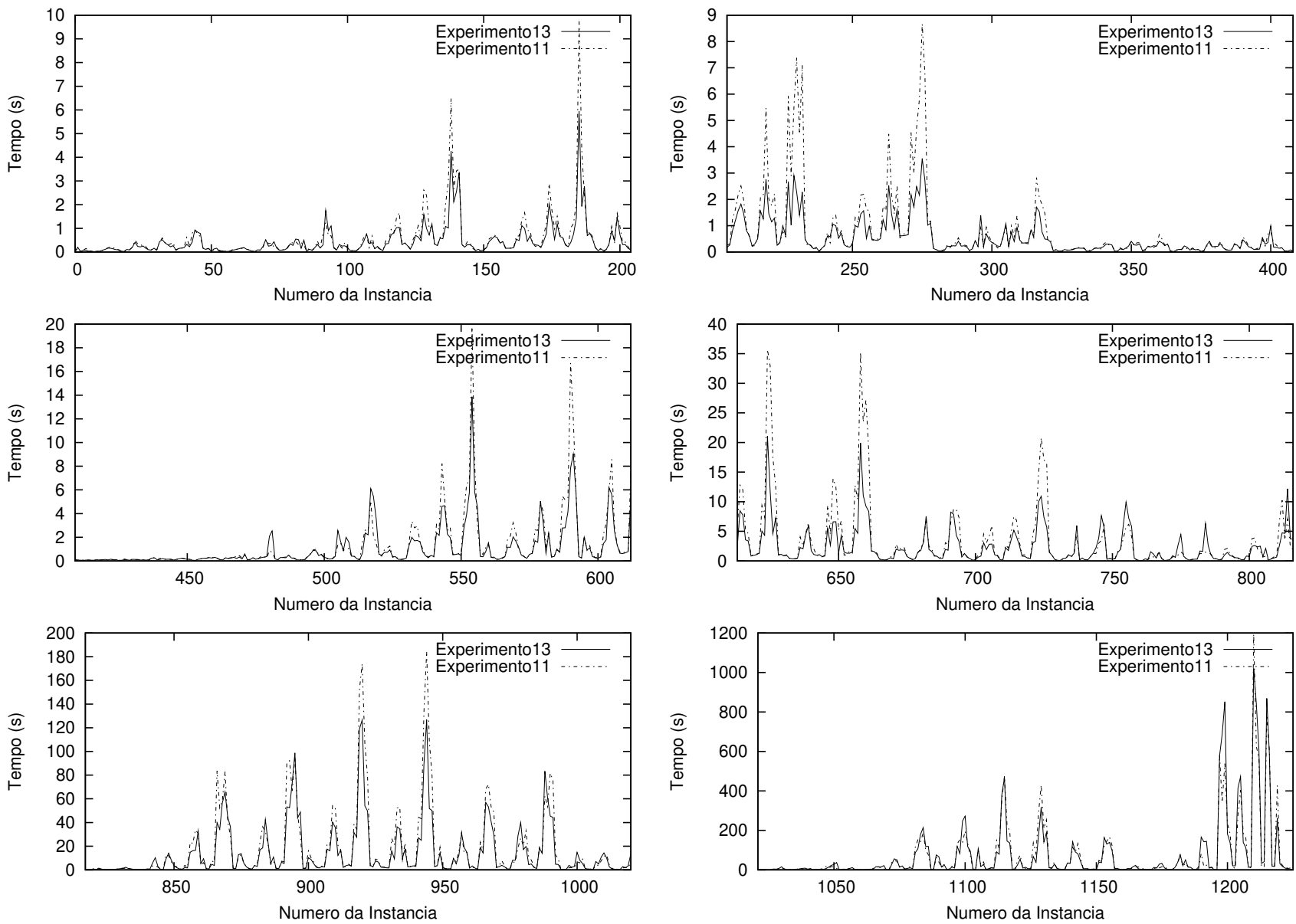


Figura 5.13: Comparação entre os tempos do Experimento13 e Experimento11.

