LI Simpósio Brasileiro de Pesquisa Operacional
Limeira–SP, 02 a 06 de setembro de 2019

Algoritmos de Planos de Corte para o Problema Livre de Garra

Felipe Crispim Fragoso
Centro de Informática / Universidade Federal da Paraíba
João Pessoa – PB – Brasil
felipefragoso@ppgi.ci.ufpb.br

Gilberto Farias de Sousa Filho
Departamento de Computação Científica / Centro de Informática / Universidade Federal da Paraíba
João Pessoa – PB – Brasil
gilberto@ci.ufpb.br

Fábio Protti
Departamento de Ciência da Computação / Instituto de Computação / Universidade Federal Fluminense
Niteroi – RJ – Brasil
fabio@ic.uff.br

RESUMO

O grafo bipartido completo $K_{1,3}$ é uma árvore denominada *garra*. Um grafo G é considerado *livre de garra* se não houver nenhum subgrafo induzido em G isomorfo ao grafo $K_{1,3}$. O *Problema Livre de Garra* (PLG) é NP-Completo e consiste em encontrar o conjunto mínimo $S \subset V(G)$, tal que, $G - S$ é livre de garra. Este trabalho apresenta um estudo poliédrico do politopo associado ao PLG, explicitando sua dimensionalidade e propondo duas famílias de facetas: inequações garra e estrela. Além disso, dois algoritmos de *Planos de Corte* com heurísticas de separação são construídos para teste do desempenho das facetas apresentadas em instâncias de grafos aleatórios. Os resultados mostram que o modelo que utiliza a heurística de separação das inequações estrela obtém melhor resultado em quase todos os casos.

PALAVRAS CHAVE. Problema Livre de Garra. planos de corte. combinatória poliédrica.
Área Principal: PM - Programação Matemática

ABSTRACT

The complete bipartite graph $K_{1,3}$ is a tree know as *claw*. A graph is said to be *claw-free* if it contains no induced subgraph isomorphic to the complete bipartite graph $K_{1,3}$. Given a graph G, the NP-hard *Graph Declawing Problem* (GDP) consists of finding a minimum set $S \subseteq V(G)$ such that $G - S$ is claw-free. This work develops a polyhedral study of the GDP polytope, expliciting its full dimensionality, proposing and testing two families of facets: claw inequalities and star inequalities. In total, two Branch-and-Cut algorithms with separation heuristics have been developed to test the computational benefits of each proposed family on random graph instances. Our results show that the model that uses a separation heuristics proposed for star inequalities achieves better results in almost all cases.

KEYWORDS. Graph Declawing problem. Cutting planes. Polyhedral combinatorics.
Main Area: MP - Mathematical Programming

https://proceedings.science/p/106927?lang=pt-br

1. Introdução

Um grafo G é considerado *livre de garra* quando não há nenhum subgrafo induzido que seja isomorfo ao grafo bipartido completo $K_{1,3}$, também chamado de *garra*. O *Problema Livre de Garra* (PLG) consiste em encontrar o menor conjunto $S \subset V(G)$, tal que, $G - S$ é um grafo livre de garra. Lewis e Yannakakis [1980] apresentam uma prova genérica, mostrando que qualquer problema de remoção de vértices de um grafo direcionado ou não direcionado que tenha por objetivo produzir um grafo com uma determinada propriedade Π, onde Π é *não trivial* e *hereditária*, é um problema NP-Difícil. Π é não trivial se existem infinitos grafos que possuam a propriedade Π e infinitos grafos que não possuam a propriedade Π. Π é hereditária se, para qualquer grafo H que satisfaz Π, todo subgrafo induzido por vértice de H também satisfaz Π. Como estamos considerando a propriedade *livre de garra*, que é uma propriedade não trivial e hereditária, temos que o PLG é um problema NP-Difícil.

O PLG é uma das etapas necessárias para a resolução do problema de remoção de vértices para conversão de um grafo G em um *grafo de intervalos unitários* H, isso é alcançado ao produzir um grafo H que não tenha subgrafo induzido isomorfo aos seguintes grafos: garra, *nets*, *tents* e ciclos C_k ($k \geq 4$) (Vide Figura 1). Esse problema é estudado por van Bevern et al. [2010].

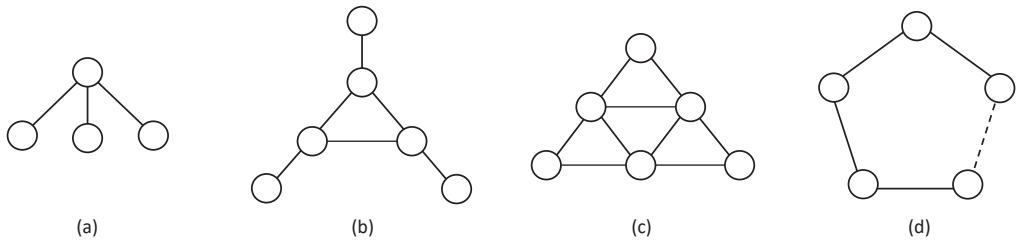


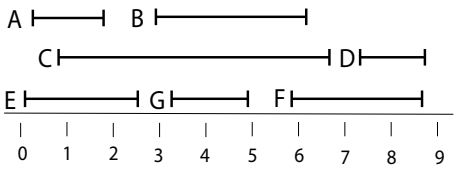
Figura 1: Subgrafos proibidos em um grafo de intervalos unitários: (a) garra, (b) net, (c) tent, (d) $C_k, k \geq 4$.

