

# Counting Convex 5-Holes

Bachelorarbeit Informatik

Manfred Scheucher

1. August 2013

# Aufgabenstellung

## gegeben

Die Anzahl der Punkte ( $n$ ) für Punktmenge in allgemeiner Lage ( $\mathbb{R}^2$ )

## gesucht

Die minimale Anzahl  $h_5(n)$  leerer konvexer 5-Ecke ("convex 5-holes") für alle Punktmenge mit  $n$  Punkten.

Bereits bekannt...

$n$	$h_5(n)$
$\leq 9$	$0^*$
10	$1^*$
11	$2^*$
12	$3^*$

# Order Types

- Um Punktmengen zu unterscheiden
- Orientierung von je 3 Punkten  
(keine geometrische Information)
- Orientierung  $\Lambda_{ijk} \in \{-1, +1\}$
- $\Lambda$ -Matrix ( $\Theta(n^3)$  Speicher)
- $\lambda_{ij} = \#\text{Punkte links von } p_i - p_j$
- $\lambda$ -Matrix ( $\Theta(n^2)$  Speicher)

# Order Types

- Um Punktmengen zu unterscheiden
- Orientierung von je 3 Punkten  
(keine geometrische Information)
- Orientierung  $\Lambda_{ijk} \in \{-1, +1\}$
- $\Lambda$ -Matrix ( $\Theta(n^3)$  Speicher)
- $\lambda_{ij} = \#\text{Punkte links von } p_i - p_j$
- $\lambda$ -Matrix ( $\Theta(n^2)$  Speicher)

# Order Types

- Um Punktmengen zu unterscheiden
- Orientierung von je 3 Punkten  
(keine geometrische Information)
- Orientierung  $\Lambda_{ijk} \in \{-1, +1\}$
- $\Lambda$ -Matrix ( $\Theta(n^3)$  Speicher)
- $\lambda_{ij} = \#\text{Punkte links von } p_i - p_j$
- $\lambda$ -Matrix ( $\Theta(n^2)$  Speicher)

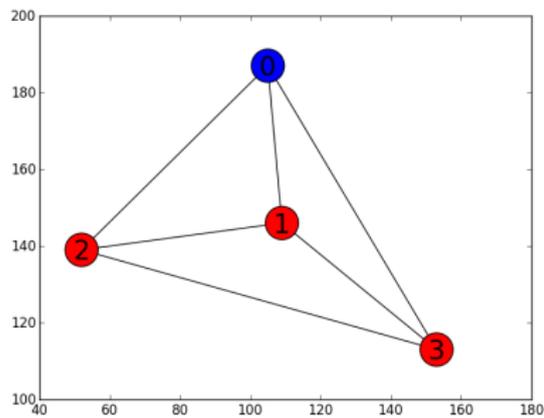
# Order Types

- Um Punktmengen zu unterscheiden
- Orientierung von je 3 Punkten  
(keine geometrische Information)
- Orientierung  $\Lambda_{ijk} \in \{-1, +1\}$
- $\Lambda$ -Matrix ( $\Theta(n^3)$  Speicher)
- $\lambda_{ij} = \#\text{Punkte links von } p_i - p_j$
- $\lambda$ -Matrix ( $\Theta(n^2)$  Speicher)

# Order Types

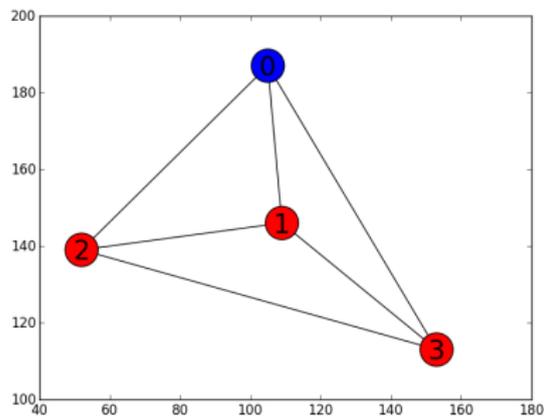
- Um Punktmengen zu unterscheiden
- Orientierung von je 3 Punkten  
(keine geometrische Information)
- Orientierung  $\Lambda_{ijk} \in \{-1, +1\}$
- $\Lambda$ -Matrix ( $\Theta(n^3)$  Speicher)
- $\lambda_{ij} = \#\text{Punkte links von } p_i - p_j$
- $\lambda$ -Matrix ( $\Theta(n^2)$  Speicher)

