Geometria Computacional

Cristina G. Fernandes

Departamento de Ciência da Computação do IME-USP

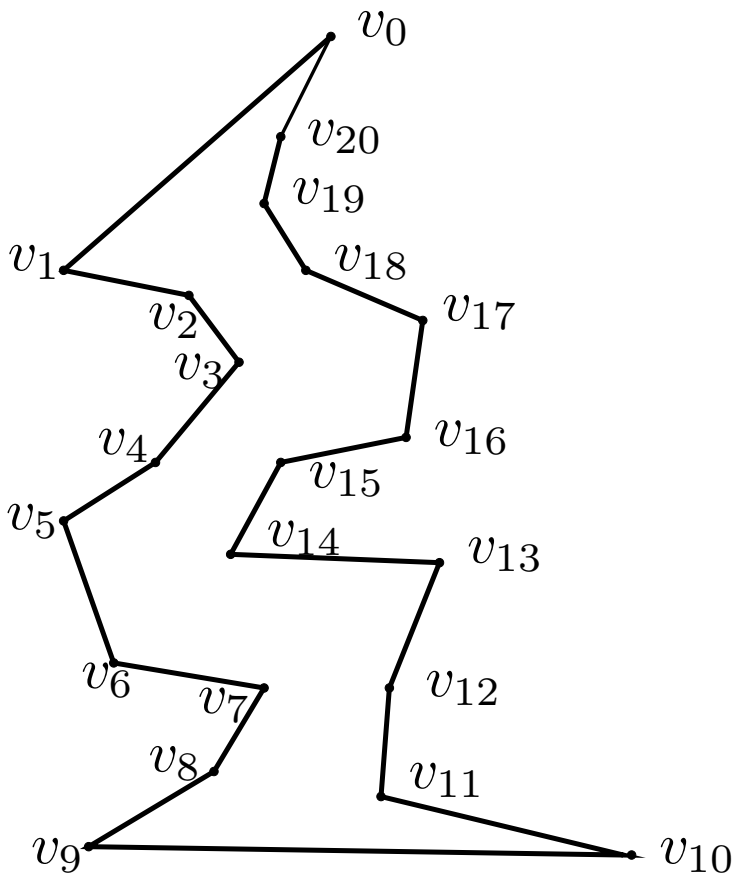
http://www.ime.usp.br/~cris/

segundo semestre de 2018

Polígonos monótonos

Um polígono P é monótono em relação a uma reta L se $P \cap L'$ é conexo para toda reta L' perpendicular a L.

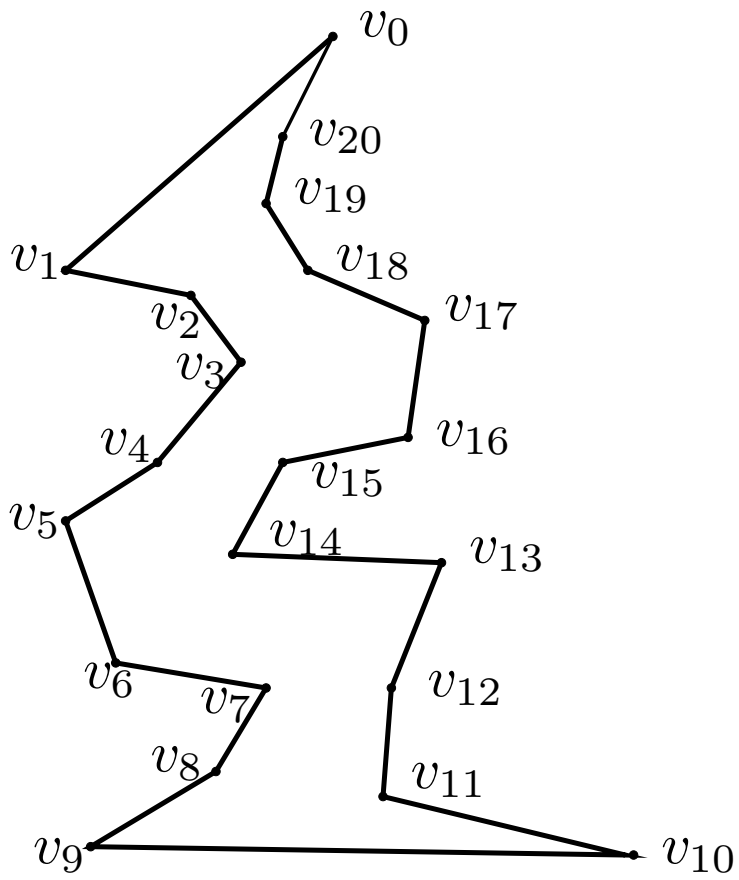
Se L é o eixo y, dizemos que P é y-monótono.