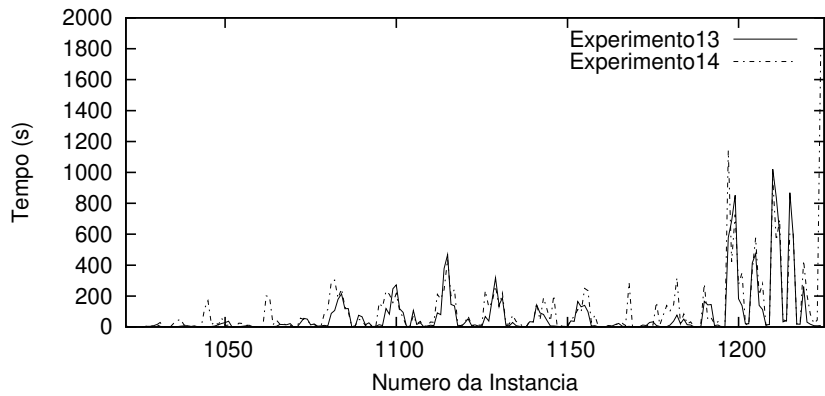
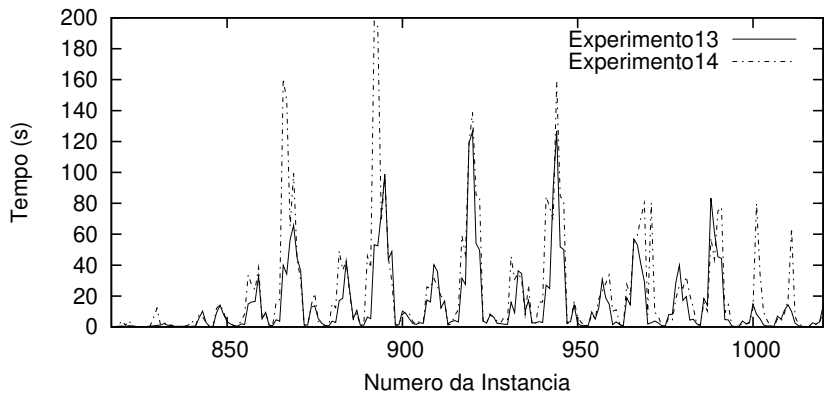
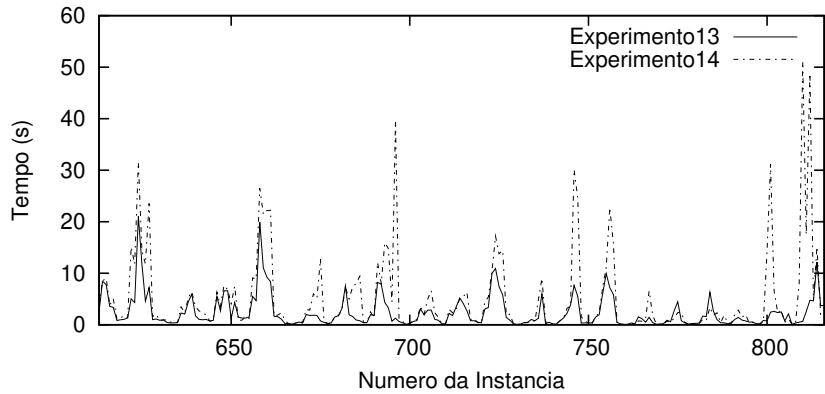
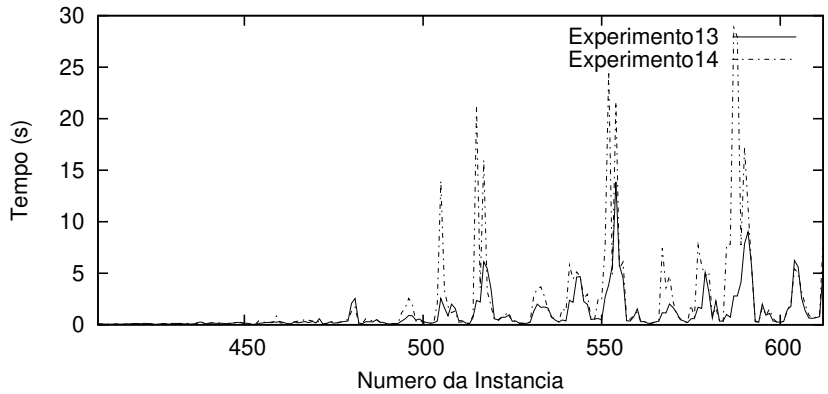
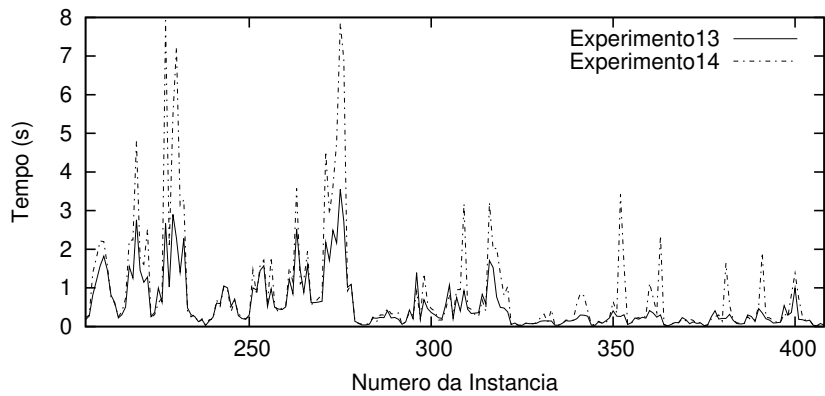
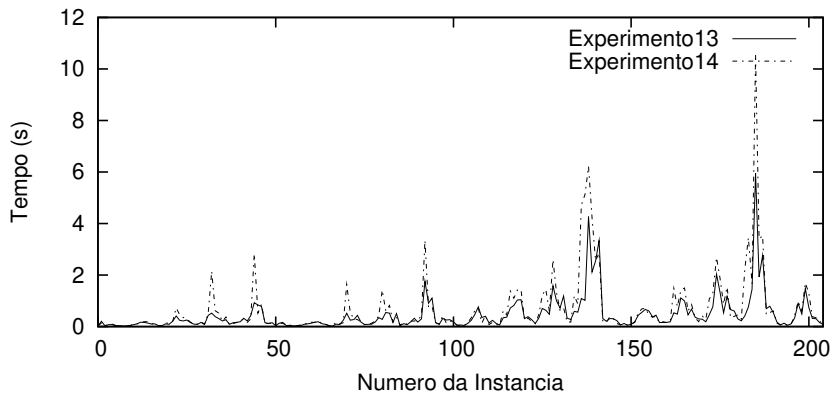


Figura 5.14: Comparação entre os tempos do Experimento14 e Experimento13.

