

O conteúdo do presente relatório é de única responsabilidade do(s) autor(es).  
(The contents of this report are the sole responsibility of the author(s).)

**Uma Abordagem de Programação Inteira para  
o Problema da Triangulação de Custo Mínimo**

*Aminadab Pereira Nunes*

*Cid Carvalho de Souza*

**Relatório Técnico IC-96-19**

Dezembro de 1996

# Uma Abordagem de Programação Inteira para o Problema da Triangulação de Custo Mínimo\*

Aminadab Pereira Nunes      Cid Carvalho de Souza<sup>†</sup>

*Instituto de Computação  
Universidade Estadual de Campinas — UNICAMP  
Caixa Postal 6176 – CEP: 13081-970 – Campinas, SP – Brasil  
e-mail: [aminadab,cid]@dcc.unicamp.br*

## Sumário

Dado um conjunto finito de pontos no plano, uma triangulação planar é definida como um conjunto maximal de segmentos de reta conectando estes pontos, tal que dois destes segmentos não se interceptam, exceto nos extremos. Chamamos de triangulação de custo mínimo a triangulação planar cuja soma total dos comprimentos de seus segmentos de reta é mínimo dentre todas as triangulações planares do conjunto de pontos em questão. Não se conhece algoritmo polinomial que resolva o problema de determinar a triangulação de custo mínimo de um conjunto de pontos no caso geral, contudo, também não está provado tratar-se de um problema *NP-difícil*. Neste artigo estamos interessados na resolução exata deste problema. Nossa abordagem é baseada em técnicas de programação inteira, em particular avaliamos duas formulações distintas para o problema. A primeira formulação é baseada em uma equivalência entre o problema da triangulação de custo mínimo e uma versão restrita do problema do conjunto independente em um grafo. Além das desigualdades obtidas através da observação desta equivalência, mostramos como fortalecer a formulação através de certas propriedades geométricas do problema. Avaliamos ainda uma outra formulação baseada principalmente no trabalho apresentado por Loera *et. al* em [dLHSS96]. Finalmente, reportamos nossos experimentos computacionais.

**Palavras-chaves:** Programação Inteira, Combinatória Poliédrica, Geometria Computacional e Triangulação de Custo Mínimo.

## 1 Introdução

Seja  $P$  um conjunto finito de  $n \geq 3$  pontos no plano, tal que  $P$  não esteja contido em nenhuma reta. Uma *triangulação planar* de  $P$  é um conjunto maximal de segmentos de

---

\*Este trabalho recebeu apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior — CAPES, e é atualmente financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo — FAPESP (Proc. No. 95/8929-9).

<sup>†</sup>Este autor é bolsista do Conselho Nacional de Desenvolvimento Científico e Tecnológico — CNPq (Proc. No. 300833/94-3).

reta com extremos em  $P$ , que não se interceptam, exceto possivelmente nos extremos. Por simplicidade, no restante deste texto diremos que dois segmentos no plano se interceptam apenas quando esta interseção se der um ponto que não seja extremidade dos segmentos considerados. Uma *triangulação de custo mínimo* (TCM) do conjunto  $P$ , é definida como sendo uma triangulação planar cuja soma dos comprimentos de seus segmentos de reta é mínimo dentre todas as triangulações planares de  $P$ . Na figura 1 vemos duas triangulações planares distintas do mesmo conjunto de pontos, sendo que a triangulação da esquerda é uma TCM. A TCM encontra aplicações em análise numérica [Llo77, PS85, LL87]. Além disso, Wang e Aggarwal [WA86] também a utilizam em um algoritmo para reconstrução de superfícies a partir de seus contornos.

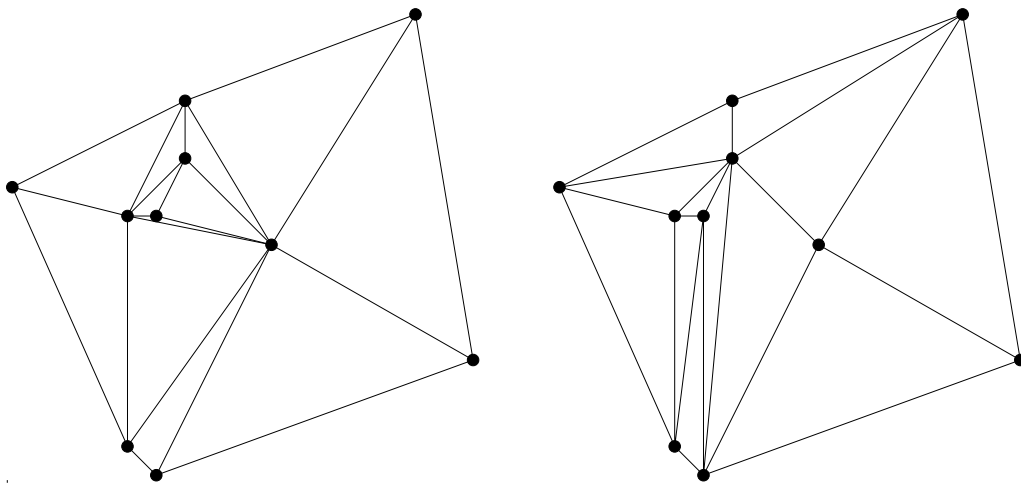


Figura 1: Dois exemplos de triangulações planares.

O problema de determinar a TCM (PTCM) de um conjunto de pontos pertence a classe de problemas cuja complexidade permanece desconhecida. Apesar de não se conhecer algoritmos polinomiais que o resolva não se conseguiu provar tratar-se de um problema *NP-difícil*. Tradicionalmente, este problema tem sido tratado de forma heurística [LL87, DRA94, HP94]. O objetivo destes métodos heurísticos é produzir triangulações planares que se aproximem da TCM. Em geral, as triangulações planares utilizadas como este intuito são as triangulações gulosa e de Delaunay. Contudo, existem trabalhos recentes que apresentam formas de se computar um subconjunto da TCM a partir do qual pode ser possível computar a TCM em tempo polinomial [BKMS96, DM96].

Neste artigo apresentamos uma abordagem de programação inteira (PI) para resolução exata do PTCM. Neste sentido avaliamos duas formulações de PI distintas para o problema. A primeira formulação apresentada, a qual chamamos de *formulação baseada em segmentos*, visa identificar os segmentos de uma TCM utilizando-se de uma equivalência entre o PTCM e uma versão restrita do problema do conjunto independente em um grafo. Além das desigualdades obtidas através da observação desta equivalência, mostramos que é possível fortalecer esta formulação através da inserção de novas desigualdades válidas que expressam certas propriedades geométricas do problema. Mostramos ainda que, otimi-

zar sobre o politopo definido pela formulação em questão é equivalente a otimizar sobre o politopo monótono.

Como pode ser visto na figura 1, a condição de maximalidade impõe que todas regiões limitadas em uma triangulação planar sejam triângulos (daí o nome triangulação planar). Assim, o PTCM também pode ser visto como o problema de determinar quais triângulos são formados em uma TCM (os triângulos fornecem implicitamente os segmentos utilizados). De fato, a segunda formulação apresentada tenta identificar os triângulos de uma TCM, ao invés dos segmentos, e por isso a chamamos de *formulação baseada em triângulos*. Esta última formulação é fundamentada principalmente no trabalho apresentado por Loera *et. al* em [dLHSS96].

Finalmente, também apresentamos neste artigo uma avaliação computacional das formulações propostas, em especial, no caso da formulação baseada em segmentos estes resultados foram obtidos através de uma implementação de um algoritmo *branch-and-cut*.

O restante deste texto está dividido da seguinte forma. Na seção 2 apresentamos uma coletânea dos principais resultados relativos ao PTCM. Nas seções 3 e 4 discutimos, respectivamente, as formulações de PI baseadas em segmentos e em triângulos. Na seção 5 apresentamos uma descrição sucinta dos algoritmos utilizados na resolução do problema utilizando as formulações propostas. Na seção 6 apresentamos alguns resultados preliminares obtidos utilizando estes algoritmos. Na seção 7 são apresentadas as conclusões deste trabalho.

## 2 Abordagens Anteriores

O PTCM é um problema que há muito desafia os pesquisadores de geometria computacional. No sentido de mostrar que trata-se de um problema *NP-difícil*, existem dois resultados principais. Seja  $S(P)$  o conjunto de todos os segmentos de reta com extremos em  $P$ . Em [Llo77] Lloyd mostra que o problema de determinar se existe uma triangulação planar que utilize somente um certo subconjunto de segmentos  $S' \subseteq S(P)$  é *NP-completo*. Recentemente, em [HP94], Heath e Pemmaraju apresentam o resultado mais próximo à prova de que o PTCM é *NP-difícil*. Dados um subconjunto  $S' \subseteq S(P)$ , tal que  $S'$  contém pelo menos uma triangulação de  $P$ , e um número racional positivo  $k$ , os autores mostram que determinar se existe uma triangulação planar em  $S'$  cujo custo não exceda à  $k$  é um problema *NP-difícil*.

Os métodos heurísticos existentes visam construir triangulações planares que se aproximem da TCM. As mais tradicionais aproximações são as triangulações gulosa e de Delaunay. A triangulação gulosa é construída de forma iterativa, onde a cada passo se adiciona à triangulação o menor segmento que não intercepta nenhum dos segmentos previamente adicionados. De forma ingênua a triangulação gulosa pode ser construída em tempo  $O(n^3)$ , entretanto, através de abordagens mais sofisticadas pode ser obtida em tempo  $O(n^2 \log n)$  [PS85]. A triangulação de Delaunay consiste no dual do diagrama de Voronoi, e pode ser computada em  $O(n \log n)$  [PS85]. Nenhuma das duas triangulações conduzem obrigatoriamente a TCM. De fato, em [MZ79] Manacher e Zobrist mostram que as triangulações gulosa e de Delaunay podem ser, respectivamente, pelo menos  $\Omega(n^{1/3})$  e  $\Omega(n)$  maior que a TCM.

Posteriormente, em [Lev87], Levcopoulos mostrou que a triangulação gulosa pode ser pelo menos  $\Omega(\sqrt{n})$  maior que a TCM. Desta forma, tanto a triangulação gulosa quanto a de Delaunay podem fornecer aproximações arbitrariamente ruins.

O algoritmo que oferece a melhor garantia de qualidade para o PTCM, conhecido até o momento, é capaz de produzir uma triangulação que é no máximo um fator de  $O(\log n)$  maior que a TCM, e cujo tempo de computação é  $O(n^2 \log n)$  [HP94]. Ou seja, não existe algoritmo eficiente capaz de produzir uma triangulação cujo custo seja limitado por um fator constante do custo da TCM. Entretanto, apesar das aproximações acima não oferecerem boas garantias de qualidade, algumas destas, a exemplo da triangulação gulosa, produzem bons resultados na prática.