# Order Types

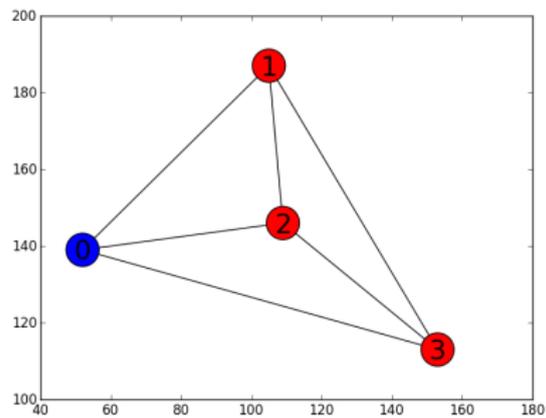


0	1	2	0
1	0	1	1
0	1	0	2
2	1	0	0

# Order Types



0	1	2	0
1	0	1	1
0	1	0	2
2	1	0	0



0	0	1	2
2	0	1	0
1	1	0	1
0	2	1	0

# Order Types

- Lexikographisches Minimum  $\lambda_{min}$
- Berechnung:  $\Theta(n^3)$  Zeit
- Realisierung: NP-schwer

# Order Types

- Lexikographisches Minimum  $\lambda_{min}$
- Berechnung:  $\Theta(n^3)$  Zeit
- Realisierung: NP-schwer

# Abstrakte Erweiterung

- Um alle Punktmengen mit  $n$  Punkten aufzuzählen
- Punkt einfügen  $\leftrightarrow$  Punkt entfernen  
 $\Rightarrow$  OT mit  $n$  Punkten hat  $n$  Vorgänger OTs
- $\exists$  Punkt auf KH  
 $\Rightarrow \exists$  Vorgänger OT ohne diesen äußeren Punkt <sup>1</sup>  
 $\Rightarrow$  Erweiterung im Äußeren ist vollständig

---

<sup>1</sup>Punkt auf KH entfernen  $\leftrightarrow$  Einfügen im Äußeren

# Abstrakte Erweiterung

- Um alle Punktfolgen mit  $n$  Punkten aufzuzählen
- Punkt einfügen  $\leftrightarrow$  Punkt entfernen  
 $\Rightarrow$  OT mit  $n$  Punkten hat  $n$  Vorgänger OTs
- $\exists$  Punkt auf KH  
 $\Rightarrow \exists$  Vorgänger OT ohne diesen äußeren Punkt <sup>1</sup>  
 $\Rightarrow$  Erweiterung im Äußeren ist vollständig

---

<sup>1</sup>Punkt auf KH entfernen  $\leftrightarrow$  Einfügen im Äußeren

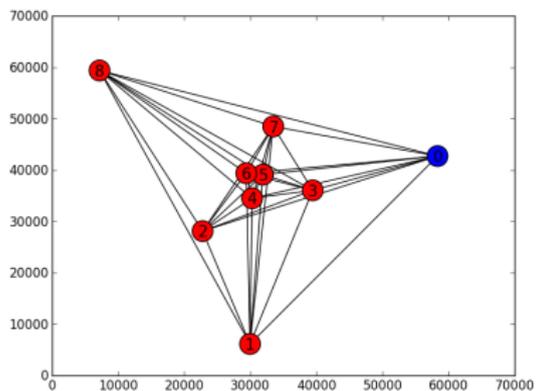
# Abstrakte Erweiterung

- Um alle Punktfolgen mit  $n$  Punkten aufzuzählen
- Punkt einfügen  $\leftrightarrow$  Punkt entfernen  
 $\Rightarrow$  OT mit  $n$  Punkten hat  $n$  Vorgänger OTs
- $\exists$  Punkt auf KH  
 $\Rightarrow \exists$  Vorgänger OT ohne diesen äußeren Punkt <sup>1</sup>  
 $\Rightarrow$  Erweiterung im Äußeren ist vollständig

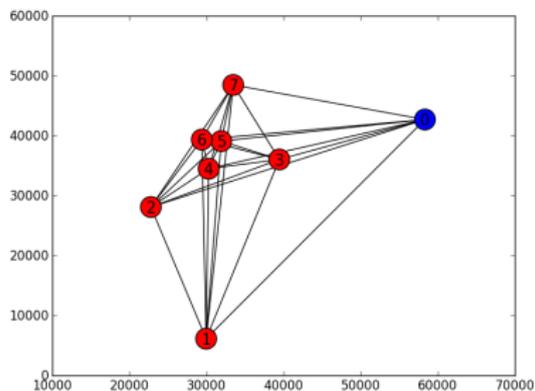
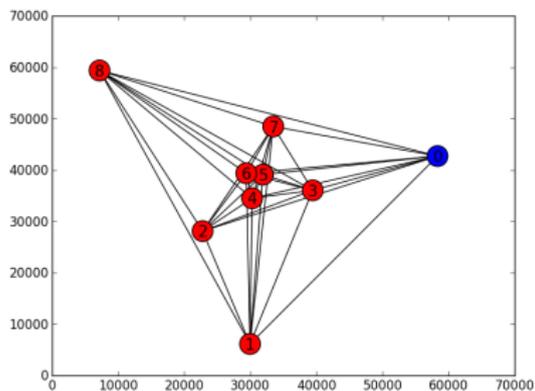
---