Polígonos monótonos

Um polígono P é monótono em relação a uma reta L se $P \cap L'$ é conexo para toda reta L' perpendicular a L.

Se L é o eixo y, dizemos que P é y-monótono.



Sabemos triangular P em tempo linear.

Triangulação em $\mathrm{O}(n \lg n)$

P: polígono arbitrário com n vértices

Idéia do algoritmo:

Triangulação em $\mathrm{O}(n \lg n)$

P: polígono arbitrário com n vértices

Idéia do algoritmo:

- particionar P em polígonos monótonos
- triangular cada um deles em tempo linear

Triangulação em $\mathrm{O}(n \lg n)$

P: polígono arbitrário com n vértices

Idéia do algoritmo:

- particionar P em polígonos monótonos
- triangular cada um deles em tempo linear

Partição tem que consumir tempo $\mathrm{O}(n \lg n)$!

Triangulação em $\mathrm{O}(n \lg n)$

P: polígono arbitrário com n vértices

Idéia do algoritmo:

- particionar P em polígonos monótonos
- triangular cada um deles em tempo linear

Partição tem que consumir tempo $\mathrm{O}(n \lg n)$!

Como fazemos isso?

Triangulação em $\mathrm{O}(n \lg n)$

P: polígono arbitrário com n vértices

Idéia do algoritmo:

- particionar P em polígonos monótonos
- triangular cada um deles em tempo linear

Partição tem que consumir tempo $\mathrm{O}(n \lg n)$!

Como fazemos isso?

Usando uma trapezoidação especial de P.

Trapezoidação

Trapézio: quadrilátero com duas arestas paralelas

Trapezoidação

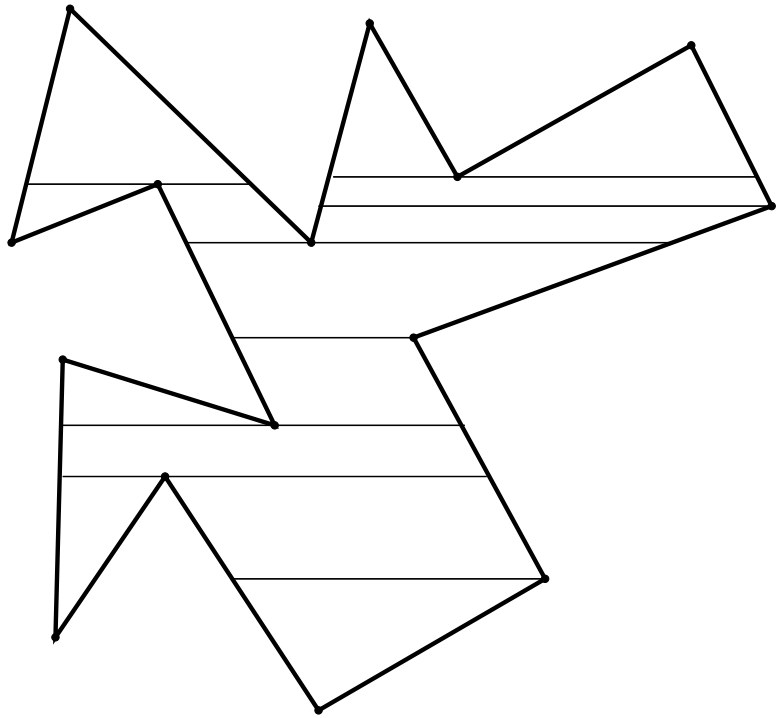
Trapézio: quadrilátero com duas arestas paralelas

Trapezoidação horizontal de um polígono P:
resultado de traçar segmentos horizontais maximais contidos em P, passando por cada vértice de P.