O primeiro resultado importante quanto a resolução exata do PTCM foi apresentado em [Gil79]. Neste trabalho Gilbert mostra que é possível, através de uma abordagem de programação dinâmica, computar a TCM de um polígono simples em  $O(n^3)$ . Este resultado foi depois estendido [HP94] para possibilitar a computação da TCM de um objeto mais genérico chamado célula, onde uma célula de um conjunto de pontos  $P$  consiste em um conjunto de segmentos que formam uma árvore geradora do grafo  $G_P(P, S(P))$  acrescido dos segmentos que limitam a envoltória convexa de  $P$ . Assim, o PTCM pode ser reduzido ao problema de encontrar uma célula que esteja em alguma TCM, pois o problema resultante pode ser resolvido em tempo polinomial.

A possibilidade de computar a TCM de uma célula em tempo polinomial motivou trabalhos no sentido de determinar conjuntos de segmentos  $S' \subseteq S(P)$  que estão em alguma TCM [BKMS96, DM96, Yan95]. Note que se o grafo  $\hat{G}_P(P, S')$  for conexo teremos uma célula, e o problema resultante pode ser resolvido em  $O(n^3)$ . Mesmo que  $\hat{G}_P$  não seja conexo o problema poderá ser resolvido em  $O(n^{k+2})$  [CGT95], onde  $k$  é o número de componentes conexos de  $\hat{G}_P$ . Novamente, ainda não existe um método capaz de produzir um conjunto  $S'$  tal que  $k$  seja limitado por uma constante.

Em artigo recente [dLHSS96] Loera, Hosten, Santos e Sturmfels também propõem uma formulação de PI que pode modelar o PTCM. Esse trabalho não trata especificamente de triangulações planares, e sim de uma generalização deste conceito onde os pontos pertencem a  $\mathbb{R}^d$  com  $d \geq 2$ . De fato, a formulação apresentada na seção 4 é uma versão, um pouco modificada, da apresentada por esses autores.

### 3 A Abordagem de PI Baseada em Segmentos

Como mencionado, o PTCM consiste em determinar um subconjunto maximal  $T_s(P) \subseteq S(P)$ , onde a soma total dos comprimentos dos segmentos em  $T_s(P)$  seja mínima e tal que quaisquer dois de seus segmentos não se interceptem. Assim, podemos ver  $T_s(P)$  como um conjunto independente maximal de elementos de  $S(P)$ . De fato, o PTCM pode ser modelado como uma versão restrita do problema do conjunto independente em um grafo.

#### 3.1 O PTCM e o Problema do Conjunto Independente

Dizemos que  $G_P^{is}(V, E)$  é o *grafo das interseções de segmentos* de  $P$  se: para cada segmento  $\overline{ij} \in S(P)$  existe um vértice  $v_{ij} \in V$  que o representa, e para cada dois segmentos  $\overline{ij}$

e  $\overline{k\ell}$  que se interceptam existe uma aresta  $(v_{ij}, v_{k\ell}) \in E$ . Atribuindo-se a cada vértice  $v_{ij} \in V$  um custo igual ao comprimento do segmento  $\overline{ij}$ , verificamos que o problema de determinar a TCM de um conjunto  $P$  de pontos é equivalente ao problema de determinar um conjunto independente maximal de custo mínimo no grafo das interseções de segmentos de  $P$ . É conhecido que este último problema é *NP-difícil* [HP94].

Com base no que foi descrito acima, o PTCM pode ser formulado da seguinte forma: a cada segmento  $\overline{ij} \in S(P)$ , cujo custo, ou comprimento, será representado por  $c_{ij}$ , associa-se uma variável  $x_{ij} \in \{0, 1\}$ . Seja  $x$  o vetor das variáveis  $x_{ij}$ . O valor de  $x_{ij}$  será 1 se e somente se o segmento  $\overline{ij}$  pertence a triangulação planar representada por  $x$ . Sendo assim, uma formulação do PTCM é dada por:

$$\begin{aligned} \min \quad & \sum_{\overline{ij} \in S(P)} c_{ij} x_{ij} \\ \text{Sujeito a} \quad & x \in \mathcal{X}_s^{CI} \quad (\text{I}) \\ & \sum_{\overline{ij} \in S(P)} x_{ij} = 3(n-1) - h \quad (\text{II}) \end{aligned} \quad (1)$$

onde  $\mathcal{X}_s^{CI}$  representa a família de todos vetores de incidência dos conjuntos independentes de  $G_P^{is}$ , e  $h$  o número de pontos de  $P$  sobre a fronteira de sua envoltória convexa. A restrição (I) diz que  $x$  deve representar um conjunto independente de  $G_P^{is}$ . A restrição (II) faz uso de uma outra propriedade das triangulações planares. Esta propriedade determina o número exato de segmentos em qualquer triangulação planar de um conjunto de pontos. Isto é utilizado para garantir a condição de maximalidade inicialmente imposta.

É fácil verificar que a formulação (1) é válida para o PTCM. Entretanto, resta esclarecer como a restrição (I) pode ser representada através de um conjunto de desigualdades lineares. Isto pode ser feito utilizando-se as duas desigualdades abaixo:

$$\begin{aligned} x_{ij} + x_{k\ell} &\leq 1 \quad \forall (v_{ij}, v_{k\ell}) \in E \quad (\text{III}), \\ x_{ij} &\in \{0, 1\} \quad \forall \overline{ij} \in S(P) \quad (\text{IV}). \end{aligned}$$

As desigualdades da classe (III), conhecidas como *desigualdades de aresta*, dizem que dois vértices vizinhos em  $G_P^{is}$  não podem estar simultaneamente em um conjunto independente. As restrições da classe (IV) determinam o conjunto de valores que as variáveis podem assumir, no caso 0 ou 1. Estas duas classes de restrições são suficientes para garantir que  $x \in \mathcal{X}_s^{CI}$ .

Visando fortalecer a formulação podemos ainda utilizar outras desigualdades válidas para o problema do conjunto independente. De fato, em nossos experimentos computacionais utilizamos dois outros tipos de desigualdades válidas para o politopo dos conjuntos independentes. Trata-se das *desigualdades de clique* e *de odd hole*. Uma desigualdade de clique diz que, dado uma clique no grafo  $G_P^{is}$ , no máximo um vértice desta clique pode estar em um conjunto independente. Uma desigualdade de *odd hole* diz que, dado um ciclo  $C$ , sem cordas, de tamanho ímpar em  $G_P^{is}$ , no máximo  $\lfloor |C|/2 \rfloor$  vértices de  $C$  podem estar simultaneamente em um conjunto independente. As desigualdades de clique definem facetas no politopo dos conjuntos independentes, dado pela envoltória convexa dos vetores

pertencentes a  $\mathcal{X}_s^{CI}$ . Já as desigualdades de *odd hole* não definem facetas no mesmo politopo no caso geral<sup>1</sup> [Pad73, NT74].

Uma outra forma de fortalecer a formulação (1) consiste em explorar melhor a questão da maximalidade. Além da restrição (II), pode se impor que  $x$  represente um conjunto independente maximal através da desigualdade abaixo:

$$x_{ij} + \sum_{v_{kl} \in N(v_{ij})} x_{kl} \geq 1, \forall v_{ij} \in V \quad (\text{V}),$$

onde  $N(v_{ij})$  representa o conjunto de vértices vizinhos a  $v_{ij}$  em  $G_P^{is}$ . Esta desigualdade diz que se um vértice não pertence a um conjunto independente maximal, então pelo menos um de seus vizinhos deve pertencer. Claramente, esta desigualdade é válida. Além disso, diferentemente da igualdade (II), como a desigualdade (V) tem uma atuação local (atua somente em um subgrafo de  $G_P^{is}$ ) ela será denominada de *desigualdade da maximalidade local*. Como veremos na seção 6, esta classe de desigualdades se mostra de grande valia computacional. Na figura 2 vemos duas soluções fracionárias ótimas, para o mesmo conjunto de pontos, obtidas utilizando a formulação (1). A solução (a), ao contrário da (b), não utiliza as desigualdades de maximalidade local. Nesta figura os segmentos cheios representam que as variáveis que os correspondem na formulação assumiram o valor 1, enquanto os segmentos tracejados representam variáveis que assumiram valores fracionários (entre 0 e 1). Vemos através da solução (a) que uma solução fracionária da formulação (1) possui áreas com grande concentração de segmentos cujos as variáveis correspondentes estão no suporte da solução ( $x_{ij} > 0$ ), e outras onde esta concentração é muito baixa. A solução (b) mostra que a inserção das desigualdes de maximalidade local diminui esta diferença impondo um número mínimo deste tipo de segmentos nas áreas de baixa concentração.

## O Politopo Monótono

Devido à igualdade (II) e as desigualdades (V), o politopo definido pela formulação apresentada não é de dimensão cheia. Este fato dificulta a comprovação teórica da força das desigualdades obtidas. Entretanto, modificando-se adequadamente a função objetivo e transformando-se a igualdade (II) e as desigualdades (V) em desigualdades do tipo ‘ $\leq$ ’ vemos que otimizar sobre o politopo original é equivalente a otimizar sobre o politopo monótono. Para isto, as seguintes alterações devem ser efetuadas. A nova função objetivo é definida da seguinte forma:

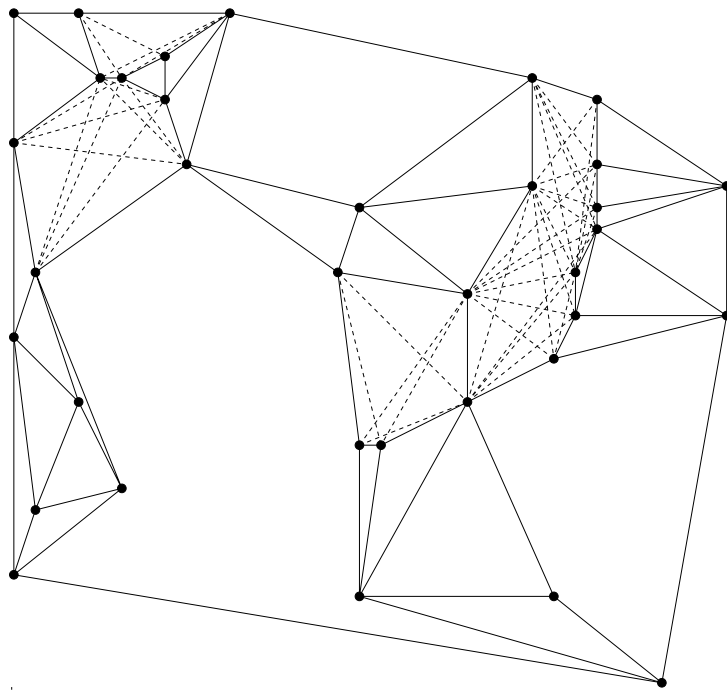
$$\max \sum_{\overline{ij} \in S(P)} (d - c_{ij})x_{ij}, \text{ onde } d = \max\{c_{ij} : \overline{ij} \in S(P)\} + 1.$$