<sup>1</sup>Punkt auf KH entfernen  $\leftrightarrow$  Einfügen im Äußeren

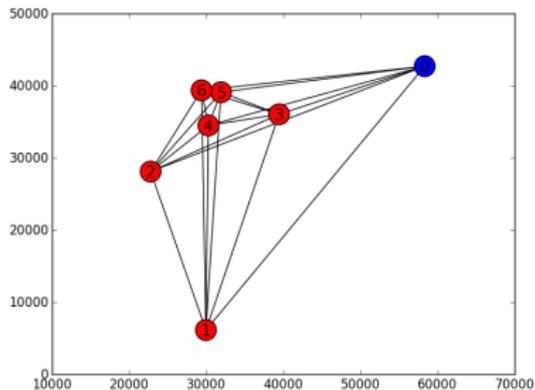
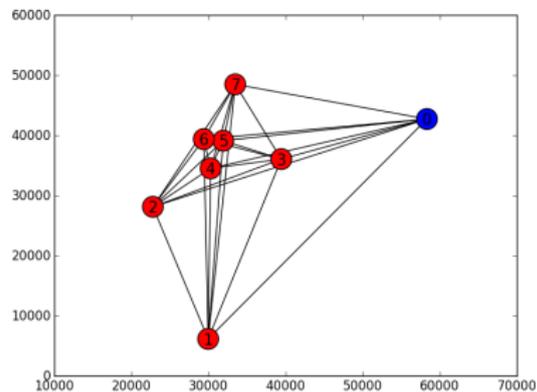
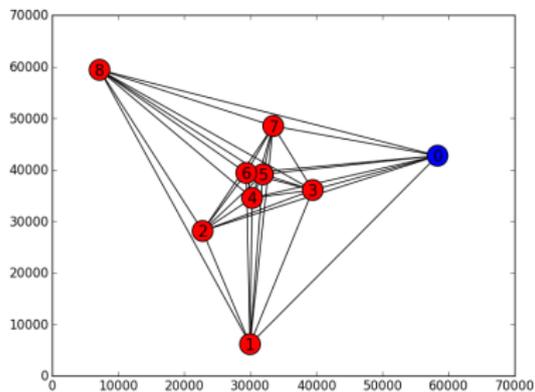
# Abstrakte Erweiterung



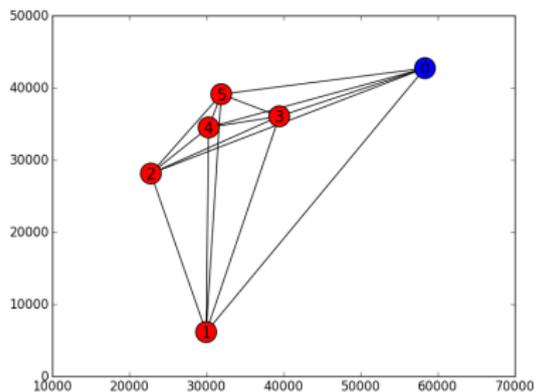
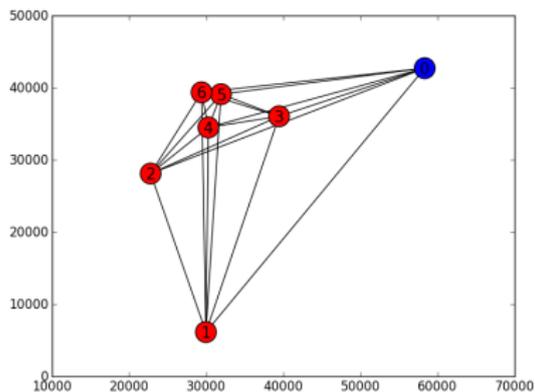
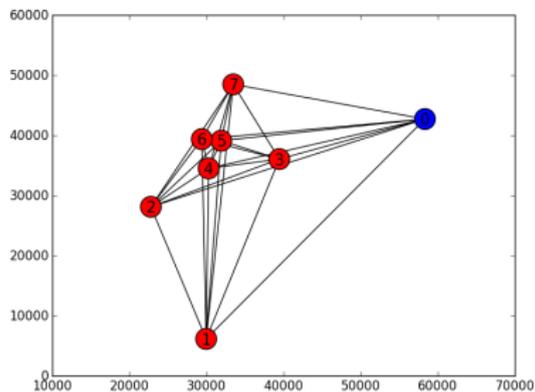
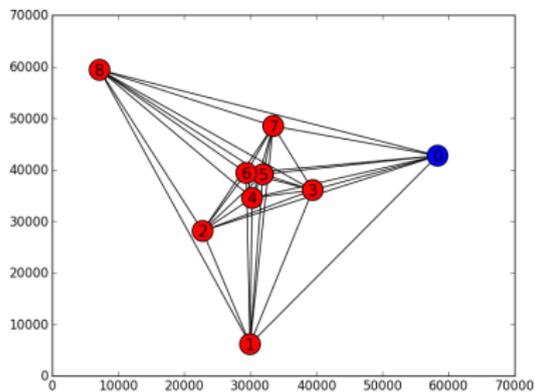
# Abstrakte Erweiterung