Trapezoidação

Trapézio: quadrilátero com duas arestas paralelas

Trapezoidação horizontal de um polígono P:
resultado de traçar segmentos horizontais maximais contidos em P, passando por cada vértice de P.



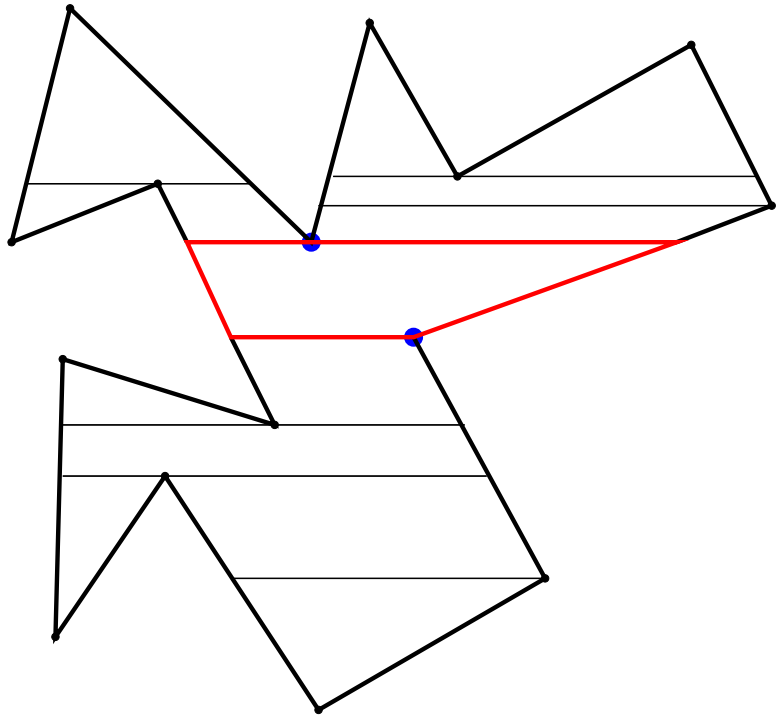
Trapezoidação

Hipótese simplificadora:
não há dois vértices com a mesma Y-coordenada.

Trapezoidação

Hipótese simplificadora:
não há dois vértices com a mesma Y-coordenada.

Afirmação: todo trapézio tem exatamente dois vértices de P em sua fronteira (vértices de suporte), um na aresta superior, outro na inferior.



Representação de mapas retilineares

Como acoplar o algoritmo de triangulação de polígonos monótonos ao algoritmo de Lee e Preparata?

Representação de mapas retilineares

Como acoplar o algoritmo de triangulação de polígonos monótonos ao algoritmo de Lee e Preparata?

Doubly-connected edge list – DCEL

Cada aresta aparece duas vezes nesta ED: uma vez em cada direção, uma cópia apontando para a outra (*twins* ou *half-edges*).

Representação de mapas retilineares

Como acoplar o algoritmo de triangulação de polígonos monótonos ao algoritmo de Lee e Preparata?

Doubly-connected edge list – DCEL

Cada aresta aparece duas vezes nesta ED: uma vez em cada direção, uma cópia apontando para a outra (*twins* ou *half-edges*).

Cada meia-aresta está em uma face do mapa, e aponta para a próxima meia-aresta e a anterior na face.

Representação de mapas retilineares

Como acoplar o algoritmo de triangulação de polígonos monótonos ao algoritmo de Lee e Preparata?

Doubly-connected edge list – DCEL

Cada aresta aparece duas vezes nesta ED: uma vez em cada direção, uma cópia apontando para a outra (*twins* ou *half-edges*).

Cada meia-aresta está em uma face do mapa, e aponta para a próxima meia-aresta e a anterior na face.