A igualdade (II) é substituída pela desigualdade:

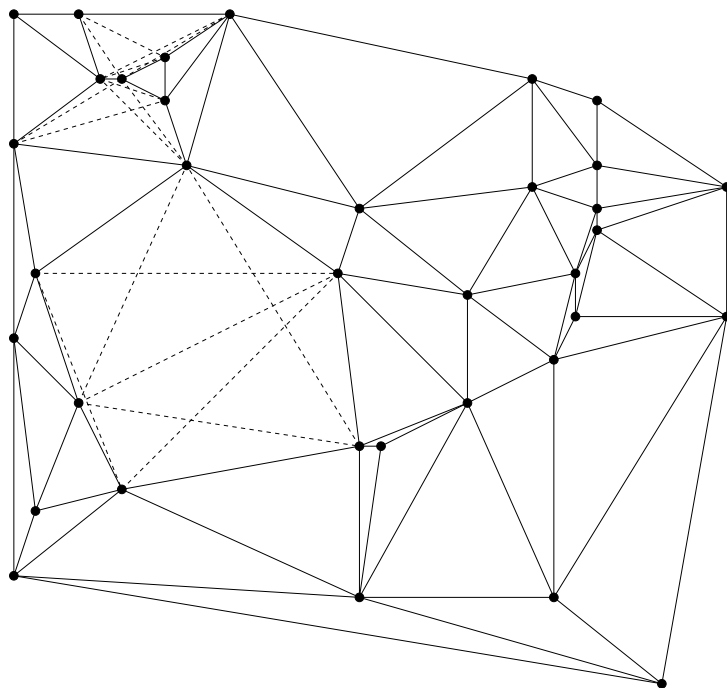
$$\sum_{\overline{ij} \in S(P)} x_{ij} \leq 3(n - 1) - h \quad (\text{VI}).$$

---

<sup>1</sup>Em [Pad73] Padberg apresenta um procedimento de *lifting* capaz de produzir desigualdades que definem facetas, no politopo dos conjuntos independentes, a partir de uma desigualdade de *odd hole*.



(a)



(b)

Figura 2: Duas soluções fracionárias ótimas para o PTCM.



E finalmente, visto que uma solução óptima para o PTCM sempre satisfaz (VI) na igualdade, podemos substituir as desigualdades da classe (V) pela classe de desigualdades abaixo:

$$\sum_{v_k \notin N(v_{ij}) \cup \{v_{ij}\}} x_{k\ell} \leq 3(n-1) - h - 1, \forall v_{ij} \in V \quad (\text{VII}).$$

Pode-se provar, de forma análoga a prova realizada para o politopo do problema do caixeiro viajante em [GP85], que uma solução óptima encontrada utilizando o politopo original possui o mesmo custo de uma solução óptima obtida utilizando o politopo monótono. Observe ainda que, o novo conjunto de restrições não representa mais apenas as triangulações planares de  $P$ , e sim todos os conjuntos  $T'_s \subseteq T_s(P)$ , para qualquer triangulação planar  $T_s(P)$  de  $P$ . Desta forma, todo resultado válido para o politopo dos conjuntos independentes é válido também para o politopo monótono, p.ex., a desigualdade de clique também define faceta no politopo monótono.

Com relação a nova desigualdade de maximalidade local (VII), apesar do bom desempenho computacional, mencionado anteriormente, ela não define faceta no caso geral. Para ver isto, considere a figura 3. Nesta figura, a desigualdade de maximalidade local com relação ao segmento  $\overline{24}$  é:

$$x_{12} + x_{14} + x_{15} + x_{16} + x_{23} + x_{25} + x_{26} + x_{34} + x_{45} + x_{56} \leq 8.$$

Contudo, a desigualdade abaixo também é válida, e domina a desigualdade anterior.

$$x_{12} + x_{14} + x_{15} + x_{16} + x_{23} + x_{25} + x_{26} + x_{34} + x_{45} + x_{56} + x_{36} \leq 8.$$

Portanto, a primeira desigualdade não define faceta no politopo monótono.

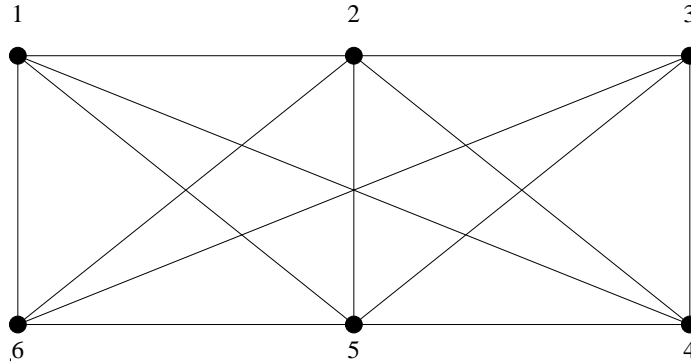


Figura 3: Exemplo no qual a desigualdade de maximalidade local não define faceta.

### 3.2 Explorando o Caráter Geométrico do Problema

Até o momento, com exceção da igualdade (II), todas as desigualdades apresentadas são desigualdades válidas para o problema do conjunto independente maximal em um grafo. A questão é se ainda é possível fortalecer a formulação apresentada utilizando outras propriedades geométricas do problema. Abaixo vemos que sim.

Como mencionado, a cardinalidade de uma triangulação planar  $T_s(P)$  é  $3(n-1)-h$ . Isto significa que qualquer conjunto  $S' \subseteq S(P)$ , tal que  $|S'| > 3(n-1)-h$ , não pode representar uma figura planar. Como isto é válido para qualquer conjunto  $P'$  de pontos no plano, com  $|P'| \geq 3$  e tal que os pontos do conjunto não sejam todos colineares, esta propriedade nos leva a uma nova desigualdade linear, a qual chamaremos de *desigualdade de planaridade*. Seja  $P' \subset P$  um conjunto de pontos no plano tal como descrito acima. Toda triangulação planar de  $P$  satisfaz a desigualdade abaixo:

$$\sum_{\overline{ij} \in S(P')} x_{ij} \leq 3(|P'| - 1) - h' \quad (\text{VIII}),$$

onde  $h'$  é o número de pontos de  $P'$  na fronteira de sua envoltória convexa. Esta desigualdade diz qual o número máximo de segmentos em  $S(P')$  que podem pertencer a uma triangulação planar. Observe que, no caso  $|P'| = 3$  esta desigualdade nunca é violada, pois neste caso  $|S(P')| = 3(n-1)-h' = 3$ . Assim, só precisamos considerar as desigualdades de planaridade referentes a conjuntos de pontos  $P' \subset P$  com  $|P'| \geq 4$ . Na figura 4 vemos outra solução fracionária ótima para o mesmo conjunto de pontos utilizado na figura 2. Esta solução foi obtida utilizando a mesma formulação que gerou a solução (b) da figura 2 acrescida da desigualdade de planaridade para o conjunto  $P'$  indicado na figura 4 por círculos brancos.

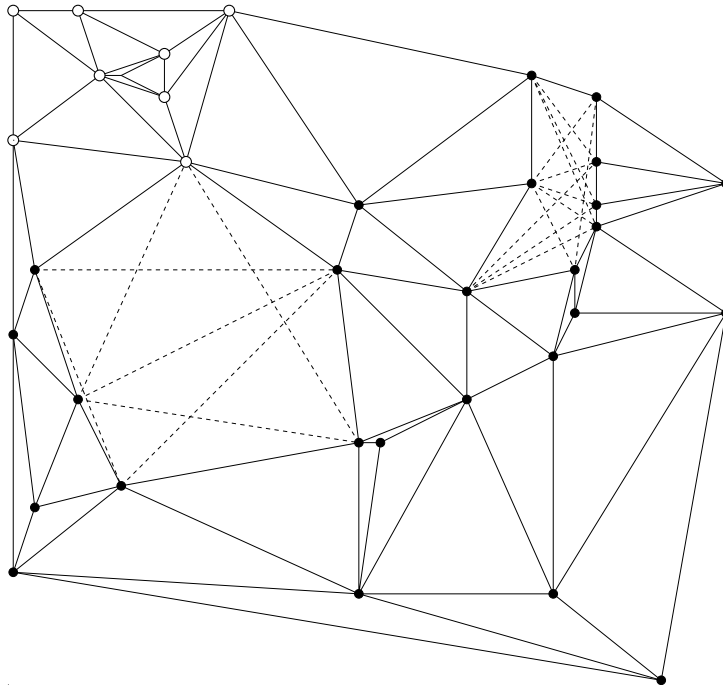


Figura 4: Uma solução fracionária ótima com a desigualdade de planaridade.

Assim como a desigualdade de maximalidade local, a desigualdade de planaridade também não define faceta no caso geral. Na figura 5, a desigualdade de planaridade referente aos pontos 1, 2, 3 e 4 é:

$$x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} \leq 5.$$

Se considerarmos agora o segmento  $\overline{ij}$ , vemos que a desigualdade abaixo não só é válida como domina a desigualdade anterior.

$$x_{12} + x_{13} + x_{14} + x_{23} + x_{24} + x_{34} + 3x_{ij} \leq 5.$$

Logo, a desigualdade de planaridade não define faceta no caso geral.

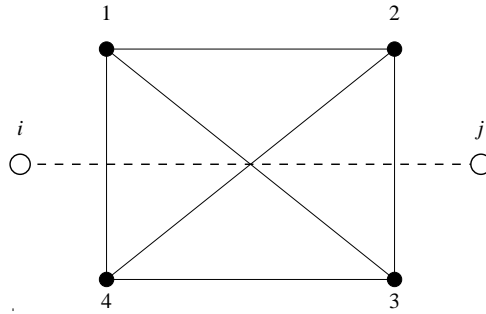


Figura 5: Exemplo no qual a desigualdade de planaridade não define faceta.