# Abstrakte Erweiterung



# Abstrakte Erweiterung



# Abstrakte Erweiterung

- OBdA: der zuletzt eingefügte Punkt (im Äußeren) hat Index  $n^2$
- $n$  Punkte
  - ⇒  $\leq 2n$  natürliche Labelings (gespiegelt)
  - ⇒ Nur lexikographisches Minimum ausgeben<sup>3</sup>
  - ⇒ Völlständig und keine Duplikate!
- Realisierbarkeit? ⇒ abstract Order Types

---

<sup>2</sup>OT ⇒ lex.min. Labeling ⇒ natürliches Labeling ⇒  $n$ -ter Punkt auf KH

<sup>3</sup>lex.min. Labeling ⇒ natürliches Labeling

# Abstrakte Erweiterung

- OBdA: der zuletzt eingefügte Punkt (im Äußeren) hat Index  $n^2$
- $n$  Punkte
  - ⇒  $\leq 2n$  natürliche Labelings (gespiegelt)
  - ⇒ Nur lexikographisches Minimum ausgeben <sup>3</sup>
  - ⇒ **Vollständig und keine Duplikate!**
- Realisierbarkeit? ⇒ abstract Order Types

---

<sup>2</sup>OT ⇒ lex.min. Labeling ⇒ natürliches Labeling ⇒  $n$ -ter Punkt auf KH

<sup>3</sup>lex.min. Labeling ⇒ natürliches Labeling

# Abstrakte Erweiterung

- OBdA: der zuletzt eingefügte Punkt (im Äußeren) hat Index  $n^2$
- $n$  Punkte
  - ⇒  $\leq 2n$  natürliche Labelings (gespiegelt)
  - ⇒ Nur lexikographisches Minimum ausgeben <sup>3</sup>
  - ⇒ Völlständig und keine Duplikate!
- Realisierbarkeit? ⇒ **abstract** Order Types

---

<sup>2</sup>OT ⇒ lex.min. Labeling ⇒ natürliches Labeling ⇒  $n$ -ter Punkt auf KH

<sup>3</sup>lex.min. Labeling ⇒ natürliches Labeling

# Zählen von Convex 5-Holes

Grundidee:

- Erweiterung im Äußeren
- $\Rightarrow$  Keine alten  $c5h^4$  zerstört
- $\Rightarrow$  Neue  $c5h$  nur beim zuletzt eingefügten Punkt

---

<sup>4</sup>altes  $c5h$  ...  $c5h$  vor Erweiterung (im Vorgänger OT)

# Zählen von Convex 5-Holes

Grundidee:

- Erweiterung im Äußeren
- $\Rightarrow$  Keine alten  $c5h^4$  zerstört
- $\Rightarrow$  Neue  $c5h$  nur beim zuletzt eingefügten Punkt

---

<sup>4</sup>altes  $c5h$  ...  $c5h$  vor Erweiterung (im Vorgänger OT)

# Zählen von Convex 5-Holes

Vorgehensweise:

- $n \xrightarrow{\text{erw.}} n + 1 \xrightarrow{\text{erw.}} \dots \xrightarrow{\text{erw.}} n + k$
- Zählen in/nach jedem Schritt
- **Nur neue c5h Zählen**
- Nur relevante (a)OTs erweitern!