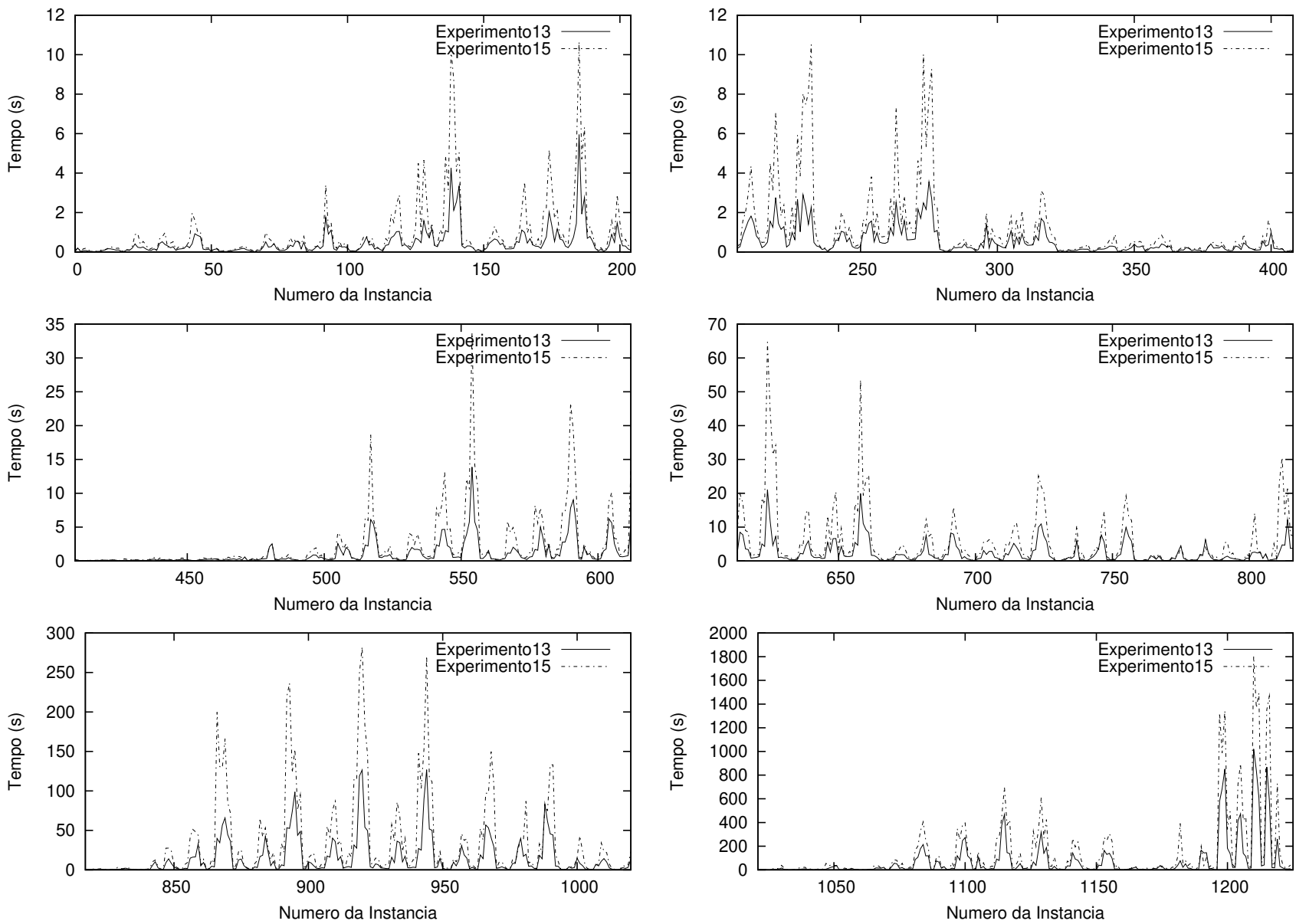


Figura 5.15: Comparação entre os tempos do Experimento15 e Experimento13.

muito rápido. Assim, é natural pensar que pelo menos algumas das idéias usadas por esse algoritmo possam ser usadas para melhorar os nossos algoritmos baseados em **PLI**. Pensando nisso, implementamos algumas dessas idéias da forma descrita na Seção 5.1.6 e essas implementações foram testadas primeiro usando a estratégia de adicionar uma única variável por subproblema (Experimento16) e em seguida utilizando a estratégia de adicionar um conjunto de variáveis (que representavam um IS no grafo associação) por subproblema (Experimento17).

Ambos experimentos usando estratégias do Cliquer se mostraram muito menos eficientes que o Experimento13. No Experimento16, apenas 1,55% dos testes foram mais rápidos que o do Experimento13, havendo uma piora de, em média, 958,63% em relação ao tempo deste. Além disso, neste experimento o tempo de preparação não foi irrisório chegando a um valor máximo de 14,296s. O Experimento17, apesar de um pouco melhor que o Experimento16, também não apresentou bons resultados. Em apenas 2,04% dos casos ele foi mais rápido que o Experimento13, tendo uma piora nos tempos de, em média, 186,10%. As Figuras 5.16 e 5.17 mostram comparações entre os tempos do Experimento13, respectivamente, com os Experimento16 e Experimento17.

O último experimento que realizamos foi para comparar o desempenho do algoritmo que desenvolvemos, configurado da melhor forma possível, com o Cliquer (Experimento18). Em relação ao Experimento13, em 99,67% dos casos, o tempo de resolução do Cliquer foi melhor e a melhora do tempo foi de 95,34%, em média. A superioridade do Cliquer no tempo de resolução, apesar de todas as técnicas empregadas para melhorar a eficiência do algoritmo de **B&B**, nos leva a crer que para problemas como este, algoritmos combinatórios são mais eficientes do que aqueles baseados em **PLI** (pelo menos no caso em que as formulações são baseadas nas mesmas variáveis usadas por nós). Na Figura 5.18 podemos ter uma idéia mais clara da diferença entre os tempos dos dois experimentos (os tempos do Experimento18 são tão pequenos que a linha que os representa quase não aparece nos gráficos).