### Outras Desigualdades Válidas

Seja  $q$  um ponto interior da envoltória convexa de  $P$ , tal que  $q \notin P$  e  $q \cap \overline{ij} = \emptyset, \forall \overline{ij} \in S(P)$ . Seja  $S' \subseteq S(P)$  o conjunto de segmentos que definem a menor região do plano limitada por segmentos de  $S(P)$  contendo  $q$ . Uma outra desigualdade válida para o PTCM é a seguinte:

$$\sum_{\overline{ij} \in S'} x_{ij} \leq 3 \quad (\text{IX}).$$

Esta desigualdade diz que em uma triangulação planar o ponto  $q$  está no interior de um único *triângulo vazio* com vértices em  $P$ . Entende-se por triângulos vazios com vértices em  $P$  qualquer triângulo formado por segmentos de  $S(P)$  que não contenha pontos de  $P$  em seu interior. Embora, a validade desta desigualdade não seja tão intuitiva quanto a das anteriores, ela pode ser provada por contradição.

O problema desta desigualdade, assim como de outras desigualdades válidas que expresam certas propriedades geométricas das triangulações planares, é que, diferentemente da desigualdade de planaridade, elas não apresentam um bom desempenho computacional.

## 4 A Abordagem de PI Baseada em Triângulos

O PTCM também pode ser visto como o problema de determinar quais triângulos vazios estão em uma TCM. De fato, as triangulações planares, e por conseguinte a TCM, podem ser definidas alternativamente como se segue. Dados dois triângulos com extremos no plano,

diremos que estes se interceptam caso a região formada pelo conjunto de pontos na interseção dos dois triângulos tenha dimensão igual à 2. Uma triangulação planar do conjunto  $P$  é um conjunto maximal de triângulos vazios, com extremos em  $P$ , que não se interceptam. Uma TCM de  $P$ , pode então ser redefinida como, uma triangulação planar de  $P$  cuja soma dos comprimentos dos segmentos de reta definidos pelos lados de seus triângulos vazios é mínimo dentre todas as triangulações planares de  $P$ .

Seja  $\Delta(P)$  o conjunto de todos os triângulos vazios com extremos em  $P$ . Em vista do que foi dito acima, podemos considerar o PTCM como o problema de determinar um subconjunto maximal  $T_\Delta(P) \subseteq \Delta(P)$ , onde a soma total dos comprimentos dos segmentos de reta induzidos pelos triângulos em  $T_\Delta(P)$  seja mínima e tal que quaisquer dois triângulos em  $T_\Delta(P)$  não se interceptem. Note que, existe uma grande semelhança na forma como o problema foi definido sobre segmentos e sobre triângulos, pois em ambas as abordagens o problema consiste em determinar um conjunto independente maximal de elementos de um outro conjunto. De fato, também a versão sobre triângulos pode ser modelada como uma versão restrita do problema do conjunto independente em um grafo.

Dizemos que  $G_P^{it}(W, F)$  é o *grafo das interseções de triângulos* de  $P$  se: para cada triângulo  $\Delta_{klm} \in \Delta(P)$  existe um vértice  $v_{klm} \in W$  que o representa, e para cada dois triângulos  $\Delta_{klm}$  e  $\Delta_{nop}$  que se interceptam existe uma aresta  $(v_{klm}, v_{nop}) \in F$ . Assim, determinar uma triangulação planar de um conjunto  $P$  de pontos torna-se equivalente à determinar um conjunto independente maximal no grafo das interseções de triângulos de  $P$ .

Resta, para que possamos representar o PTCM através de PI, definir como atribuir custo aos triângulos vazios de  $\Delta(P)$  em termos de seus segmentos (lados). A cada triângulo  $\Delta_{klm}$ , cujo os lados são os segmentos  $\overline{kl}$ ,  $\overline{km}$  e  $\overline{lm}$ , atribuímos um custo  $c_{klm} = \alpha_{kl}c_{kl} + \alpha_{km}c_{km} + \alpha_{lm}c_{lm}$ , onde  $c_{ij}$  é o comprimento do segmento  $\overline{ij}$  e  $\alpha_{ij} = 1$  se o segmento  $\overline{ij}$  limita a fronteira da envoltória convexa de  $P$ , e  $1/2$  caso contrário. Note que, dada uma triangulação planar, os segmentos desta triangulação que limitam a envoltória convexa de  $P$  aparecem como lado de um único triângulo vazio, enquanto o restante dos segmentos aparecem como lado de dois triângulos. Assim, o custo de uma triangulação medido através da soma dos custos de seus triângulos é igual a soma dos comprimentos de segmentos induzidos por estes triângulos.

Podemos então formular o PTCM através de PI da seguinte forma: a cada triângulo  $\Delta_{klm} \in \Delta(P)$ , cujo custo é representado por  $c_{klm}$ , associa-se uma variável  $x_{klm} \in \{0, 1\}$ . Seja  $x$  o vetor das variáveis  $x_{klm}$ . É possível garantir que o valor de  $x_{ij}$  será 1 se e somente se  $\overline{ij}$  pertence a triangulação planar representada por  $x$  através da seguinte formulação:

$$\begin{aligned} \min \quad & \sum_{\Delta_{klm} \in \Delta(P)} c_{klm} x_{klm} \\ \text{Sujeito a} \quad & x \in \mathcal{X}_t^{CI} \quad (X) \\ & \sum_{\Delta_{klm} \in \Delta(P)} x_{klm} = 2(n-1) - h \quad (XI) \end{aligned} \quad (2)$$

onde  $\mathcal{X}_t^{CI}$  representa a família de todos vetores de incidência dos conjuntos independentes de  $G_P^{it}$ , e  $h$  o número de pontos de  $P$  sobre a fronteira de sua envoltória convexa. A restrição (X) diz dois triângulos de uma mesma triangulação planar não devem se interceptar. A

restrição (XI), a qual é obtida diretamente da restrição (II) da seção 3 e da fórmula de Euler, determina o número exato de triângulos em qualquer triangulação planar de um conjunto de pontos. Esta restrição é utilizada para garantir a condição de maximalidade do número de triângulos.

Note que esta formulação é idêntica a formulação (1), i.e., em ambas as formulações o vetor solução é o vetor de incidência de um conjunto independente maximal em um certo grafo. Assim, todos os resultados existentes para o problema do conjunto independente em um grafo utilizados na abordagem baseada em segmentos, a exemplo das desigualdades de aresta, clique e *odd hole*, e mesmo a desigualdade de maximalidade local, podem ser utilizados nesta abordagem. Além disso, observe que, também é possível obter um politopo monótono a partir do politopo definido pela formulação (2) da mesma forma como foi feito com o politopo definido pela formulação (1). Finalmente, note que a desigualdade de planaridade, apresentada na seção 3, também pode ser reescrita em termos de triângulos.

Assim, os principais resultados (desigualdades) válidos para a formulação baseada em segmentos também são válidos para a formulação baseada em triângulos. Contudo, enquanto o número de variáveis e restrições utilizadas na formulação (1) são limitados, respectivamente, por  $O(n^2)$  e  $O(n^4)$ , estes mesmos números na formulação (2) são limitados por  $O(n^3)$  e  $O(n^6)$ . Ainda assim, trabalhar com triângulos nos possibilita expressar outras propriedades geométricas das triangulação planares em termos de desigualdades lineares, as quais não se sabe como expressar desta forma em termos de segmentos.

#### 4.1 Explorando o Caráter Geométrico do Problema

Dada uma triangulação planar de  $P$ , todo segmento nesta triangulação que não limita a fronteira da envoltória convexa de  $P$  é lado de exatamente dois triângulos. Cada um destes triângulos está localizado em um dos semi-planos definidos pela reta que contém o segmento (conforme pode ser observado na figura 1). Além disso, se um certo segmento não está nesta triangulação, obviamente, nenhum dos triângulos que o possuem como lado deverá ser formado. Ou seja, se  $L_{k\ell}^+$  e  $L_{k\ell}^-$  são os semi-planos definidos pela reta que contém o segmento  $\overline{k\ell}$ , o número de triângulos em  $L_{k\ell}^+$  que possuem  $\overline{k\ell}$  como lado é igual ao número de triângulos em  $L_{k\ell}^-$  que também possuem  $\overline{k\ell}$  como lado. Esta propriedade das triangulações planares pode ser expressa na forma de uma equação linear da seguinte forma:

$$\sum_{\Delta_{k\ell m} \subset L_{k\ell}^+} x_{k\ell m} - \sum_{\Delta_{k\ell n} \subset L_{k\ell}^-} x_{k\ell n} = 0, \forall \overline{k\ell} \in S(P) \setminus CH(P) \quad (\text{XII}),$$

onde  $CH(P) \subseteq S(P)$  é o conjunto de segmentos que estão na fronteira da envoltória convexa de  $P$ .

Esta igualdade foi apresentada originalmente por Loera *et. al* [dLHSS96] para uma generalização das triangulações planares, onde os pontos pertencem a  $\mathbb{R}^d$  com  $d \geq 2$ , e as regiões limitadas formadas (triângulos quando  $d = 2$ ) não são necessariamente vazias. Nesse trabalho, os autores mostraram que esta classe de igualdades acrescida de certas igualdades não homogêneas são suficientes para garantir a validade da formulação. De fato, a seguinte formulação também é válida para o PTCM:

$$\begin{aligned}
\min \quad & \sum_{\Delta_{k\ell m} \in \Delta(P)} c_{k\ell m} x_{k\ell m} \\
\text{Sujeito a} \quad & \sum_{\Delta_{k\ell m} \subset L_{k\ell}^+} x_{k\ell m} - \sum_{\Delta_{k\ell n} \subset L_{k\ell}^-} x_{k\ell n} = 0, \quad \forall \overline{k\ell} \in S(P) \setminus CH(P) \quad (\text{XII}) \\
& \sum_{\Delta_{k\ell m} \in \Delta(P)} x_{k\ell m} = 2(n-1) - h \quad (\text{XI}) \\
& x_{k\ell m} \in \{0, 1\} \quad \forall \Delta_{k\ell m} \in \Delta(P) \quad (\text{XII})
\end{aligned} \tag{3}$$

A validade desta formulação não é tão intuitiva quanto a da formulação (2). A fim de demonstrá-la faremos uso do seguinte resultado:

**Lema 1** *Seja  $x'$  uma solução inteira viável para a formulação (3). A união de todos os triângulos  $\Delta_{k\ell m}$ , com  $x_{k\ell m} = 1$ , cobre toda a envoltória convexa de  $P$ .*