Pode se manter registro para cada face ou vértice do mapa. Cada face/vértice tem apontador para uma das (meia-)arestas incidentes a ele.

Doubly-connected edge list – DCEL

Cada aresta $e = uv$ tem duas entrada na ED: (u, v) e (v, u).

Doubly-connected edge list – DCEL

Cada aresta $e = uv$ tem duas entrada na ED: (u, v) e (v, u).

Para cada meia-aresta (u, v):

- $\texttt{twin}(u, v) = (v, u)$
- $\texttt{prox}(u, v)$: próxima meia-aresta na face
- $\texttt{prev}(u, v)$: meia-aresta anterior na face

Doubly-connected edge list – DCEL

Cada aresta $e = uv$ tem duas entrada na ED: (u, v) e (v, u).

Para cada meia-aresta (u, v):

- $\texttt{twin}(u, v) = (v, u)$
- $\texttt{prox}(u, v)$: próxima meia-aresta na face
- $\texttt{prev}(u, v)$: meia-aresta anterior na face

Para cada face f, apontador para uma meia-aresta da face.

Doubly-connected edge list – DCEL

Cada aresta $e = uv$ tem duas entrada na ED: (u, v) e (v, u).

Para cada meia-aresta (u, v):
- $\texttt{twin}(u, v) = (v, u)$
- $\texttt{prox}(u, v)$: próxima meia-aresta na face
- $\texttt{prev}(u, v)$: meia-aresta anterior na face

Para cada face f, apontador para uma meia-aresta da face.

Revisite a primeira fase do algoritmo de Lee e Preparata (partição em monótonos) e a adapte para devolver essa representação da partição obtida em vez do conjunto de diagonais apenas.

Doubly-connected edge list – DCEL

Cada aresta $e = uv$ tem duas entrada na ED: (u, v) e (v, u).

Para cada meia-aresta (u, v):
- $\texttt{twin}(u, v) = (v, u)$
- $\texttt{prox}(u, v)$: próxima meia-aresta na face
- $\texttt{prev}(u, v)$: meia-aresta anterior na face

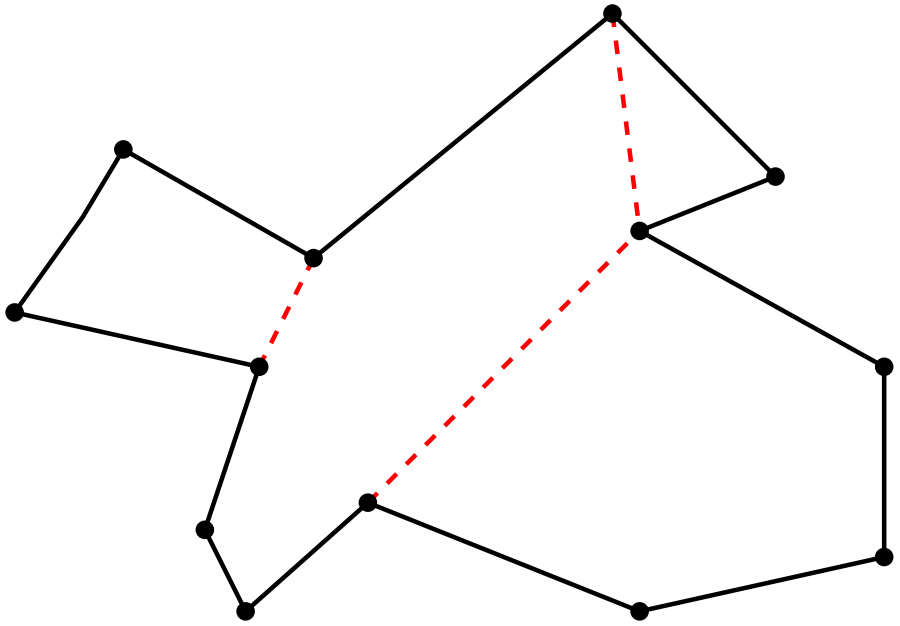
Para cada face f, apontador para uma meia-aresta da face.