Obviamente, o fato do Cliquer ter obtido um desempenho tão bom com essas instâncias não significa que ele consegue resolver qualquer instância de tamanho razoável em um tempo aceitável. Afinal, o problema da Clique é \mathcal{NP} -difícil e, portanto, o Cliquer possui complexidade exponencial no pior caso. Para se ter uma idéia, em um teste isolado com uma instância formada por dois grafos gerados aleatoriamente com 45 vértices cada um e densidade de aproximadamente 50%, o Cliquer levou aproximadamente 1.435,61s para encontrar uma clique máxima no grafo associação formado por estes dois grafos (evidentemente, nosso algoritmo de **B&B** se saiu muito pior, não tendo conseguido resolver o problema à otimalidade após 12 horas).

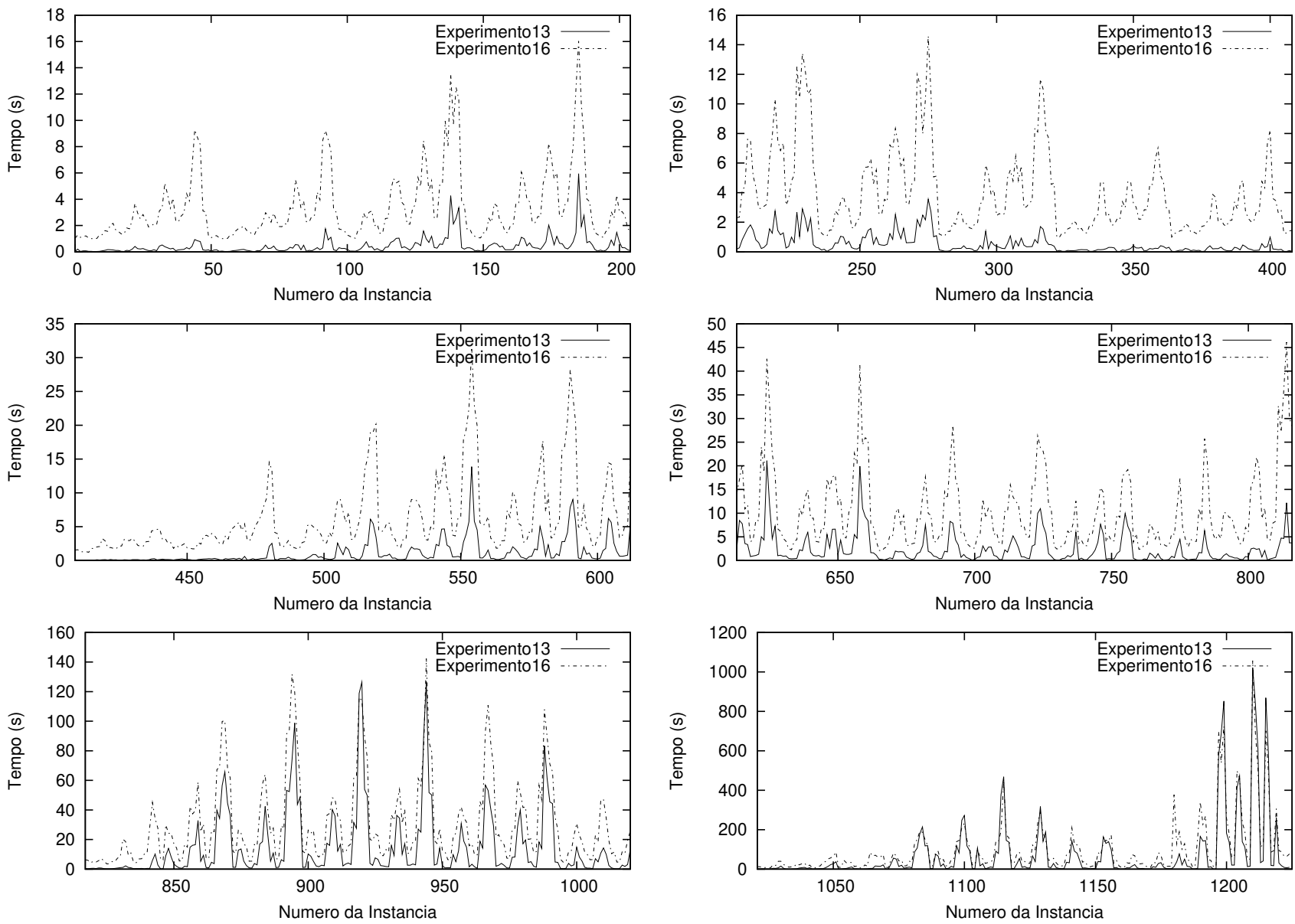


Figura 5.16: Comparação entre os tempos do Experimento16 e Experimento13.