**Prova:** Obviamente, pela igualdade (XI), pelo menos parte da envoltória convexa de  $P$  deve ser coberta pelos triângulos mencionados. Contudo, por contradição, suponha que existam regiões dentro da envoltória convexa de  $P$  que não estão sendo cobertas por estes triângulos. Claramente, estas regiões são separadas das regiões cobertas por polilinhas (abertas ou fechadas). Assim, existe pelo menos um triângulo  $\Delta_{k\ell m}$ , com  $x_{k\ell m} = 1$ , tal que pelo menos um de seus lados é um dos segmentos da polilinha que separa a região que ele ajuda a cobrir de uma região não coberta (veja figura 6). Sem perda de generalidade, suponha que  $\overline{k\ell}$  seja este segmento e que  $\Delta_{k\ell m} \subset L_{k\ell}^+$ . Como  $\overline{k\ell}$  está na fronteira entre uma região coberta e uma não coberta, não existe nenhum triângulo  $\Delta_{k\ell n} \subset L_{k\ell}^-$ , com  $x_{k\ell n} = 1$ , e assim a desigualdade (XII) referente ao segmento  $\overline{k\ell}$  não está sendo satisfeita. Entretanto, isto contraria a hipótese de que  $x'$  satisfaz todas as restrições da formulação (3).  $\square$

De posse deste resultado, podemos finalmente provar a validade da formulação (3):

**Teorema 1** *A formulação (3) é válida para o PTCM, i.e., um vetor solução  $x'$ , cujo os elementos são inteiros, satisfaz todas as restrições da formulação (3) se e somente se é um vetor de incidência de uma triangulação planar.*

**Prova:** ( $\Rightarrow$ ) Seja  $T'_\Delta$  o conjunto de triângulos do qual  $x'$  é o vetor de incidência, i.e.,  $T'_\Delta = \{\Delta_{k\ell m} \in \Delta(P) : x_{k\ell m} = 1\}$ . Seja  $I(\Delta_{k\ell m})$ , onde  $\Delta_{k\ell m} \in T'_\Delta$ , o conjunto de triângulos  $\Delta_{nop} \in T'_\Delta$ , que interceptam  $\Delta_{k\ell m}$ . Como, pela restrição (XI),  $|T'_\Delta| = 2(n-1) - h$ , provar que  $T'_\Delta$  é uma triangulação planar de  $P$  é equivalente a mostrar que não existe um triângulo  $\Delta_{k\ell m} \in T'_\Delta$ , tal que  $I(\Delta_{k\ell m}) \neq \emptyset$ . Suponha, por contradição, que existe tal triângulo. Seja  $\mathcal{R}$  o conjunto de regiões limitadas formadas no interior do triângulo  $\Delta_{k\ell m}$  pelos lados dos triângulos em  $I(\Delta_{k\ell m}) \cup \{\Delta_{k\ell m}\}$  (ver figura 7). Seja  $R$  uma destas regiões. Existe pelo menos um segmento  $s$  que limita  $R$ , o qual está contido em um segmento de  $S(P)$  que, pela igualdade (XII), é lado de dois triângulos em  $I(\Delta_{k\ell m})$ . Seja  $\Delta_{nop}$  um destes triângulos e suponha que  $R$  e  $\Delta_{nop}$  estão contidos no mesmo semi-plano definido pela reta que contém

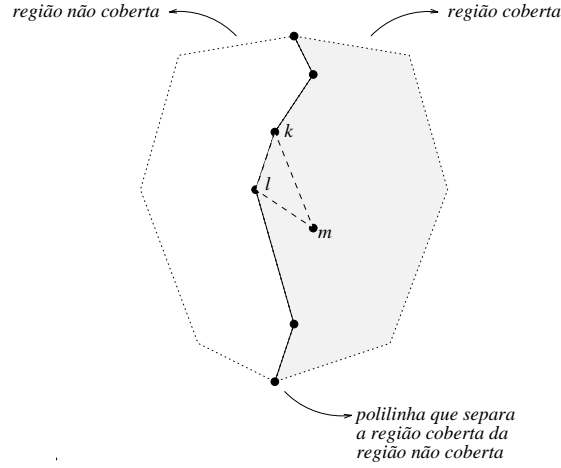


Figura 6: Exemplo de um triângulo cujo um dos lados é um dos segmentos de uma polilinha que separa uma região coberta de uma região não coberta.

s. Observe que,  $R \subset \Delta_{nop}$ , pois caso contrário a região  $R$  seria interceptada por um dos lados do triângulo  $\Delta_{nop}$  contrariando a definição de  $\mathcal{R}$ . Assim, concluímos que todas as regiões de  $\mathcal{R}$  são cobertas por pelo menos um triângulo em  $I(\Delta_{klm})$ .

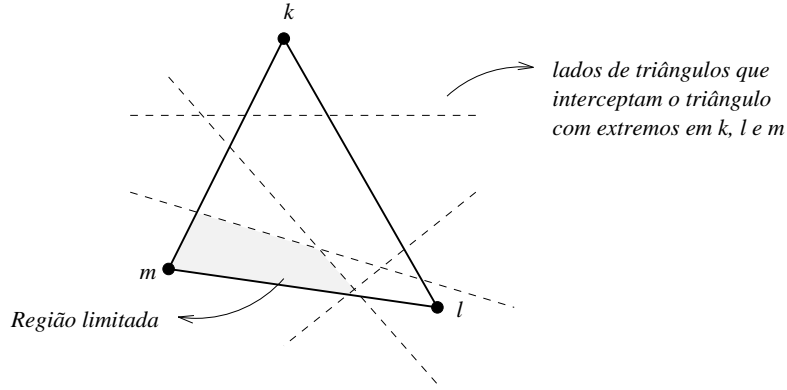


Figura 7: Exemplo de regiões limitadas formadas no interior do triângulo  $\Delta_{klm}$  pelos lados dos triângulos em  $I(\Delta_{klm}) \cup \{\Delta_{klm}\}$ .

Com base neste resultado e no fato que  $\bigcup_{R \in \mathcal{R}} R = \Delta_{klm}$ , se removermos  $\Delta_{klm}$  de  $T'_\Delta$  ( $x_{klm} = 0$ ) a região antes coberta por  $\Delta_{klm}$  continuará coberta pelos triângulos em  $I(\Delta_{klm})$ . Mais que isto, visto que os triângulos em  $T'_\Delta$ , pelo lema 1, cobrem toda a envoltória convexa de  $P$ , temos que os triângulos em  $T'_\Delta \setminus \{\Delta_{klm}\}$  também cobrirão toda a envoltória convexa de  $P$ . De fato, se removermos iterativamente todos os triângulos  $\Delta_{klm}$ , com  $I(\Delta_{klm}) \neq \emptyset$ , de  $T'_\Delta$  restará um conjunto  $T''_\Delta$  de triângulos que cobrem toda a envoltória convexa de  $P$ , e que dois a dois não se interceptam. Obviamente,  $T''_\Delta$  é uma triangulação planar. Entretanto, se  $T'_\Delta \neq T''_\Delta$ , como  $T''_\Delta$  é uma triangulação planar, teríamos que  $|T'_\Delta| > 2(n-1) - h$ , e  $x'$  não satisfaria a restrição (XI), contrariando a hipótese de que  $x'$  é uma solução viável para

a formulação (3). Concluindo, não é possível remover nenhum dos triângulos da solução original  $T'_\Delta$ , i.e., para todo  $\Delta_{klm} \in T'_\Delta$  temos que  $I(\Delta_{klm}) = \emptyset$ .

( $\Leftarrow$ ) Trivial. □

Assim temos que, a formulação (3) é válida para o PTCM, e em contraposição com a formulação (2), possui apenas  $O(n^2)$  restrições. Além disso, o politopo definido por esta formulação é igual a envoltória convexa das soluções inteiras para o caso onde todos os pontos de  $P$  estão sobre a fronteira de sua envoltória convexa [dLHSS96]. Ou seja, neste caso, a solução da relaxação linear desta formulação é sempre inteira e o problema pode ser resolvido em tempo polinomial. Esta é a formulação baseada em triângulos que será avaliada na seção 6. Como veremos, para todas as instâncias testadas até o momento a relaxação linear desta formulação sempre forneceu uma solução inteira.

## 5 Algoritmos Utilizados

Para resolver o PTCM empregando as formulações propostas utilizamos uma abordagem em dois níveis. Inicialmente, utilizamos algumas rotinas de pré-processamento, com a finalidade de reduzir o número de variáveis do problema, i.e., reduzir o tamanho do programa linear correspondente. Então utilizamos algoritmos baseados em relaxações lineares das formulações propostas, para determinar o vetor de incidência de uma TCM do conjunto de pontos em questão. No caso da formulação (3), como mencionado, para todas as instâncias testadas até o momento, bastou resolver uma relaxação linear para obter tal vetor. O caso da formulação (1) é diferente. Via de regra, uma relaxação linear desta formulação, mesmo utilizando as desigualdades de maximalidade local, fornece uma solução fracionária. Para resolver o problema utilizando esta formulação implementamos um algoritmo *branch-and-cut*. Abaixo discutimos sucintamente estes algoritmos.

### Rotinas de Pré-processamento

Ao todo utilizamos três rotinas de pré-processamento distintas. A primeira delas, utiliza um resultado publicado em [BKMS96] para computar um conjunto de segmentos que está em qualquer triangulação de custo mínimo. O conjunto destes segmentos é chamado pelos autores de *1.17682-esqueleto*, e o algoritmo implementado é capaz de computá-lo em tempo  $O(n^2)$ . Esta rotina nos permite eliminar do problema todos os segmentos que interceptam um dos segmentos deste conjunto, no caso da abordagem baseada em segmentos, ou todos os triângulos que possuem um lado que intercepta um dos segmentos deste conjunto, no caso da abordagem baseada em triângulos.

As duas outras rotinas de pré-processamento restantes utilizam-se de relaxações lineares preliminares para tentar reduzir o tamanho do programa linear final, e são utilizadas apenas para a formulação baseada em segmentos. Dada um solução aproximada para um problema de PI com variáveis que assumem apenas os valores 0 ou 1 (no nosso caso a triangulação gulosa), e uma relaxação linear para este mesmo problema, é possível, dependendo dos custos reduzidos das variáveis e da diferença entre os valores da solução aproximada e da solução da



relaxação linear, determinar que algumas destas variáveis não podem assumir o valor 1 na solução ótima [JRT94]. Este resultado é utilizado pela segunda rotina de pré-processamento com o objetivo de eliminar tais variáveis do problema. Finalmente, utilizamos um resultado publicado em [NT75], segundo o qual as variáveis que assumem valores binários (0 ou 1) na solução ótima da relaxação linear de uma formulação para o problema do conjunto independente que utiliza somente desigualdades de arestas, também assumem os mesmos valores em alguma solução ótima do problema. Visto que uma triangulação planar é um conjunto independente de segmentos, aplicamos este resultado a fim de eliminar qualquer variável que assumo o valor 0 nesta relaxação.