O PLG também está relacionado ao problema de conversão de um *grafo de intervalos* em um grafo de intervalos unitários. Um grafo de intervalos é construído através das interseções de uma família de intervalos. De maneira formal, G é um grafo de intervalos se cada um dos seus vértices podem ser associados com um intervalo na reta dos números reais, tal que, $uv \in E(G)$ se e somente se $I_u \cap I_v \neq \emptyset$, onde I_u e I_v são intervalos associados a u e v. Um grafo de intervalos unitários é um grafo onde os intervalos possuem todos tamanho 1 (ou, de forma equivalente, o mesmo tamanho). A Figura 2 mostra o processo de conversão de um grafo de intervalos em grafo de intervalos unitário.

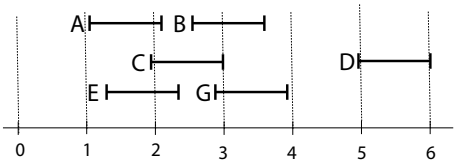
Grafos de intervalos unitários coincidem com a classe de *grafos de intervalos perfeitos* (grafos onde nenhum intervalo contém outro perfeitamente) e *grafos de indiferença* (definidos abaixo). Veja Bogart e West [1999] para uma prova que G é um grafo de intervalos unitários, se e somente se, G é um grafo de intervalos perfeitos, e Roberts [1979] para um prova de que G é um grafo de intervalos unitários, se e somente se, G é um grafo de indiferença.

Grafos de indiferença são definidos no contexto de ciências sociais, discutidos no trabalho de Roberts [1979]. Um grafo G é um grafo de indiferença se existe uma função $f : V(G) \to \mathbb{R}$, tal que,

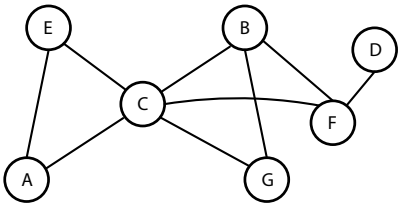
$$(u, v) \in E(G) \iff |f(u) - f(v)| \leq \epsilon,$$



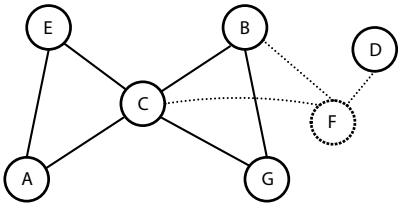
(a) Intervalos na reta dos reais.



(b) Intervalos unitários na reta dos reais.



(c) Grafo de intervalos G.



(d) Grafo de intervalos unitários H, obtido ao remover F de G.

Figura 2: Conversão do grafo de intervalos G (c) em um grafo de intervalos unitários (d) ao remover-se F.

onde ϵ é um número positivo que mede a *proximidade* entre u e v. Informalmente, isso significa que u e v são itens indistiguíveis se, de acordo com f, existe uma diferença menor ou igual a ϵ entre seus valores $f(u)$ e $f(v)$. Como esses valores de f são, em geral, obtidos de forma empírica, existe a possibilidade de itens que são próximos serem considerados como distinguíveis. Portanto, surge o interesse de remover a menor quantidade de items que criem um grafo de indiferença. Esse problema é o problema de conversão de um grafo G em um grafo de intervalos unitários H. Fishburn [1985] apresenta o problema de conversão de uma *ordem intervalar* em uma *ordem de indiferença* através da remoção de elementos; em termos de grafos, esse problema é equivalente ao processo ilustrado na Figura 2.

As propriedades de grafos livres de garra são profundamente investigadas pela comunidade científica (Veja o survey de Faudree et al. [1997]). Por exemplo, Minty [1980] apresenta um algoritmo polinomial para encontrar o conjunto independente de peso máximo em grafos livre de garra. Hsu e Nemhauser [1982] apresentam um algoritmo polinomial para o problema de Cobertura Minima de Cliques em grafos perfeitos livres de garras. Não foi encontrado nenhum trabalho que apresente estudo poliédrico do PLG ou que tente resolvê-lo via programação inteira. Portanto, os autores acreditam que os estudos desenvolvidos no trabalho são justificáveis.