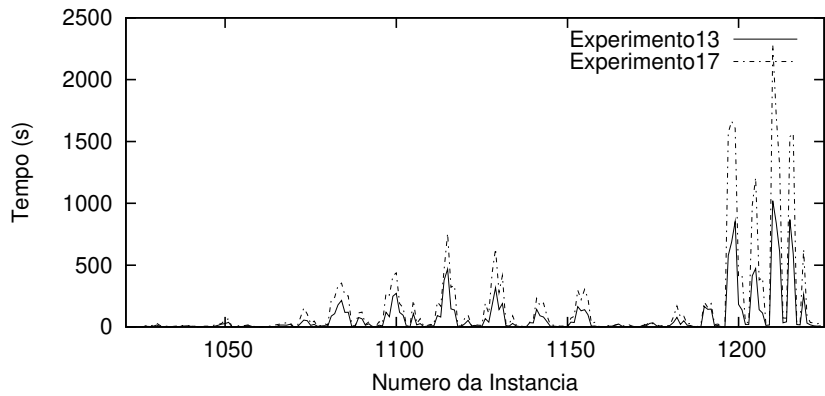
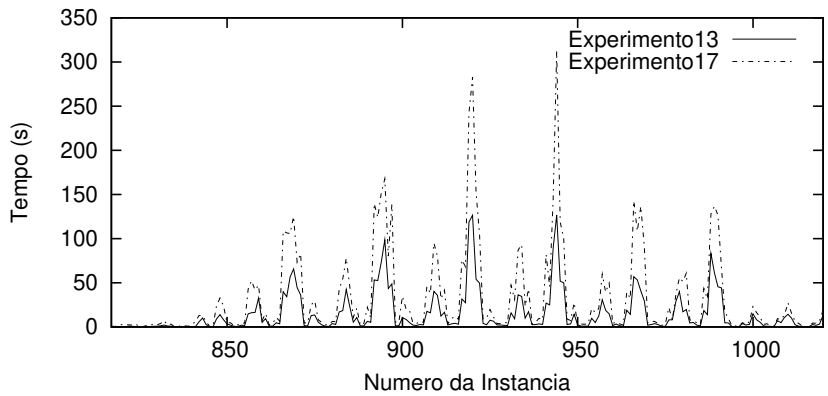
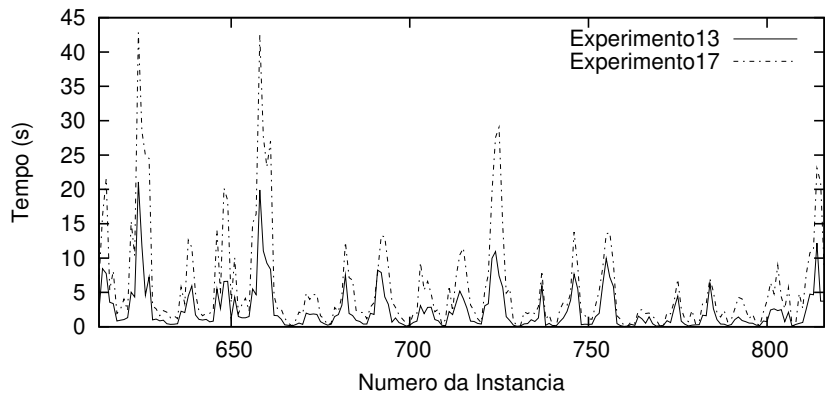
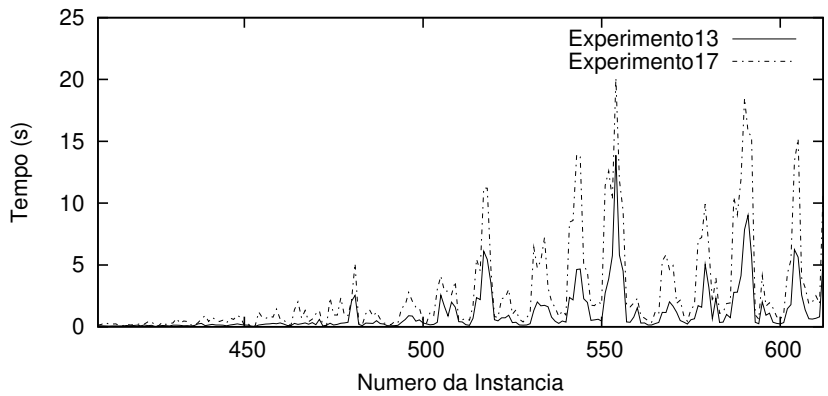
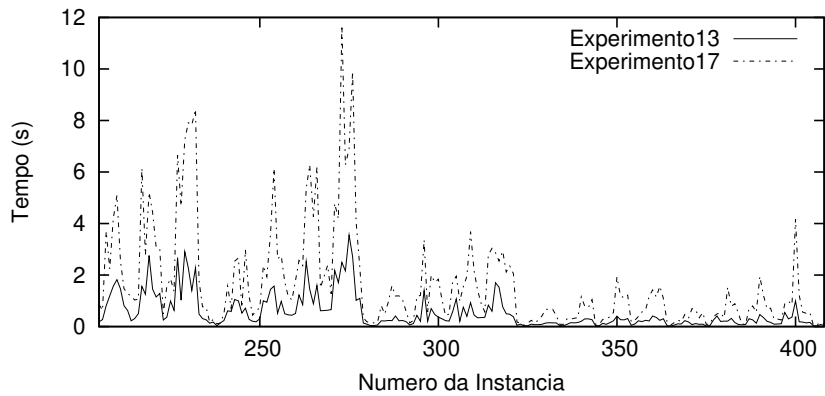
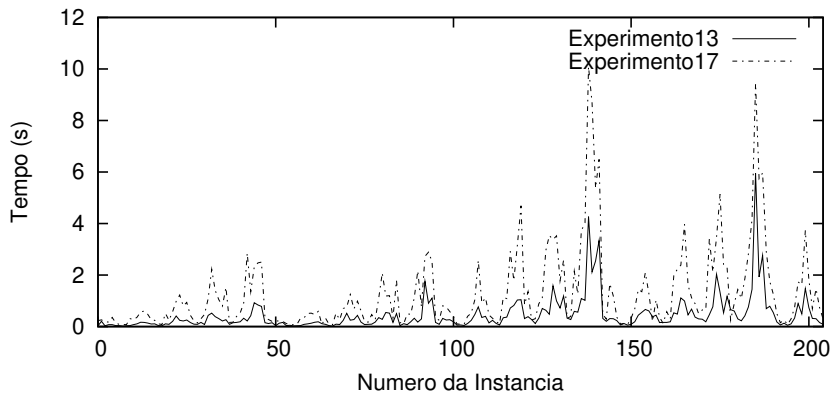


Figura 5.17: Comparação entre os tempos do Experimento17 e Experimento13.