### O Branch-and-Cut para a Abordagem Baseada em Segmentos

Um algoritmo *branch-and-cut* consiste basicamente de um algoritmo *branch-and-bound*, onde a cada nó da árvore de enumeração executa-se um algoritmo *cutting-plane*. Considere uma iteração onde o nó  $i$  da árvore de enumeração está sendo analisado. Seja  $x^i$  a solução ótima obtida com a relaxação linear. Se  $x^i$  contém elementos fracionários, utiliza-se de uma *rotina de separação* para encontrar uma nova desigualdade válida a qual  $x^i$  não satisfaça. Se a rotina obtém sucesso a nova desigualdade é adicionada a relaxação linear (*cutting-plane*), e reinicia-se o processo. Este ciclo é quebrado quando o *bound* da relaxação linear é pior que o *bound* fornecido por uma solução inteira já conhecida, ou  $x^i$  não possui elementos fracionários, ou se a rotina de separação não conseguir encontrar uma nova desigualdade válida violada por  $x^i$ . Neste último caso, uma variável fracionária é escolhida e faz-se um *branching* sobre ela. A solução final é a melhor solução inteira obtida no processo<sup>2</sup>.

Neste tipo de algoritmo, as rotinas mais intrinsecamente ligadas ao problema que está se atacando são as rotinas de separação. Nossa implementação, em especial, é capaz de separar três tipos de desigualdades (cortes): as desigualdades de clique, *odd hole*, e de planaridade. As rotinas de separação para as desigualdades de clique e de *odd hole*, são as mesmas descritas por Nemhauser e Sigismondi em [NS92], para o problema do conjunto independente. A rotina utilizada para separar desigualdades de planaridade é apresentada a seguir

**Separando desigualdades de planaridade.** Seja  $x'$  uma solução fracionária viável para uma formulação (1). Uma rotina de separação de desigualdades de planaridade deve tentar identificar conjuntos de pontos  $P' \subset P$ , tal que  $\sum_{\overline{ij} \in S(P')} x_{ij} > 3(|P'| - 1) - h'$ , onde  $h'$  representa o número de pontos em  $P'$  sobre a fronteira de sua envoltória convexa. Seja  $T'_s \subseteq S(P)$  o conjunto de segmentos tal que as variáveis que os representam, na formulação, assumem valores fracionários em  $x'$ , i.e.,  $T'_s = \{\overline{ij} \in S(P) : 0 < x_{ij} < 1\}$ . Observe que, se um certo conjunto de pontos  $P' \subset P$  não satisfaz a desigualdade de planaridade então, obrigatoriamente, o conjunto de segmentos  $S(P') \cap T'_s$  não é vazio, e existe interseção entre seus segmentos. Assim, a idéia da nossa heurística é identificar conjuntos maximais de pontos  $P' \subset P$ , com  $|P'| \geq 4$ , tal que para todo ponto em  $P'$  exista um segmento em  $S(P') \cap T'_s$ , do qual ele é um dos extremos, que intercepta pelo menos um outro segmento

---

<sup>2</sup>Uma descrição detalhada do *framework*, e dos detalhes de implementação, de um algoritmo *branch-and-cut* podem ser encontrados em [JRT94].

em  $S(P') \cap T'_s$ . Para cada um destes conjuntos  $P'$ , a heurística verifica se a desigualdade de planaridade referente a ele está sendo violada. Em caso afirmativo, esta desigualdade é gerada, i.e, ela é acrescentada à formulação de PI corrente. Caso contrário, remove-se iterativamente pontos de  $P'$  até que  $|P'| = 3$ , ou até que um dos subconjuntos gerados no processo não satisfaça a desigualdade de planaridade. Neste último caso, a desigualdade violada é gerada.

O algoritmo utilizado é apresentado de forma mais detalhada na figura 8, onde  $\phi(P') = 3(|P'| - 1) - h'$  e  $h'$  é o número de pontos em  $P'$  sobre a fronteira de sua envoltória convexa. Os passos 1 e 2 do algoritmo utilizam o grafo das interseções de segmentos de  $P$ ,  $G_P^{is}$ , para identificar conjuntos de segmentos de  $T'_s$  onde todo segmento intercepta pelo menos um outro segmento do mesmo conjunto. No algoritmo estes conjuntos de segmentos são representados pelos vértices dos componentes conexos em  $\mathcal{C}$ . Então, no passo 3, para cada um destes conjuntos de segmentos  $Q$ , tenta-se determinar se um subconjunto de seus pontos extremos viola a desigualdade de planaridade. Neste sentido, a primeira tarefa realizada é identificar o conjunto  $\overline{P}$  de pontos extremos de  $Q$ , o que é feito no passo 3(a). Em seguida, no passo 3(b), testa-se se a desigualdade de planaridade referente  $\overline{P}$  está satisfeita. Em caso afirmativo, escolhe-se um ponto  $q \in \overline{P}$ , faz-se  $\overline{P} := \overline{P} \setminus \{q\}$ , e repete-se o passo 3(b) para o novo conjunto de pontos  $\overline{P}$ . Este processo pára se o conjunto de pontos corrente  $\overline{P}$  não satisfaz a desigualdade de planaridade, ou se  $|\overline{P}| = 3$ . No primeiro caso,  $|\overline{P}| \geq 4$ , e assim o passo 3(c) gera a desigualdade de planaridade correspondente a  $\overline{P}$ . Observe que, a remoção de pontos de  $\overline{P}$ , no passo 3(b), é feita de forma gulosa, onde procura-se produzir com a remoção de um ponto  $q \in \overline{P}$  o subconjunto de pontos  $\overline{P} \setminus \{q\}$  mais próximo à violação de uma desigualdade de planaridade. A complexidade de tempo deste algoritmo é dominada pelo passo 3(b)i, e é igual a  $O(|T'_s|n^2 \lg n) = O(|S(P')|n^2 \lg n) = O(n^4 \lg n)$ , se utilizarmos o algoritmo *Graham-scan* [CLR90], para computar  $\phi(\overline{P})$ , no passo 3(b). Ainda assim, na prática, este algoritmo apresenta um bom desempenho computacional. Isto se deve ao fato que, em geral,  $|T'_s| \ll |S(P')|$ , i.e., o número de variáveis que assumem valores fracionários em uma relaxação linear da formulação (1) é pequeno quando comparado ao total de variáveis desta mesma formulação.

## 6 Resultados Computacionais

Apresentamos nesta seção alguns resultados preliminares quanto à resolução exata do PTCM utilizando as formulações propostas, tal como descrito na seção anterior. Todas as instâncias utilizadas foram geradas determinando-se aleatoriamente (utilizando a rotina `random` do C) a posição de cada ponto dentro de uma grade de tamanho pré-estipulado. Os tempos de computação são apresentados em segundos, e foram obtidos em uma SUN SPARC 1000, utilizando o CPLEX 3.0 na resolução das relaxações lineares.

### Desempenho Computacional da Abordagem Baseada em Segmentos

Em todos os testes computacionais realizados com a formulação baseada em segmentos, utilizamos as desigualdades de aresta para garantir que a formulação inicial fornecida ao *branch-and-cut* satisfaça a restrição (I) da formulação (1). Ou seja, estas formulações iniciais

- 
1. Determinar o subgrafo  $\widehat{G}_P^{is}(\widehat{V}, \widehat{E})$  de  $G_P^{is}(V, E)$  induzido por  $\widehat{V}$ , onde  $\widehat{V} = \{v_{ij} \in V : \overline{ij} \in T'_s\}$ .
  2. Identificar o conjunto  $\mathcal{C}$  de componentes conexos de  $\widehat{G}_P^{is}$ .
  3. Para cada componente  $\overline{G}_P^{is}(\overline{V}, \overline{E}) \in \mathcal{C}$ 
    - (a) Determinar o conjunto  $\overline{P} \in P$  de todos os pontos que são extremos dos segmentos representados em  $\overline{V}$ .
    - (b) Enquanto  $|\overline{P}| \geq 4$  e  $\sum_{\overline{ij} \in S(\overline{P})} x_{ij} \leq \phi(\overline{P})$ 
      - i.  $pv[p] := \phi(\overline{P} \setminus \{p\}) - \sum_{\overline{ij} \in S(\overline{P} \setminus \{p\})} x_{ij}, \forall p \in \overline{P}$ .
      - ii.  $q := \min_p \{pv[p] : p \in \overline{P}\}$ .
      - iii.  $\overline{P} := \overline{P} \setminus \{q\}$ .
    - (c) Se  $|\overline{P}| \geq 4$   
Gerar a desigualdade de planaridade referente ao conjunto de pontos  $\overline{P}$ .
- 

Figura 8: Rotina de separação de desigualdades de planaridade.

sempre continham pelo menos a desigualdades (VI) e as desigualdades de arestas. Contudo, os melhores resultados alcançados, em termos de tempo de computação, foram obtidos aliando-se formulações iniciais que utilizavam as restrições de maximalidade local, com um *branch-and-cut* capaz de gerar os três tipos de cortes mencionados na seção anterior.

Para que se possa avaliar o impacto da inserção da desigualdade de maximalidade local na formulação inicial, apresentamos na tabela 1 uma comparação dos tempos de computação obtidos pelo *branch-and-cut* (utilizando apenas cortes do tipo clique e *odd hole*) para dois tipos de formulações iniciais distintas. Ambas as formulações contém a desigualdade (VI) e as desigualdades de aresta. A diferença entre as duas reside no fato que apenas uma delas utiliza as restrições de maximalidade local. Cada linha da tabela apresenta a instância que obteve o pior tempo de CPU entre quatro outras de mesmo tamanho. Na primeira coluna encontra-se o número de pontos de cada instância. Nas colunas 2 e 3 encontramos, respectivamente, os tempos de computação gastos em pré-processamento e no *branch-and-cut*, para a formulação que contém as desigualdades de maximalidade local. As colunas 4 e 5 trazem, respectivamente, os tempos de computação gastos em pré-processamento e no *branch-and-cut*, para a formulação que não contém tais desigualdades. A diferença entre os tempos de pré-processamento (colunas 2 e 4) se deve ao fato de que uma das rotinas executadas nesta fase resolve uma relaxação linear do problema, o que se mostrou mais rápido quando a formulação contém a desigualdade de maximalidade local. É fácil observar que as desigualdades de maximalidade local melhoram sobremaneira o desempenho

computacional do algoritmo.

$n$	Com a desigualdade de de maximalidade local		Sem a desigualdade de de maximalidade local	
	Tempo de pré-processamento	Tempo de <i>branch-and-cut</i>	Tempo de pré-processamento	Tempo de <i>branch-and-cut</i>
20	1	0	1	4
25	3	0	4	54
30	3	0	12	357
35	8	5	13	3206

Tabela 1: Verificação do desempenho computacional da desigualdade (VII).