Revisite a primeira fase do algoritmo de Lee e Preparata (partição em monótonos) e a adapte para devolver essa representação da partição obtida em vez do conjunto de diagonais apenas.

Omita a face externa da representação,
já que ela não é usada na segunda fase.

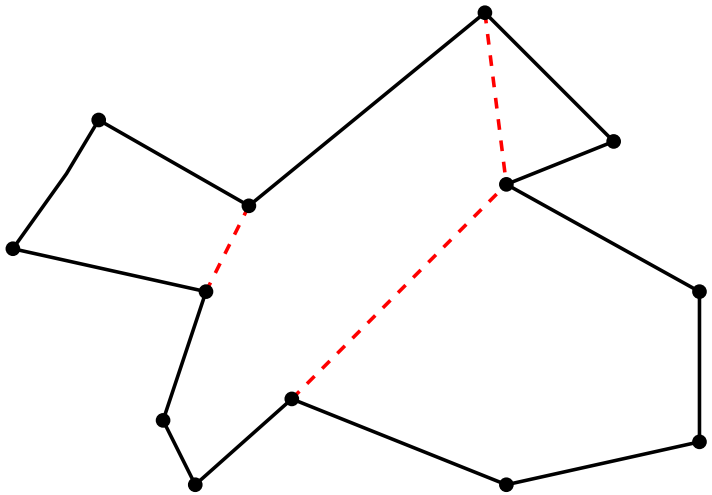
Partição em polígonos convexos

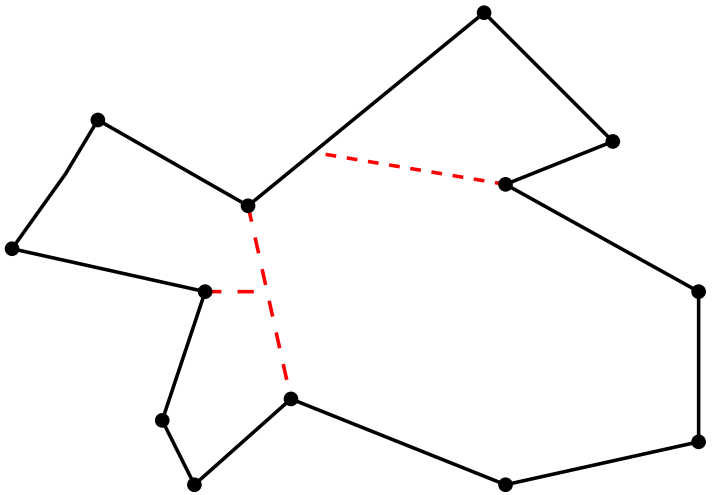
Problema: Dado um polígono P, determinar uma partição de P em um número mínimo de partes convexas.



Partição em polígonos convexos

Problema: Dado um polígono P, determinar uma partição de P em um número mínimo de partes convexas.





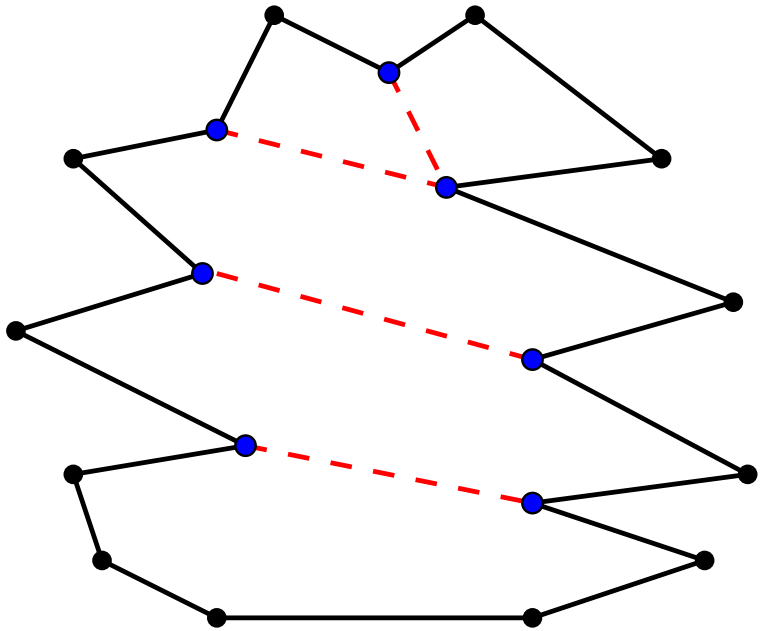
Dois tipos de partição:

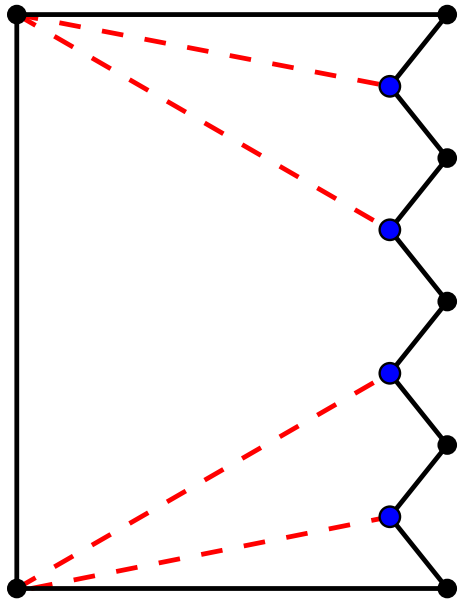
- por diagonais de P
- por segmentos arbitrários contidos em P

Partição em polígonos convexos

Teorema: Seja Φ o menor número de partes convexas em que um polígono com r vértices reflexos pode ser particionado. Então

$$\lceil r/2 \rceil + 1 \leq \Phi \leq r + 1.$$

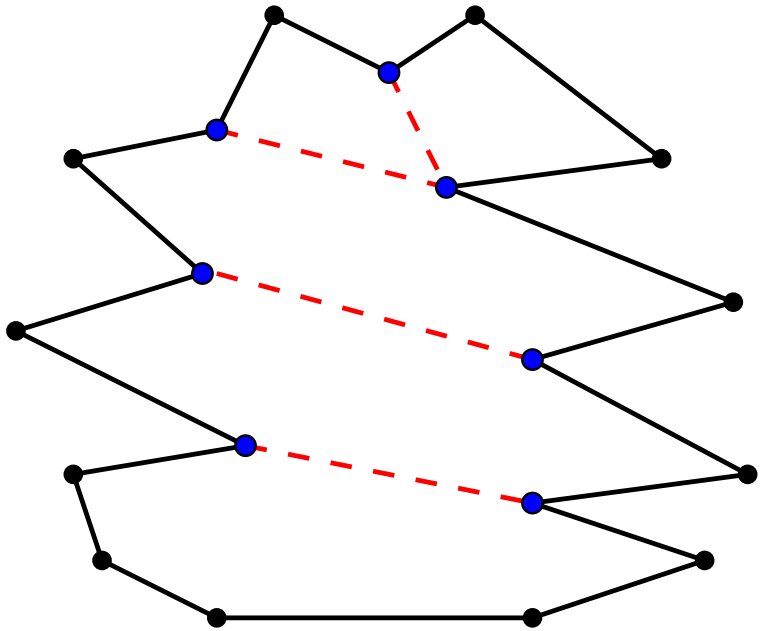


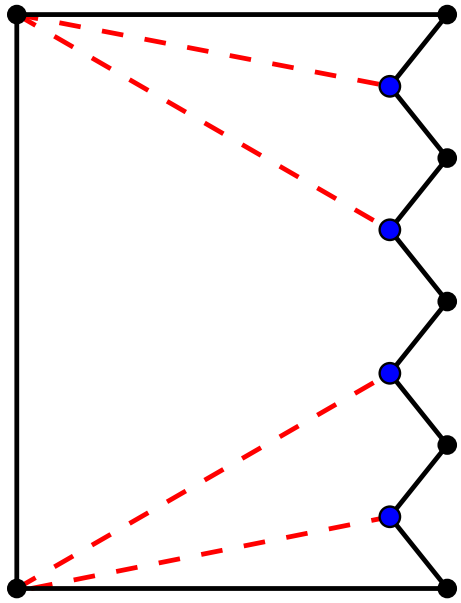


Partição em polígonos convexos

Teorema: Seja Φ o menor número de partes convexas em que um polígono com r vértices reflexos pode ser particionado. Então