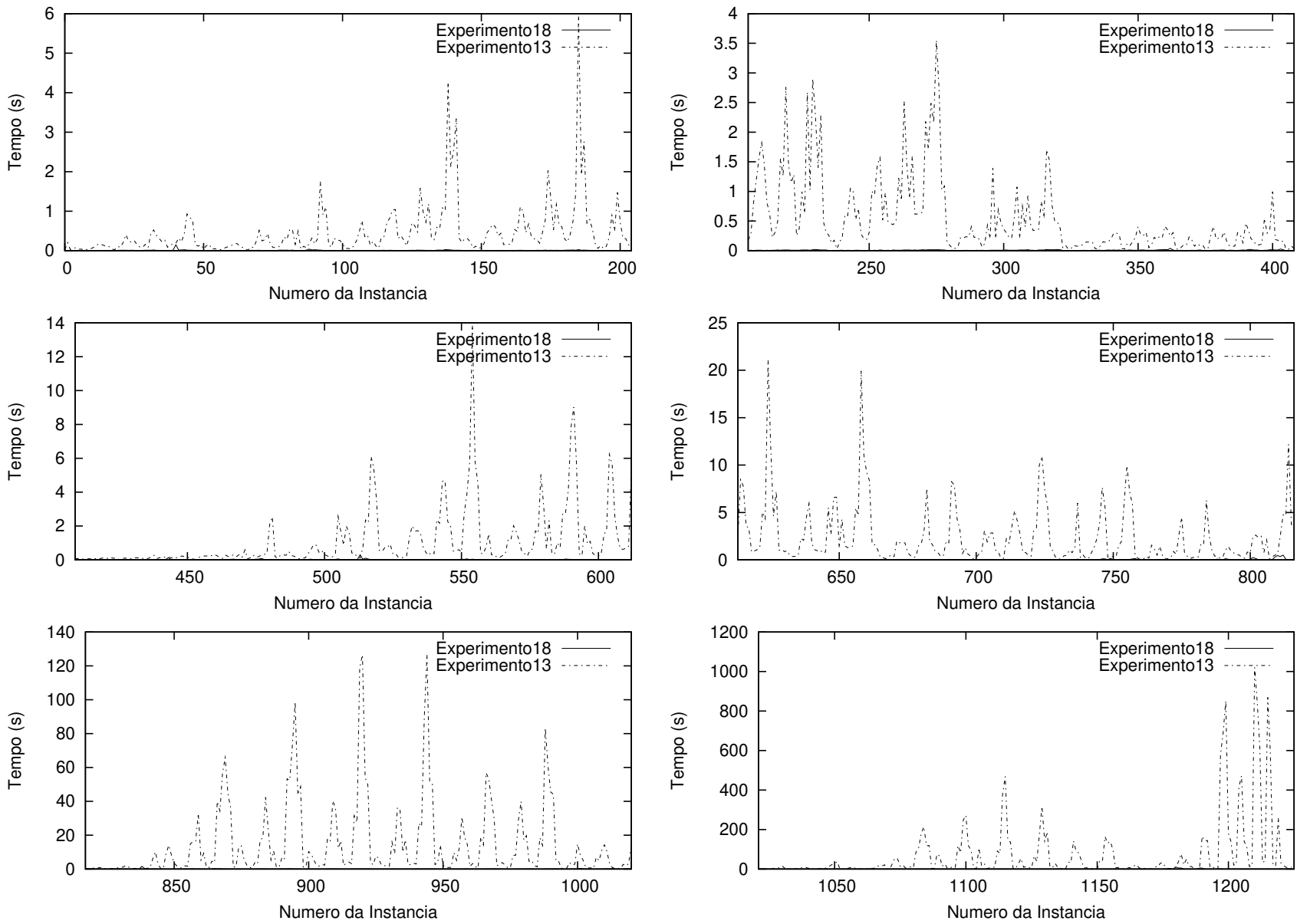


Figura 5.18: Comparação entre os tempos do Experimento18 e Experimento13.

Capítulo 6

Conclusões

No estudo teórico que realizamos sobre o poliedro do **MSIC**, encontramos diversas desigualdades válidas e provamos que algumas delas representam facetas. Além disso, provamos a equivalência entre uma formulação para o **MSIC** e outra para o problema da clique máxima. Com isso, a rigor, podemos utilizar todas as desigualdades válidas conhecidas para o problema da clique para resolver o **MSIC** e vice-versa. Porém, as desigualdades de mapeamento único e de arestas do **MSIC** que são equivalentes a desigualdades de IS no problema da clique, são mais facilmente descritas neste último problema.

Experimentos computacionais comparando a formulação Clique-IS com a formulação MaxIS Light (que é na verdade uma formulação para o problema da clique), mostrou que apesar das desigualdades para ambos os problemas pertencerem a uma mesma classe, escolher as desigualdades que farão parte da formulação inicial baseando-se nas desigualdades do **MSIC** produz melhores resultados que escolhê-las baseando-se em desigualdades IS quaisquer. Isso acontece porque apesar da envoltória convexa ser equivalente e, neste caso específico, as desigualdades do problema da clique utilizadas serem mais fortes que as utilizadas do **MSIC** (as desigualdades IS representam facetas ao contrário das utilizadas na formulação Clique-IS), quando escolhemos as desigualdades baseando-nos no problema do **MSIC**, provavelmente, utilizamos mais desigualdades ativas na solução ótima do que quando fazemos a escolha baseando-nos em informações genéricas do problema da clique. Assim, apesar da caracterização das desigualdades ser mais difícil no **MSIC**, vale a pena utilizá-las na resolução deste problema.