Visto que o número de desigualdades de planaridade para certas configurações de pontos é  $\Theta(\sum_{n=4}^k \binom{n}{k}) = \Theta(2^n)$ , mesmo para pequenos valores de  $n$ , é impraticável inserir todas estas desigualdades na formulação inicial. Assim, como mencionado na seção anterior, estas desigualdades são inseridas na formulação na forma de cortes gerados pelo *branch-and-cut*. Na tabela 2 apresentamos mais uma comparação de desempenhos computacionais. Desta vez, utilizamos uma mesma formulação inicial (contendo as desigualdades de maximalidade local). O objetivo aqui é comparar o desempenho do *branch-and-cut* quando este utiliza, e quando não utiliza, os cortes de planaridade. Assim como na tabela anterior, cada linha da tabela apresenta a instância que obteve o pior tempo de CPU entre quatro outras de mesmo tamanho. Na coluna 1 encontra-se o número de pontos de cada instância. Na coluna 2 o tempo de computação gasto em pré-processamento. As colunas 3 e 4 apresentam o tempo de computação do *branch-and-cut* dependendo dos cortes que este utiliza. Assim como no caso das desigualdades de maximalidade local, observe que, a utilização das desigualdades de planaridade proporciona um incremento no desempenho computacional do algoritmo.

$n$	Com a desigualdade planaridade		Sem a desigualdade planaridade
	Tempo de pré-processamento	Tempo de <i>branch-and-cut</i>	Tempo de <i>branch-and-cut</i>
50	49	65	78
55	53	177	340
60	117	5158	4561
65	110	9902	23674
70	151	16847	28848

Tabela 2: Verificação do desempenho computacional da desigualdade (VIII).

Um observação interessante com relação aos resultados obtidos, diz respeito a variação do tempo de computação para instâncias do mesmo tamanho. Um exemplo desta variação

são as instâncias de 65 pontos. Enquanto a instância apresentada gastou 9902 segundos para ser resolvida (tabela 2), a instância que gastou menos tempo levou apenas 48 segundos. Isto é um indicativo que a dificuldade deste problema, nesta abordagem, está mais intrinsecamente ligada à disposição geométrica dos pontos que ao tamanho da instância.

### Desempenho Computacional da Abordagem Baseada em Triângulos

Conforme mencionado, para todas instâncias testadas até o momento, a relaxação linear da formulação (3) sempre forneceu uma solução inteira, i.e., um vetor de incidência de uma TCM. Este fato fez com que não houvesse grande variação nos tempos obtidos para resolver instâncias do mesmo tamanho. Assim, nos testes realizados utilizando esta abordagem, a dificuldade de se resolver o problema se mostrou mais ligada ao tamanho da instância que à disposição geométrica dos pontos. De fato, o tempo de computação gasto para resolver qualquer instância testada foi inferior ao dobro do tempo gasto pela rotina de pré-processamento, a qual possui uma complexidade de tempo igual a  $O(n^2)$ .

Para ilustrar os fatos acima, apresentamos na tabela 3 os resultados obtidos, utilizando a formulação (3), para instâncias de 50 a 350 pontos. Cada linha da tabela apresenta a média dos valores de cinco instâncias do mesmo tamanho. O tamanho das instâncias representadas nestas linhas é apresentado na coluna 1. As colunas 2 e 3 apresentam respectivamente  $|\Delta(P)|$  e  $|S(P)|$ . A coluna 4 apresenta o tempo gasto com o pré-processamento. As colunas 5 e 6 apresentam os números de variáveis e restrições utilizadas nas relaxações lineares, i.e., o número de triângulos e segmentos que não foram eliminados na fase de pré-processamento. A coluna 7 apresenta o tempo de computação gasto para resolver a relaxação linear resultante. E finalmente, a coluna 8 apresenta o tempo total gasto no processo.

Uma primeira observação possível, com relação aos resultados apresentados na tabela 3, diz respeito ao desempenho da rotina de pré-processamento. Observe que, em alguns casos esta rotina consegue eliminar 90% dos triângulos em  $\Delta(P)$ . Para que se possa ter uma idéia da quantidade de segmentos de uma triangulação planar que esta rotina consegue identificar, apresentamos nas figuras 9 e 10, respectivamente, o 1.17682-esqueleto e uma TCM de um mesmo conjunto de 350 pontos. Outro ponto relevante, com relação aos resultados apresentados na tabela 3, é o tempo de computação. Enquanto na abordagem baseada em segmentos foi necessário 16847 segundos (aproximadamente 4,6 horas) para resolver um problema de 70 pontos, utilizando a formulação (3) fomos capazes de resolver instâncias de 350 pontos em apenas 2865 segundos (aproximadamente 48 minutos).

Finalmente, vale ressaltar que, apesar de que nos testes realizados, utilizando a formulação (3) para resolver o PTCM, a solução ótima da relaxação foi sempre inteira, o polítopo definido por esta formulação não equivale a envoltória convexa das soluções inteiras. Em alguns testes realizados, verificamos que, na maioria dos casos, basta inverter o sentido da função objetivo (maximizar) para que a solução ótima da relaxação linear seja fracionária.

Tabela 3: Resultados computacionais obtidos com a formulação (3).

$n$	Pré-processamento			Relaxação linear			Tempo total
	# de triângulos	# de segmentos	Tempo	# de variáveis	# de restrições	Tempo	
50	3875	1215	3	1518	523	2	5
75	9431	2764	11	3003	960	10	21
100	17647	4935	27	4602	1429	22	49
125	28463	7731	62	5886	1813	40	102
150	41505	11153	119	7643	2311	80	199
175	57362	15202	184	7908	2623	115	299
200	75100	19874	299	10170	3088	149	448
225	95510	25170	403	12300	3687	228	631
250	118641	31088	566	13699	4094	275	841
275	145274	37627	798	15561	4613	354	1152
300	171917	44796	1069	17660	5197	557	1626
325	201975	52586	1226	19284	5627	696	1922
350	234755	60998	1865	21518	6257	1000	2865

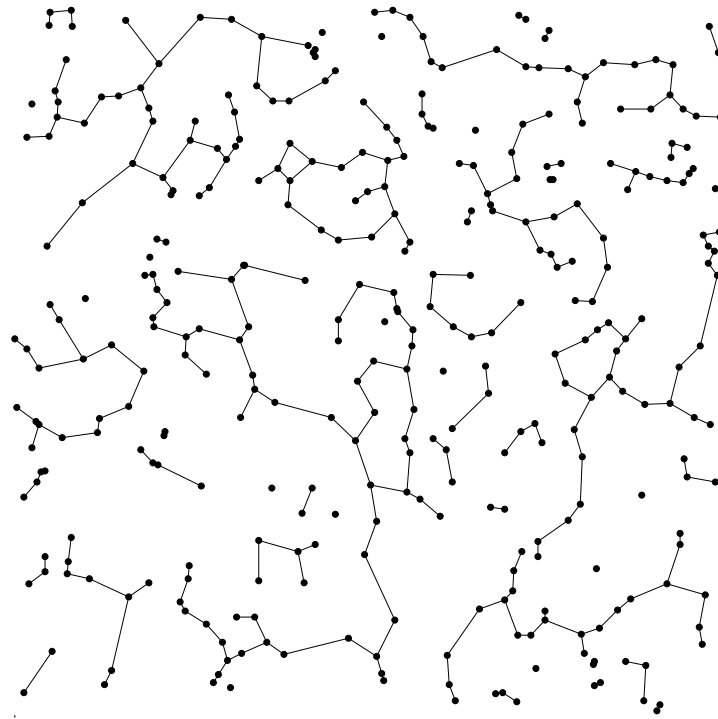


Figura 9: O 1.17682-esqueleto de 350 pontos (resultado do pré-processamento).

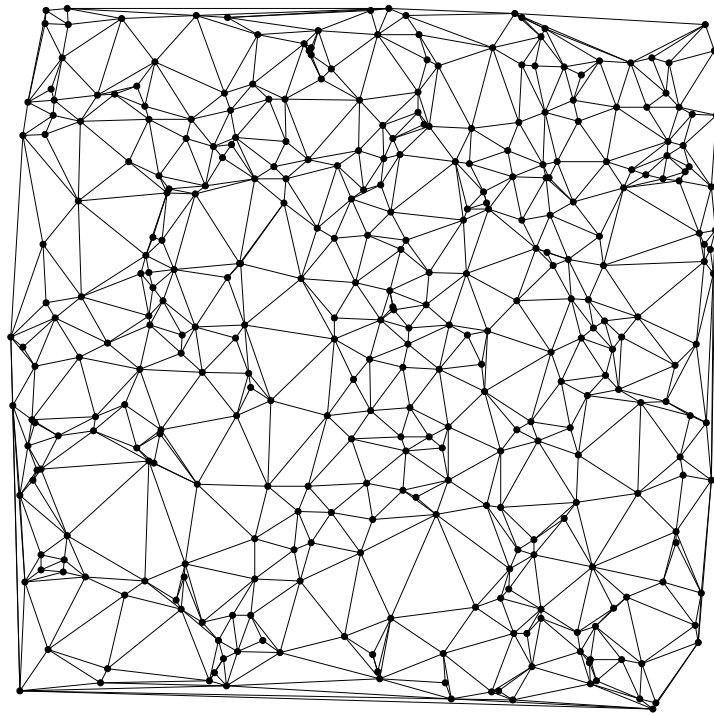


Figura 10: Uma TCM de 350 pontos.

## 7 Conclusões

Apresentamos neste artigo duas formulações de PI distintas. Mostramos que, em ambas as abordagens existe uma equivalência entre o PTCM e uma versão restrita do problema do conjunto independente em um grafo, a qual nos possibilita utilizar resultados já conhecidos para o politopo dos conjuntos independentes na caracterização do politopo das triangulações planares. Além disso, mostramos como fortalecer as formulações propostas através de novas desigualdades válidas que expressam certas propriedades inerentemente geométricas do problema. E finalmente, utilizamos estas formulações para resolver certas instâncias do problema de forma exata.

Os resultados apresentados, em especial os obtidos com a formulação (3), parecem indicar a viabilidade de PI na resolução de exata do PTCM. Como mencionado, na prática o tempo de computação necessário para resolver uma instância de tamanho  $n$  se mostrou da ordem de  $O(n^2)$ . Entretanto, um problema desta abordagem é a quantidade de memória exigida para tais computações que é da ordem de  $O(n^5)$ . No sentido de contornar este problema, estamos empenhados no momento na implementação de um algoritmo de geração de colunas, o qual possibilita resolver o problema sem termos que considerar todas as variáveis da formulação simultaneamente. Acreditamos que este algoritmo nos possibilite resolver instâncias maiores, e possivelmente com um tempo de computação menor.