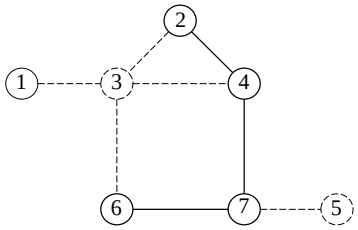
O restante do trabalho é dividido em 5 seções. A seção 2 define o politopo do PLG e apresenta as inequações definidoras de facetas. A seção 3 detalha a heurística de separação dos subgrafos de suporte necessários para a geração das facetas. A seção 4 mostra os resultados computacionais em instâncias de grafos aleatórios. A seção 5 contém as conclusões obtidas.

2. Politopo Livre de Garra

Seja G um grafo com $|V(G)| = \{1, \ldots, n\}$. Seja P_n a envoltória convexa dos vetores de incidência χ^H que representam os subgrafos livres de garra H de G, isto é:

$$P_n = conv\{\chi^H \in \{0,1\}^n \mid \ H \text{ é subgrafo livre de garra de } G\}.$$

P_n é definido como o *politopo livre de garra* de G. A Figura 3 mostra um subgrafo livre de garra H induzido por $\{1, 2, 4, 6, 7\}$ e seu vetor de incidência.



(a) Subgrafo livre de garra H

$$\chi^H = \begin{vmatrix} 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \end{vmatrix}$$

(b) Vetor de incidência $\chi^H \in \{0,1\}^7$

Figura 3: (a) Um subgrafo livre de garra $H = G[\{1, 2, 4, 6, 7\}]$ e (b) vetor incidência associado a H.

O PLG pode ser descrito como um problema de otimização que procura o vetor de incidência $x^T = (x_1, x_2, \ldots, x_n)$ com menor número de zeros (vértices removidos):

$$\min \sum_{i=1}^{n} (1 - x_i)$$
$$s.t. \quad x \in P_n$$

A fim de aplicar técnicas de Programação Linear Intera (PLI) para o PLG, é necessário descrever P_n, que é um politopo n-dimensional e contém o vetor nulo e n vetores unitários, somando $n + 1$ vetores afim independentes. Isso significa que P_n tem dimensão cheia. Ter dimensão cheia implica que toda faceta é unicamente definida através de uma inequação multiplicada por alguma escalar α. Este fato será usado durante a prova das facetas apresentadas. Outro resultado útil é que as facetas de um politopo n-dimensional possuem n vetores afim independentes.

2.1. Inequações triviais e garra

Dado que P_n está contido em um hipercubo unitário, as *inequações triviais*

$$0 \leq x_v \leq 1 \ \ \forall v \in V \tag{1}$$

são claramente válidas. Além disso, toda garra $abcd$ de G, que possui c como vértice central de grau 3, ilustrado na Figura 4, é proibida pela *inequação garra*,

$$(x_a + x_b + x_d) + x_c \leq 3, \tag{2}$$

que é satisfeita por todo vetor em P_n, e portanto, válida para P_n.

Teorema 2.1. *Para todo politopo livre de garra P_n, podemos afirmar que:*

1. *Toda restrição de não negatividade $x_v \geq 0$ define uma faceta de P_n;*
2. *Toda inequação de limite superior $x_v \leq 1$ define uma faceta de P_n;*
3. *Toda inequação garra $(x_a + x_b + x_d) + x_c \leq 3$ define uma faceta de P_n.*

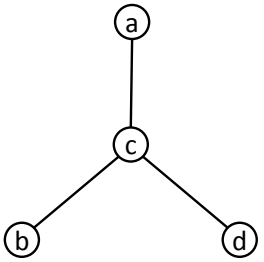


Figura 4: Grafo garra $abcd$.

Demonstração. (1) Seja x_v a v-ésima coordenada de x. Então, $x_v = 0$ é satisfeita pelo vetor nulo e todos os vetores unitários $x_u = 1$ com $u \neq v$. Estes $|V|$ vetores são pertencentes a P_n e são afim independentes.

(2) $x_v = 1$ é satisfeito pelo vetor unitário $x_v = 1$ e os vetores com $x_v = 1$ e $x_u = 1$, $\forall u \neq v$. Estes $|V|$ vetores são pertencentes a P_n e são afim independentes.

(3) Seja $abcd$ uma garra em G. Seja também $a^T x \leq 3$ a inequação (2), e seja $b^T x \leq b_0$ uma faceta definida para P_n tal que $F_a = \{x \in P_n \mid a^T x = 3\} \subseteq F_b = \{x \in P_n \mid b^T x = b_0\}$. Claramente, $F_a \neq \emptyset$ e $F_a \neq P_n$; assim, se for possível provar que $b = \alpha a$ para algum $\alpha \in \mathbb{R}$, concluiria-se que, a inequação (2) define uma faceta de P_n.