$$\lceil r/2 \rceil + 1 \leq \Phi \leq r + 1.$$





Prova: Feita na aula.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Seja Φ^* o número mínimo de diagonais que resulta em uma partição convexa de P.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Seja Φ^* o número mínimo de diagonais que resulta em uma partição convexa de P.

Algoritmo de aproximação: produz uma partição convexa de P por diagonais com no máximo $\alpha\Phi^*$ diagonais, e consome tempo polinomial.

Um tal algoritmo é chamado de α-aproximação, e α é sua razão de aproximação.

Algoritmo de Hertel e Mehlhorn

Problema: Dado P, determinar uma partição de P por diagonais, em um número mínimo de partes convexas.

Seja Φ^* o número mínimo de diagonais que resulta em uma partição convexa de P.

Algoritmo de aproximação: produz uma partição convexa de P por diagonais com no máximo $\alpha\Phi^*$ diagonais, e consome tempo polinomial.

Um tal algoritmo é chamado de α-aproximação, e α é sua razão de aproximação.

O algoritmo de Hertel e Mehlhorn produz uma partição de P por diagonais, sempre com no máximo $4\Phi^*$ diagonais, ou seja, é uma 4-aproximação.

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Consumo de tempo: linear, usando o algoritmo
de triangulação de Chazelle (que não vimos).

Algoritmo de Hertel e Mehlhorn

Considere uma partição convexa de P por diagonais.

Vértice reflexo: vértice com ângulo interno maior que π.

Uma diagonal d é essencial para um vértice v se
a remoção de d torna v reflexo (e a partição deixa de ser convexa).

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Consumo de tempo: linear, usando o algoritmo
de triangulação de Chazelle (que não vimos).

Por que ele é uma 4-aproximação?

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem no máximo duas diagonais essenciais por vértice reflexo.

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem no máximo duas diagonais essenciais por vértice reflexo.

Prova: Feita na aula.

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem
no máximo duas diagonais essenciais por vértice reflexo.

Prova: Feita na aula.

Teorema: O algoritmo de Hertel e Mehlhorn é uma 4-aproximação.

Algoritmo de Hertel e Mehlhorn

Algoritmo de Hertel e Mehlhorn:
Construa uma triangulação de P e
remova, enquanto houver, diagonais não-essenciais.
Devolva a partição que resultar disso.

Por que ele é uma 4-aproximação?

Lema: Em uma partição convexa por diagonais, existem
no máximo duas diagonais essenciais por vértice reflexo.

Prova: Feita na aula.

Teorema: O algoritmo de Hertel e Mehlhorn é uma 4-aproximação.

Prova: Feita na aula.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n^2) = O(n^4)$, onde n é o número de vértices e r é o número de vértices reflexos do polígono.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n^2) = O(n^4)$, onde n é o número de vértices e r é o número de vértices reflexos do polígono.

Posteriormente, Keil propôs um outro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n \lg n)$.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n^2) = O(n^4)$, onde n é o número de vértices e r é o número de vértices reflexos do polígono.

Posteriormente, Keil propôs um outro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n \lg n)$.

Os dois algoritmos são de programação dinâmica.

Partição convexa ótima

Greene apresentou o primeiro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n^2) = O(n^4)$, onde n é o número de vértices e r é o número de vértices reflexos do polígono.

Posteriormente, Keil propôs um outro algoritmo que encontra uma partição convexa por diagonais ótima.

Consumo de tempo: $O(r^2 n \lg n)$.

Os dois algoritmos são de programação dinâmica.

Se a partição é por segmentos, o problema fica mais complicado. Há um algoritmo de Chazelle que consome tempo $O(n + r^3) = O(n^3)$ para este caso.