Outras desigualdades que provamos ser equivalentes foram as desigualdades relacionadas a *odd-holes*. Neste caso não fomos capazes de chegar a desigualdades que representem facetas. Contudo, as desigualdades que encontramos para o **MSIC** são mais fortes que as desigualdades sem *lifting* já conhecidas para o problema da clique. É bem possível que se possa projetar um procedimento de *lifting* capaz de fortalecer as desigualdades relacionadas a *odd-holes* encontradas no **MSIC** que leve a desigualdades que representem facetas

para este problema. Entretanto, tal investigação não foi explorada nesta dissertação.

Nos experimentos computacionais testamos diversas técnicas para obter um algoritmo baseado em **PLI** e que fosse eficiente. Isso incluiu testes com várias formulações, implementações de rotinas *lifting*, desenvolvimento de um **B&C**, estudos de diferentes critérios para seleção de variáveis para ramificação e avaliação de estratégia de ramificação alternativas, como ramificações múltiplas e até a simulação da estratégia empregada pelo Cliquer (um algoritmo combinatório conhecido para resolver o problema da clique). No entanto, o algoritmo que apresentou o melhor resultado foi aquele que utiliza formulação Clique-IS, sem *lifting*, sem algoritmos de planos de cortes, seleção de variáveis baseada no grau de mapeamento e com ramificação binária. Isto é, um algoritmo que utiliza técnicas mais básicas e uma formulação mais fraca que outras. Uma possível explicação para esse algoritmo apresentar um melhor desempenho que outros que utilizam técnicas mais elaboradas e formulações mais fortes é que essa formulação é mais leve. Com isso, as relaxações lineares são resolvidas mais rapidamente que em outras formulações. Além disso, a melhora do limitante dual obtida com procedimentos de *lifting* e algoritmos de planos de corte não compensam o aumento da dificuldade de computar as relaxações lineares associadas à formulação. Também, o aumento no número de nós ativos quando utilizamos uma estratégia de ramificações múltiplas não se mostrou vantajoso quando comparado à fixação pura e simples de uma única variável.

A comparação do melhor algoritmo baseado em **PLI** que obtivemos com o Cliquer mostrou que este último é muito mais eficiente para as instâncias testadas. Para que algoritmos baseados em **PLI** (utilizando uma formulação baseada nas mesmas variáveis usadas por nós) tivessem alguma chance de vencer um algoritmo combinatório como este, seria necessário conhecer mais desigualdades que estivessem ativas na solução ótima do problema. Obviamente, descobrir desigualdades com estas características não é uma tarefa fácil. Uma abordagem que pode ser utilizada nesta direção é estudar a envoltória convexa completa correspondente a pequenas instâncias do **MSIC**, identificando desigualdades ativas na solução ótima. Em seguida, deve-se procurar generalizar as desigualdades assim encontradas para instâncias maiores. Vale observar que esta metodologia é exequível pois existem códigos disponíveis na rede [21] que permitem calcular a envoltória convexa completa de pequenas instâncias de problemas combinatórios. Um outro modo de buscar boas desigualdades para o **MSIC** é tirando proveito da equivalência mostrada no Capítulo 4 entre este problema e o problema da clique máxima. Para tanto, poderíamos especializar a investigação poliedral do problema da clique máxima para instâncias que têm as características peculiares dos grafos associação resultantes das instâncias do **MSIC**. Em particular, estamos pensando no fato de que estes grafos possuem $nG \times nH$ vértices e são, ao mesmo tempo, nG e nH partidos, onde nG e nH são, respectivamente, os números de vértices dos grafo G e H passados na entrada do **MSIC**.

Referências Bibliográficas

- [1] F. Barahona, M. Grötschel, M. Jünger, and R. Reinelt. An application of combinatorial optimization to statistical physics and circuit layout design. *Operations Research*, 36:493–513, 1988.
- [2] J. A. Bondy and U.S.R. Murty. *Graph Theory With Applications*. Elsevier Science, 1976.
- [3] R. Borndörfer. *Aspects of set packing, partitioning, and covering*. PhD thesis, Mathematics Department, Technischen Universtat, Berlin, 1997.
- [4] A. Caprara and M. Fischetti. Branch and cut algorithms. In M. Dell’Amico, F. Maffioli, and S. Martello, editors, *Annotated Bibliographies in Combinatorial Optimization*, chapter 4. Wiley, Chichester, UK, 1997.
- [5] L. Chen and W. Robien. Application of the maximal common substructure algorithm to automatic interpretation of ^{13}C NMR spectra. *Journal of Chemical Information and Computer Sciences*, 34:934–941, 1994.
- [6] D. Conte, C. Guidobaldi, and C. Sansone. A comparison of three maximum common subgraph algorithms on a large database of labeled graphs. In *Lecture Notes in Computer Science*, volume 2726, pages 130–141. Springer-Verlag, 2003.
- [7] H. P. Crowder, E. Johnson, and M. W. Padberg. Solving large-scale zero-one linear programming problems. *Operations Research*, 31:803–834, 1983.
- [8] Edmonds and Matula. private communication (4.2.2; a1.4), 1975.
- [9] B. Falkenhainer, K.D. Forbus, and D. Gentner. The structure-mapping engine: algorithms and examples. *Artificial Intelligence*, 34:1–63, 1989/90.
- [10] C. E. Ferreira and Y. Wakabayashi. *Combinatória Poliédrica e Planos-de-Corte Faciais*. X Escola de Computação, Campinas, julho de 1996.