Têm-se que $\chi^{\{a,b,c\}}$, $\chi^{\{a,c,d\}}$, $\chi^{\{b,c,d\}}$, $\chi^{\{a,b,d\}} \in F_a \subseteq F_b$, e portanto, $0 = b_0 - b_0 = b^T\chi^{\{a,b,c\}} - b^T\chi^{\{b,c,d\}} = b_a - b_d$, $0 = b_0 - b_0 = b^T\chi^{\{a,b,c\}} - b^T\chi^{\{a,b,d\}} = b_c - b_d$ e $0 = b_0 - b_0 = b^T\chi^{\{a,b,c\}} - b^T\chi^{\{a,c,d\}} = b_b - b_d$; então $b_a = b_b = b_c = b_d = \alpha$.

Para qualquer $v \notin \{a, b, c, d\}$, se a vizinhança de v satisfaz $|N(v) \cap \{a, b, d\}| \leq 2$, então $\chi^{\{a,b,d,v\}} \in F_a \subseteq F_b$, e portanto, $0 = b_0 - b_0 = b^T\chi^{\{a,b,d,v\}} - b^T\chi^{\{a,b,d\}} = b_v$.

Finalmente, para um vértice $v \notin \{a, b, c, d\}$, tal que, $|N(v) \cap \{a, b, d\}| = 3$, $\chi^{\{a,b,c,v\}} \in F_a \subseteq F_b$. Então, $0 = u_0 - u_0 = u^T\chi^{\{a,b,c,v\}} - u^T\chi^{\{a,b,c\}} = u_v$.

Logo, para qualquer $v \in (V - \{a, b, c, d\})$ vale que $b_v = 0$. Essas observações implicam que a inequação (2) define uma faceta de P_n.

□

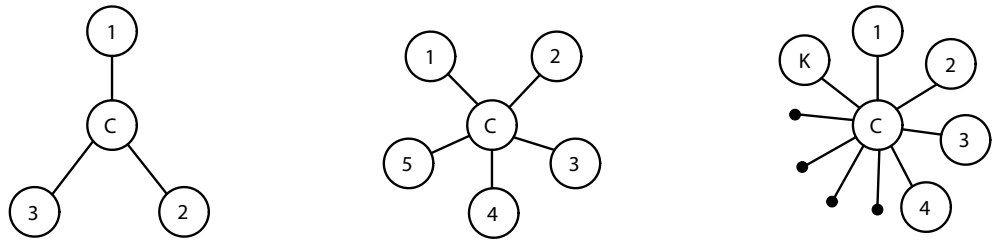
O PLG, então, pode ser modelado como o seguinte problema PLI:

$$
\begin{aligned}
(\mathcal{F}_g)\ \min \sum_{v \in V(G)} (1 - x_v) & \\
\text{s.a.} & \\
x_a + x_b + x_c + x_d \leq 3, & \qquad \forall abcd \subset G \\
0 \leq x_v \leq 1, & \qquad \forall v \in V, \\
x \text{ é inteiro.} &
\end{aligned}
$$

2.2. Inequação estrela

Definição 2.1. *(Grafo estrela) O grafo estrela S_k é um grafo bipartido completo $K_{1,k}$ de ordem $k+1$, para $k \geq 3$. S_k possui um vértice central c, cuja vizinhança $N(c)$ é um conjunto independente I_k.*

A Figura 5 ilustra possíveis topologias para o grafo estrela.



(a) S_3 (b) S_5 (c) S_k

Figura 5: Topologias S_3, S_5, and S_k.

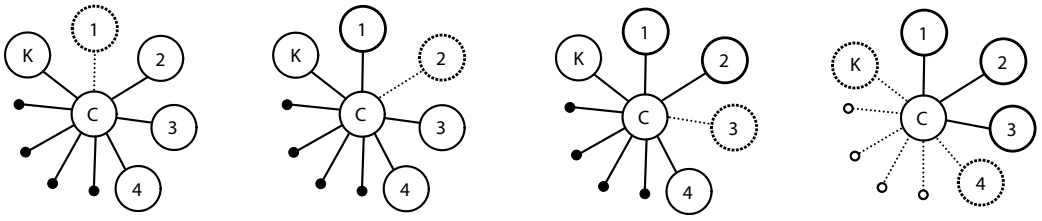
Teorema 2.2. *Seja $S_k \subset G$ um grafo estrela, a correspondente inequação*

$$\sum_{v \in I_k} x_v + (k-2)x_c \leq k, \tag{3}$$

é:

1. *Válida para P_n.*
2. *Define uma faceta de P_n.*

Demonstração. (1) O caso $k = 3$ corresponde a inequação garra. Para $k \geq 4$, a prova será por indução. Seja $S_k^v = S_k - v$ e $S_3' = S_k[\{i, j, \ell, c\}]$, onde i, j, ℓ são vértices distintos de I_k. Note que S_k^i, S_k^j, S_k^ℓ, e S_3' são subgrafos de S_k. A Figura 6 mostra exemplos, onde $i = 1, j = 2, \ell = 3$.