## Referências

- [BKMS96] P. Belleville, M. Keil, M. McAllister, and J. Snoeyink. On computing edges that are in all minimum-weight triangulations. Two-page description of video in ACM SCG '96 Video Review, 1996. <http://www.cs.ubc.ca/spider/snoeyink/papers.html>.
- [CGT95] S. Cheng, M. Golin, and J. Tsang. Expected case analysis of  $\beta$ -skeletons with applications to the construction of minimum-weight triangulations. In *Proceedings of the Seventh Canadian Conference on Computational Geometry*, 1995.
- [CLR90] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.
- [dLHSS96] J. A. de Loera, S. Hosten, F. Santos, and B. Sturmfels. The polytope of all triangulations of a point configuration. 12th European Workshop on Computational Geometry, March 1996.
- [DM96] M. T. Dikerson and M. H. Montague. The exact minimum weight triangulation. to appear in *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, 1996.
- [DRA94] R. L. S. Drysdale, G. Rote, and O. Aichholzer. A simple linear time greedy triangulation algorithm for uniformly distributed points. October 1994.



- [Gil79] P. D. Gilbert. *New Results on Planar Triangulations*. PhD thesis, University of Illinois, 1979. Report No. UILUENG 78 2243.
- [GP85] M. Grötschel and M. W. Padberg. Polyhedral theory. In E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan, and D. B. Shmoys, editors, *The Traveling Salesman Problem*, chapter 8, pages 251–306. 1985.
- [HP94] L. S. Heath and S. V. Pemmaraju. New results for the minimum weight triangulation problem. *Algorithmica*, 12:533–552, 1994.
- [JRT94] M. Jünger, G. Reinelt, and S. Thienel. Practical problem solving with cutting plane algorithms in combinatorial optimization. Technical Report 94.156, Universität zu Köln, Germany, 1994.
- [Lev87] C. Levcopoulos. An  $\Omega(\sqrt{n})$  lower bound for the greedy triangulation. *Information Processing Letters*, 25(4):247–251, 1987.
- [LL87] C. Levcopoulos and A. Lingas. On approximation behavior of greedy triangulation for convex polygons. *Algorithmica*, 2:175–193, 1987.
- [Llo77] E. L. Lloyd. On triangulations of a set of points in the plane. In *Proc. 18th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 228–240, 1977.
- [MZ79] G. K. Manacher and A. L. Zobrist. Neither the greedy nor the delaunay triangulation of a planar point set approximates the optimal triangulation. *Information Processing Letters*, 9(1):31–34, 1979.
- [NS92] G. L. Nemhauser and G. Sigismondi. A strong cutting plane/branch-and-bound algorithm for node packing. *Journal of Operational Research Society*, 43(5):443–457, 1992.
- [NT74] G. L. Nemhauser and L. E. Trotter. Properties of vertex packing and independence system polyhedra. *Mathematical Programming*, 6:48–61, 1974.
- [NT75] G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.
- [Pad73] M. W. Padberg. On the facial structure of set packing polyhedra. *Mathematical Programming*, 5:199–215, 1973.
- [PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, 1985.
- [WA86] C. A. Wang and J. K. Aggarwal. Surface reconstruction and representation of 3d-scenes. *Pattern Recognition*, 19:223–232, 1986.
- [Yan95] B. T. Yang. A better subgraph of the minimum weight triangulation. *Information Processing Letters*, 56:255–258, 1995.

## Relatórios Técnicos – 1995

- 95-01 **Paradigmas de algoritmos na solução de problemas de busca multidimensional**, *Pedro J. de Rezende, Renato Fileto*
- 95-02 **Adaptive enumeration of implicit surfaces with affine arithmetic**, *Luiz Henrique de Figueiredo, Jorge Stolfi*
- 95-03 **W3 no Ensino de Graduação?**, *Hans Liesenberg*
- 95-04 **A greedy method for edge-colouring odd maximum degree doubly chordal graphs**, *Celina M. H. de Figueiredo, João Meidanis, Célia Picinin de Mello*
- 95-05 **Protocols for Maintaining Consistency of Replicated Data**, *Ricardo Anido, N. C. Mendonça*
- 95-06 **Guaranteeing Full Fault Coverage for UIO-Based Methods**, *Ricardo Anido and Ana Cavalli*
- 95-07 **Xchart-Based Complex Dialogue Development**, *Fábio Nogueira de Lucena, Hans K.E. Liesenberg*
- 95-08 **A Direct Manipulation User Interface for Querying Geographic Databases**, *Juliano Lopes de Oliveira, Claudia Bauzer Medeiros*
- 95-09 **Bases for the Matching Lattice of Matching Covered Graphs**, *Cláudio L. Lucchesi, Marcelo H. Carvalho*
- 95-10 **A Highly Reconfigurable Neighborhood Image Processor based on Functional Programming**, *Neucimar J. Leite, Marcelo A. de Barros*
- 95-11 **Processador de Vizinhança para Filtragem Morfológica**, *Ilka Marinho Barros, Roberto de Alencar Lotufo, Neucimar Jerônimo Leite*
- 95-12 **Modelos Computacionais para Processamento Digital de Imagens em Arquiteturas Paralelas**, *Neucimar Jerônimo Leite*
- 95-13 **Modelos de Computação Paralela e Projeto de Algoritmos**, *Ronaldo Parente de Menezes e João Carlos Setubal*
- 95-14 **Vertex Splitting and Tension-Free Layout**, *P. Eades, C. F. X. de Mendonça N.*
- 95-15 **NP-Hardness Results for Tension-Free Layout**, *C. F. X. de Mendonça N., P. Eades, C. L. Lucchesi, J. Meidanis*
- 95-16 **Agentes Replicantes e Algoritmos de Eco**, *Marcos J. C. Euzébio*
- 95-17 **Anais da II Oficina Nacional em Problemas Combinatórios: Teoria, Algoritmos e Aplicações**, *Editores: Marcus Vinicius S. Poggi de Aragão, Cid Carvalho de Souza [Not available.]*

- 95-18 **Asynchronous Teams: A Multi-Algorithm Approach for Solving Combinatorial Multiobjective Optimization Problems**, *Rosiane de Freitas Rodrigues, Pedro Sérgio de Souza*
- 95-19 **wxWindows: Uma Introdução**, *Carlos Neves Júnior, Tallys Hoover Yunes, Fábio Nogueira de Lucena, Hans Kurt E. Liesenberg*
- 95-20 **John von Neumann: Suas Contribuições à Computação**, *Tomasz Kowaltowski*
- 95-21 **A Linear Time Algorithm for Binary Phylogeny using PQ-Trees**, *J. Meidanis and E. G. Munuera*
- 95-22 **Text Structure Aiming at Machine Translation**, *Horacio Saggion and Ariadne Carvalho*
- 95-23 **Cálculo de la Estructura de un Texto en un Sistema de Procesamiento de Lenguaje Natural**, *Horacio Saggion and Ariadne Carvalho*
- 95-24 **ATIFS: Um Ambiente de Testes baseado em Inje,c ao de Falhas por Software**, *Eliane Martins*
- 95-25 **Multiware Plataform: Some Issues About the Middleware Layer**, *Edmundo Roberto Mauro Madeira*
- 95-26 **WorkFlow Systems: a few definitions and a few suggestions**, *Paulo Barthelmess and Jacques Wainer*
- 95-27 **Workflow Modeling**, *Paulo Barthelmess and Jacques Wainer*

## Relatórios Técnicos – 1996

- 96-01 **Construção de Interfaces Homem-Computador: Uma Proposta Revisada de Disciplina de Graduação**, *Fábio Nogueira Lucena and Hans K.E. Liesenberg*
- 96Abs **DCC-IMECC-UNICAMP Technical Reports 1992–1996 Abstracts**, *C. L. Lucchesi and P. J. de Rezende and J. Stolfi*
- 96-02 **Automatic visualization of two-dimensional cellular complexes**, *Rober Marccone Rosi and Jorge Stolfi*
- 96-03 **Cartas Náuticas Eletrônicas: Operações e Estruturas de Dados**, *Cleomar M. Marques de Oliveira e Neucimar J. Leite*
- 96-04 **On the edge-colouring of split graphs**, *Celina M. H. de Figueiredo, João Meidanis and Célia Picinin de Mello*
- 96-05 **Estudo Comparativo de Métodos para Avaliação de Interfaces Homem-Computador**, *S'lvio Chan e Heloisa Vieira da Rocha*
- 96-06 **User Interface Issues in Geographic Information Systems**, *Juliano Lopes de Oliveira and Claudia Bauzer Medeiros*
- 96-07 **Conjunto fonte máximo em grafos de comparabilidade**, *Marcos Fernando Andrielli e Célia Picinin de Mello*
- 96-08 **96-08 The Effectiveness of Multi-Level Policing Mechanisms in ATM Traffic Control**, *J.A. Silvester, N. L. S. Fonseca, G. S. Mayor e S. P. S. Sobral*
- 96-09 **Sequential and Parallel Experimental Results with Bipartite Matching Algorithms**, *João Carlos Setubal*
- 96-10 **96-10 A CPU for Educational Applications Designed with VHDL and FPGA**, *Nelson V. Augusto, Mario L. Côrtes and Paulo C. Centoducatte*
- 96-11 **Network Design for the Provision of Distributed Home Theatre Services**, *Nelson L. S. Fonseca, Cristiane M. R. Franco, Frank Schaffa*
- 96-12 **Modelling the Output Process of an ATM Multiplexer with Correlated Priorities**, *Nelson L. S. Fonseca e John A. silvester*
- 96-13 **Algoritmos de afinamento tridimensional: exemplos de técnicas de otimização**, *F. N. Bezerra and N. J. Leite*
- 96-14 **Ensino de Estruturas de Dados e seus Algoritmos através de Implementação com Animações**, *Pedro J. de Rezende e Islene C. Garcia*
- 96-15 **Sinergia em Desenho de Grafos Usando Springs e Pequenas Heurísticas**, *H. A. D. do Nascimento, C. F. X. de Mendonça N., P. S. de Souza*

- 96-16 **A Temporal Extension to the Parsimonious Covering Theory**, *Jacques Wainer and Alexandre de Melo Rezende*
- 96-17 **Workcase-centric workflow model**, *Jacques Wainer and Paulo Barthelmess*
- 96-18 **Integrating heuristics and spatial databases: a case study**, *Cid Carvalho de Souza, Claudia Bauzer Medeiros, Ricardo S. Pereira*

*Instituto de Computação*  
*Universidade Estadual de Campinas*  
*13081-970 – Campinas – SP*  
*BRASIL*  
`reltec@dcc.unicamp.br`