- [11] M. R. Garey and D. S. Johnson. *Computer and intractability: A guide to the theory of NP-completeness*. Freeman, San Francisco, 1979.
- [12] E. Gifford, M. Johnson, D. Smith, and C. Tsai. Structure-reactivity maps as a tool for visualizing xenobiotic structure-reactivity. *Network Science*, 2:1–33, 1996.
- [13] M. Grötschel, M. Jünger, and R. Reinelt. A cutting plane algorithm for the linear ordering problem. *Operations Research*, 32:1195–1220, 1984.
- [14] M. Grötschel, C. L. Momma, and M. Stoer. Computational results with a cutting plane algorithm for designing communication networks with low connectivity constraints. *Operations Research*, 40:309–330, 1992.
- [15] R. Horaud and T. Skordas. Stereo correspondence through feature grouping and maximal cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11:1168–1180, 1989.
- [16] D. Johnson. A theoretician’s guide to the experimental analysis of algorithms. In D. S. Johnson M. H. Goldwasser and C. C. McGeoch, editors, *Data Structures, Near Neighbor Searches, and Methodology: Fifth and Sixth DIMACS Implementation Challenges*, pages 215–250, Providence, 2002. American Mathematical Society.
- [17] M. Jünger, G. Reinelt, and S. Thienel. Practical problem solving with cutting plane algorithms in combinatorial optimization. In W. J. Cook, L. Lóvasz, and P. Seymour, editors, *Combinatorial Optimization*, DIMACS Series in Discrete Mathematics in Computer Sciences. AMS, 1995.
- [18] V. Kann. On the approximability of the maximum common subgraph problem. In *Lecture Notes in Computer Science*, volume 577, pages 377–388. Springer-Verlag, 1992.
- [19] E. B. Krissinel and K. Henrick. Common subgraph isomorphism detection by backtracking search. *Softw. Pract. Exper.*, 34(6):591–607, 2004.
- [20] A. Lucena and J. Beasley. Branch and cut algorithms. In J. Beasley, editor, *Advances in Linear and Integer Programming*, chapter 5. Oxford University Press, Oxford, UK, 1996.
- [21] A. Löbel. Porta: Polyhedron representation transformation algorithm. disponível *online* (consultado em abril de 2007). <http://www.iwr.uni-heidelberg.de/groups/comopt/software/PORTA/>.

- [22] Gordana Manić. Modelagem matemática e aplicações de problemas de otimização relativos à busca de subgrafos com estruturas comuns. Primeiro Relatório Científico da FAPESP, junho de 2007. Bolsa de Pós-doutoramento, processo # 2006/01817-7.
- [23] J. J. McGregor. Backtrack search algorithms and the maximal common subgraph problem. *SOFTWARE-PRACT. AND EXPER.*, 12(1):23–34, 1982.
- [24] B. T. Messmer and H. Bunke. Decision tree approach to graph and subgraph isomorphism detection. *Pattern Recognition*, 32(12):1979–1998, 1999.
- [25] G. L. Nemhauser and G. Sigismondi. A strong cutting plane/branch-and-bound algorithm for node packing. *The Journal of the Operational Research Society*, 43(5):443–457, 1992.
- [26] G. L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley and Sons, New York, 1988.
- [27] M. Padberg and M. Grötschel. The symmetric traveling salesman problem: Polyhedral computations. In E. Lawler et al., editor, *The Traveling Salesman Problem*, pages 307–360. John Wiley & Sons, 1985.
- [28] M. Padberg and G. Rinaldi. Optimization of a 532-city traveling salesman problem by branch-and-cut. *Operations Research Letters*, 6:1–8, 1987.
- [29] M. Pelillo, K. Siddiqi, and S. W. Zucker. Matching hierarchical structures using association graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21:1105–1120, 1999.
- [30] J. W. Raymond, E. J. Gardiner, and P. Willett. Heuristics for similarity searching of chemical graphs using a maximum common edge subgraph algorithm. *Journal of Chemical Information and Computer Sciences*, 42(2):305–316, 2002.
- [31] J. W. Raymond, E. J. Gardiner, and P. Willett. RASCAL: Calculation of graph similarity using maximum common edge subgraphs. *The Computer Journal*, 45(6):631–644, 2002.
- [32] J. W. Raymond and P. Willett. Maximum common subgraph isomorphism algorithms for the matching of chemical structures. *Journal of Computer-Aided Molecular Design*, 16(7):521–533, 2002.
- [33] K. Shearer, H. Bunke, and S. Venkatesh. Video indexing and similarity retrieval by largest common subgraph detection using decision trees. *Pattern Recognition*, 34:1075–1091, 2001.

- [34] W. H. Suters, F. N. Abu-Khizam, Y. Zhang, C. T. Symons, N. F. Samatova, and M. A. Langston. A new approach and faster exact methods for the maximum common subgraph problem. In *Computing and combinatorics*, volume 3595 of *Lecture Notes in Computer Science*, pages 717–727. Springer-Verlag, 2005.
- [35] J. L. Szwarcfiter. *Grafos e Algoritmos Computacionais*. Campus, Rio de Janeiro, 1984.
- [36] Y. Wang and C. Maple. A novel efficient algorithm for determining maximum common subgraphs. In *International Conference on Information Visualisation (IV'05)*, pages 657–663. 2005.
- [37] P. Willet. Matching of chemical and biological structures using subgraph and maximal common subgraph isomorphism algorithms. *IMA Vol. Math. Appl.*, 108:11–38, 1999.
- [38] L. A. Wolsey. *Integer Programming*. John Wiley and Sons, 1998.
- [39] A. K. C. Wong and F. A. Akinniyi. An algorithm for the largest common subgraph isomorphism using the implicit net. In *Proc. Int. Conf. Systems, Man and Cybernetics*, volume 1, pages 197–201. Inst. Electrical & Electronics Engineers, 1983.
- [40] P. R. J. Östergård. A fast algorithm for the maximum clique problem. *Discrete Appl. Math.*, 120:197–207, 2002.