(a) S_k^i (b) S_k^j (c) S_k^ℓ (d) S_3'

Figura 6: Subgrafos S_k^i, S_k^j, S_k^ℓ e S_3' de S_k.

Pela hipótese de indução, as inequações associadas a S_k^i, S_k^j, S_k^ℓ e S_3' são válidas para P_n, e a soma delas gera a seguinte inequação:

$$\sum_{v \in I_k} 3x_v + (3k-8)x_c \leq 3k.$$

Adicionando $2x_c \leq 2$ obtemos

$$\sum_{v \in I_k} 3x_v + (3k-6)x_c \leq 3k + 2,$$

e, portanto,

$$\sum_{v \in I_k} x_v + (k-2)x_c \leq k + \left\lfloor \frac{2}{3} \right\rfloor = k$$

é válida para P_n.

(2) Seja S_k uma estrela em G. Seja também $a^T x \leq k$ a inequação (3), e seja $b^T x \leq b_0$ uma faceta definida para P_n tal que $F_a = \{x \in P_n \mid a^T x = k\} \subseteq F_b = \{x \in P_n \mid b^T x = b_0\}$. Claramente, $F_a \neq \emptyset$ e $F_a \neq P_n$; assim, se for possível provar que $b = \alpha a$ para algum $\alpha \in \mathbb{R}$, concluiria-se que, a inequação (2) define uma faceta de P_n.

Seja $L_{i,j}$ um subgrafo livre de garra de S_k, tal que, $L_{i,j} = S_k[\{i, j, c\}]$, onde $\{i, j\} \subset I_k$ e c é o centro de S_k. A Figura 7 mostra dois exemplos de subgrafos possíveis.



(a) $L_{1,2}$ (b) $L_{1,3}$

Figura 7: Subgrafos $L_{1,2}$ e $L_{1,3}$ de S_k.

Têm-se que χ^{I_k}, $\chi^{L_{i,j}}$ e $\chi^{L_{j,k}}$ estão em $F_a \subseteq F_b$. Então, as seguintes equações são válidas: $0 = b_0 - b_0 = b^T\chi^{L_{i,j}} - b^T\chi^{L_{j,k}} = b_i - b_k$. Portanto, $b_i = b_k = \alpha$. Aplicando o mesmo processo para todo $v \in I_k$ temos $b_v = \alpha$. Também, $0 = b_0 - b_0 = b^T\chi^{I_k} - b^T\chi^{L_{i,j}} = (\sum_{v \in I_k \setminus \{i,j\}} b_v) - b_c$, implicando em $b_c = \alpha(k-2)$.

Para um vértice $v \notin I_k \cup \{c\}$, se a sua vizinhança satisfaz $|N(v) \cap I_k| < 3$, então $\chi^{I_k \cup \{v\}} \in F_a \subseteq F_b$. Portanto, a seguinte equação é válida: $0 = b_0 - b_0 = b^T\chi^{I_k \cup \{v\}} - b^T\chi^{I_k} = b_v$.

Finalmente, para um vértice $v \notin I_k \cup \{c\}$, tal que, $|N(v) \cap I_k| \geq 3$, $\chi^{V(L_{i,j}) \cup \{v\}}$, onde $\{i, j\} \subset N(v) \cap I_k$, está em $F_a \subseteq F_b$. Então, $0 = b_0 - b_0 = b^T\chi^{V(L_{i,j}) \cup \{v\}} - b^T\chi^{L_{i,j}} = b_v$.

□

Como a inequação (2) é um caso particular da inequação (3) quando $k = 3$, pode-se reescrever ($\mathcal{F}_g$) da seguinte forma:

$$(\mathcal{F}_{S_k}) \min \sum_{v \in V(G)} (1 - x_v)$$

s.a.

$$\sum_{v \in I_k} x_v + (k-2)x_c \leq k, \qquad \forall S_k \subset G$$

$$0 \leq x_v \leq 1, \qquad \forall v \in V,$$

$$x \text{ é inteiro.}$$

3. Planos de Corte para $\mathcal{F}_g$ e $\mathcal{F}_{S_k}$

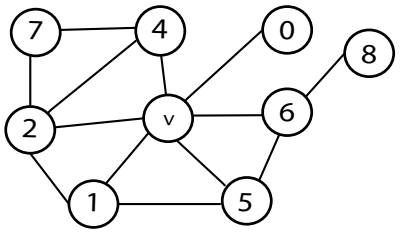
3.1. Extraíndo subgrafos garra

Subgrafos garras são obtidos através de um laço $O(n^4)$. Basta verificar se existe um vizinho em comum para cada trio não adjacente de vértices. Cada garra é adicionada a uma coleção $\mathcal{S}$ de subgrafos de suporte.

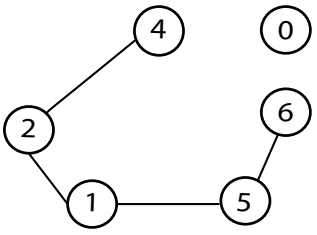
https://proceedings.science/p/106927?lang=pt-br

3.2. Extraíndo subgrafos estrela

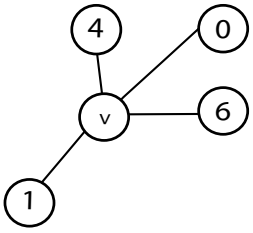
Para reduzir o número de subgrafos estrelas de um grafo, consideramos apenas estrelas formadas por um vértice central v e um conjunto independente maximal em $G[N(v)]$ ('estrelas maximais'). Por exemplo, a Figura 8 mostra duas estrelas S_3 e S_4 centradas em v que são formadas por conjuntos independentes maximais em $G[N(v)]$.



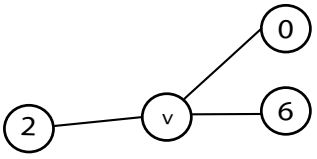
(a) Grafo G



(b) $G[N(v)]$



(c) Subgrafo estrela S_4



(d) Subgrafo estrela S_3

Figura 8: Extração de estrelas maximais centradas em v (c) e (d) a partir de (a) usando a vizinhança de v (b).

Como conjuntos independentes maximais são cliques maximais no grafo complementar, o algoritmo para extração de cliques maximais $hybrid(.)$ proposto por Eppstein e Strash [2011] pode ser utilizado para encontrar as estrelas maximais S_k centradas em um vértice v através do complemento de $G[N(v)]$.

O algoritmo 1 resume a extração das estrelas maximais S_k de um grafo G. A linha 1 inicializa uma coleção $\mathcal{S}$ de grafos de suporte. A linha 2 percorre o conjunto $V(G)$, fixando um possível centro de estrela a cada iteração. A linha 3 encontra o complemento de $G[N(v)]$. Na linha 4, são extraídos todos os conjuntos independentes maximais em $\overline{G[N(v)]}$ usando o algoritmo $hybrid(.)$. Linhas 5–8 adicionam a $\mathcal{S}$ apenas estrelas maximais com conjunto independente maior igual a 3.

3.3. Heurística de separação para ($\mathcal{F}_g$) e ($\mathcal{F}_{S_k}$)

Com o conjunto de grafos de suporte em mãos, o algoritmo de Planos de Corte resolve a relaxação linear do problema, para então, procurar alguma inequação violada pela solução ótima x^* encontrada na relaxação. A verificação de uma possível violação se dá através do cálculo $\lambda = \sum_{v \in I_k} x^*_v + (k-2)x^*_c - k$ para cada subgrafo de suporte (garra ou estrela). Se $\lambda > 0,0001$ a inequação é considerada violada e é adicionada em uma fila de prioridades de inequações violadas, para enfim, extrair as 100 melhores inequações (maiores valores para λ) que serão adicionadas na relaxação atual.

4. Resultados Computacionais

Todos os métodos estudados neste trabalho foram desenvolvidos em linguagem C++, utilizando o compilador g++ versão 4.6.3 (opção de compilação -O3), com o auxilio do resolvedor

Algoritmo 1: ALGORITMO PARA EXTRAÇÃO DE ESTRELAS MAXIMAIS

```
S = ∅
for v ∈ V(G) do
    G' = complement(G_n[N(v)])
    S_temp = hybrid(G')
    for I_k ∈ S_temp do
        if k ≥ 3 then
            S_k = G[I_k ∪ {v}]
            S = S ∪ S_k
        end
    end
end
return S
```

matemático CPLEX versão 12. Os experimentos computacionais foram executados em um computador com processador *Intel Core* I7 (64 bits), com CPU contendo 4 *cores* de 3.4 GHz, além de 16 GB de RAM e executando o sistema operacional Linux Ubuntu 14.04. Todos os testes foram realizados com 1 *thread* e limite máximo da árvore de Planos de Corte de 10 GB.

Para os experimentos realizados, foram utilizados grafos aleatórios gerados por Bastos et al. [2016]. Foram selecionadas as instâncias com $|V| = \{50, 100\}$.

4.1. Comparação entre $\mathcal{F}_g$ e $\mathcal{F}_{S_k}$

A Tabela 1 apresenta os resultados com $n = 50$. Ambos os modelos conseguiram resolver as instâncias com otimalidade. A tabela mostra o tempo de execução total do algoritmo, o tempo de execução do algoritmo de separação entre parênteses, o número de nós resolvidos e o número de cortes adicionados durante a execução do algoritmo. Para este conjunto de instâncias, $\mathcal{F}_{S_k}$ obtém um tempo de execução melhor em todas as instâncias.

O modelo $\mathcal{F}_{S_k}$ necessita de menos cortes e nós expandidos em todas as instâncias. Isso é uma boa evidência que inequações estrela dominam as inequações garra. Por exemplo, a instância 105_6_50 precisou de 8 vezes menos tempo e 4 vezes menos cortes para ser resolvida. Isso explica a diminuição do tempo necessário para encontrar a solução ótima.

A tabela 2 mostra o resultado para grafos aleatórios com $n = 100$. A tabela mostra o *lower bound*, *upper bound*, o tempo de execução, número de nós e número de cortes adicionados. O modelo $\mathcal{F}_{S_k}$ continua com melhores resultados.

5. Considerações Finais e Proposta de Trabalhos Futuros

Este trabalho tratou do Problema Livre de Garra (PLG). O PLG procura encontrar o conjunto mínimo $S \subset V$ tal que $G - S$ seja livre de garra. A remoção de garras de um grafo está relacionado ao problema de transformar grafos em grafos de indiferença (ou, de forma equivalente, grafos unitários ou perfeitos), grafos esses muito estudados pela literatura de otimização por permitir que problemas difíceis sejam resolvidos de forma mais eficiente.

Um estudo poliédrico do PLG foi proposto, identificando dois modelos PLI: o $\mathcal{F}_g$ que possui as facetas definidas pelas inequações garra e o $\mathcal{F}_{S_k}$ que possui novas facetas definidas pelas as inequações estrela como uma generalização das inequações garra. Como o número de inequações estrela é exponencial, foi proposto ainda um algoritmo baseado em planos de corte para resolver a relaxação linear deste problema de forma mais eficiente.

https://proceedings.science/p/106927?lang=pt-br

Tabela 1: Resultados computacionais para grafos aleatórios com $n = 50$.

			$\mathcal{F}_g$			$\mathcal{F}_{S_k}$		
Instâncias	Densidade	OPT	**T (s)**	**#Nós**	**#Cortes**	**T (s)**	**#Nós**	**#Cortes**
105_1_50	0,05	16,00	0,09(0,01)	110	1085	0,02(0,00)	2	337
105_2_50	0,10	19,00	0,18(0,01)	315	747	0,03(0,00)	127	115
105_3_50	0,15	23,00	2,25(0,07)	3415	1650	0,15(0,01)	776	208
105_4_50	0,22	28,00	32,92(0,87)	22593	4351	1,58(0,08)	4073	612
105_5_50	0,24	29,00	116,06(3,40)	61758	5579	3,40(0,17)	6177	913
105_6_50	0,32	32,00	363,81(9,47)	111681	8675	15,19(0,57)	13713	2016
105_7_50	0,32	32,00	278,16(7,77)	86531	9002	21,69(0,93)	20387	2057
105_8_50	0,42	35,00	643,82(17,15)	134288	13036	47,43(1,24)	22035	3694
105_9_50	0,43	35,00	727,60(20,67)	146154	13769	68,22(1,90)	28127	3990
105_10_50	0,47	35,00	666,90(22,94)	136030	13218	57,96(1,55)	21185	4588
106_1_50	0,11	12,00	0,02(0,00)	3	217	0,01(0,00)	4	103
106_2_50	0,16	21,00	0,70(0,02)	972	1246	0,09(0,00)	236	273
106_3_50	0,20	25,00	7,21(0,22)	7254	2810	0,40(0,02)	897	576
106_4_50	0,25	28,00	46,02(1,49)	29146	4913	2,75(0,11)	3853	1142
106_5_50	0,30	31,00	161,31(4,56)	59852	7689	11,08(0,49)	11397	1818
106_6_50	0,32	32,00	209,15(5,31)	65449	8636	9,82(0,35)	8575	2022
106_7_50	0,40	34,00	536,74(15,30)	124642	12034	18,79(0,54)	9289	3382
106_8_50	0,42	35,00	870,03(26,62)	191671	12187	123,19(2,99)	53993	3852
106_9_50	0,44	35,00	712,37(20,09)	147306	13218	67,79(1,83)	28457	4164
106_10_50	0,50	34,00	371,87(13,53)	79052	13039	25,60(0,84)	9194	4797

Tabela 2: Resultados computacionais para grafos aleatórios com $n = 100$.

				$\mathcal{F}_g$					$\mathcal{F}_{S_k}$		
Instâncias	Densidade	**LB**	**UB**	**T (s)**	**#Nós**	**#Cortes**	**LB**	**UB**	**T (s)**	**#Nós**	**#Cortes**
109_1_100	0,05	45,00	63,00	680,63(211,42)	27062	54678	60,00	60,00	3,54(0.48)	256	6885
109_2_100	0,11	39,51	59,00	596,37(21,99)	100594	11858	55,00	55,00	3296,50(99.59)	3387399	895
109_3_100	0,15	40,75	67,00	554,64(37,07)	46700	25418	56,19	64,00	971,26(49.50)	359124	3470
109_4_100	0,20	40,60	76,00	518,45(51,97)	28998	42083	58,57	72,00	1057,89(55.47)	86873	12149
109_5_100	0,25	41,25	79,00	532,63(70,92)	23391	55790	59,79	75,00	1325,18(56.88)	43298	25334
109_6_100	0,29	41,70	81,00	631,58(87,40)	20251	65533	61,11	78,00	1531,49(59.12)	30837	37710
109_7_100	0,36	42,67	84,00	658,79(122,85)	18571	76663	60,19	82,00	1606,44(69.09)	22971	52368
109_8_100	0,39	43,04	85,00	697,71(131,90)	17215	79505	61,02	83,00	1644,75(70.48)	21432	56833
109_9_100	0,45	44,12	86,00	678,57(159,58)	17296	85094	60,28	85,00	1596,33(71.80)	19387	65034
109_10_100	0,50	44,07	85,00	705,25(158,11)	16198	88850	61,22	85,00	1614,32(70.32)	19117	66042
110_1_100	0,13	43,10	49,00	1647,33(34,45)	253453	8783	48,00	48,00	668,77(15.82)	352227	1742
110_2_100	0,17	41,28	63,00	595,50(40,33)	53454	22135	54,42	59,00	1893,47(62.09)	313549	5195
110_3_100	0,21	41,04	72,00	663,39(49,37)	30194	40564	58,08	69,00	1281,57(57.43)	76838	14558
110_4_100	0,26	41,75	79,00	553,02(66,43)	23599	54720	57,63	77,00	1288,43(56.99)	43005	25475
110_5_100	0,30	42,00	81,00	655,51(85,08)	19422	69145	60,64	79,00	1413,11(55.85)	29670	39072
110_6_100	0,34	42,26	83,00	635,52(111,21)	18460	75576	60,10	82,00	1551,39(65.61)	25240	47224
110_7_100	0,38	43,17	86,00	692,43(124,12)	17623	78625	60,03	83,00	1608,75(69.97)	21903	55566
110_8_100	0,43	43,67	85,00	672,47(142,52)	16945	83828	61,10	83,00	1637,98(72.85)	20265	61687
110_9_100	0,46	43,69	85,00	688,50(160,54)	17271	85066	60,15	85,00	1610,98(72.27)	19061	65935
110_10_100	0,50	44,13	85,00	704,46(155,52)	16141	91062	63,01	84,00	1770,65(70.14)	18597	68311

Os resultados computacionais demonstram o impacto das novas facetas propostas em $\mathcal{F}_{S_k}$, onde a resolução de quase todas as instâncias se mostrou mais eficiente no tempo computacional, além de melhorar os valores de *lower bound* e *upper bound* das soluções não resolvidas.

Como propostas de trabalhos futuros tem-se: encontrar novas facetas para o PLG e propor um modelo baseado em geração de colunas.

Referências

Bastos, L., Ochi, L. S., Protti, F., Subramanian, A., Martins, I. C., e Pinheiro, R. G. S. (2016). Efficient algorithms for cluster editing. *Journal of Combinatorial Optimization*, 31(1):347–371. ISSN 1573-2886.

Bogart, K. P. e West, D. B. (1999). A short proof that 'proper = unit'. *Discrete Mathematics*, 201 (1):21 – 23. ISSN 0012-365X.

Eppstein, D. e Strash, D. (2011). *Listing all maximal cliques in large sparse real-world graphs*, p. 364–375. Springer Berlin Heidelberg, Berlin, Heidelberg. ISBN 978-3-642-20662-7.

Faudree, R., Flandrin, E., e Ryjáček, Z. (1997). Claw-free graphs - A survey. *Discrete Mathematics*, 164(1):87 – 147. ISSN 0012-365X.

Fishburn, P. C. (1985). Interval graphs and interval orders. *Discrete Applied Mathematics*, 55(2): 135–149.

Hsu, W.-L. e Nemhauser, G. L. (1982). A polynomial algorithm for the minimum weighted clique cover problem on claw-free perfect graphs. *Discrete Mathematics*, 38(1):65 – 71. ISSN 0012-365X.

Lewis, J. M. e Yannakakis, M. (1980). The node-deletion problem for hereditary properties is NP-complete. *Journal of Computer and System Sciences*, 20(2):219 – 230. ISSN 0022-0000.

Minty, G. J. (1980). On maximal independent sets of vertices in claw-free graphs. *Journal of Combinatorial Theory, Series B*, 28(3):284 – 304. ISSN 0095-8956.

Roberts, F. S. (1979). Indifference and seriation. *Annals of the New York Academy of Sciences*, 328 (1):173–182. ISSN 1749-6632.

van Bevern, R., Komusiewicz, C., Moser, H., e Niedermeier, R. (2010). Measuring indifference: Unit interval vertex deletion. In Thilikos, D. M., editor, *Graph Theoretic Concepts in Computer Science*, p. 232–243, Berlin, Heidelberg. Springer Berlin Heidelberg. ISBN 978-3-642-16926-7.