# Zählen von Convex 5-Holes

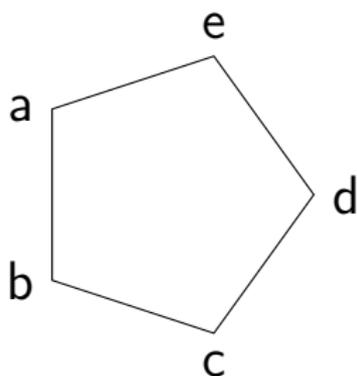
Vorgehensweise:

- $n \xrightarrow{\text{erw.}} n + 1 \xrightarrow{\text{erw.}} \dots \xrightarrow{\text{erw.}} n + k$
- Zählen in/nach jedem Schritt
- Nur neue c5h Zählen
- Nur relevante (a)OTs erweitern!

# Convex 5-Gon: Definition

## Konvexes 5-Eck (convex 5-gon)

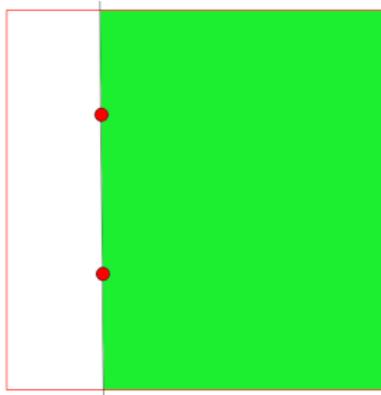
Punkte  $a, b, c, d, e$  bilden konvexes 5-Eck, wenn  $c, d, e$  links von der gerichteten Geraden  $a - b$ <sup>1</sup> (und analog für  $b - c, c - d, d - e$  und  $e - a$ ).



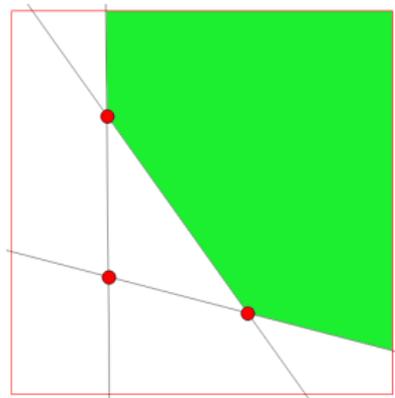
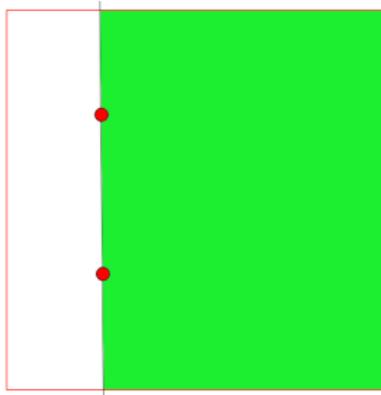
---

<sup>1</sup>Orientierungen  $\Lambda_{abc}, \Lambda_{abd}, \Lambda_{abe}, \dots$  aus  $\Lambda$ -Matrix (ohne geometrische Informationen!)

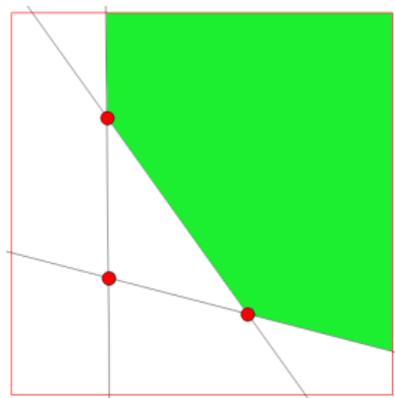
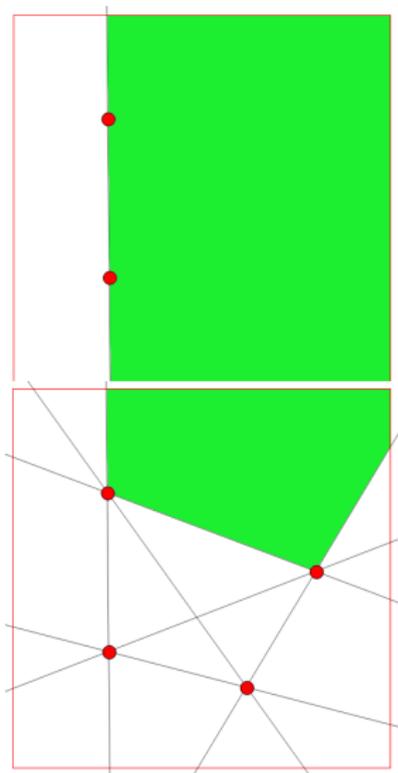
# Convex 5-Gon: Zählmethode



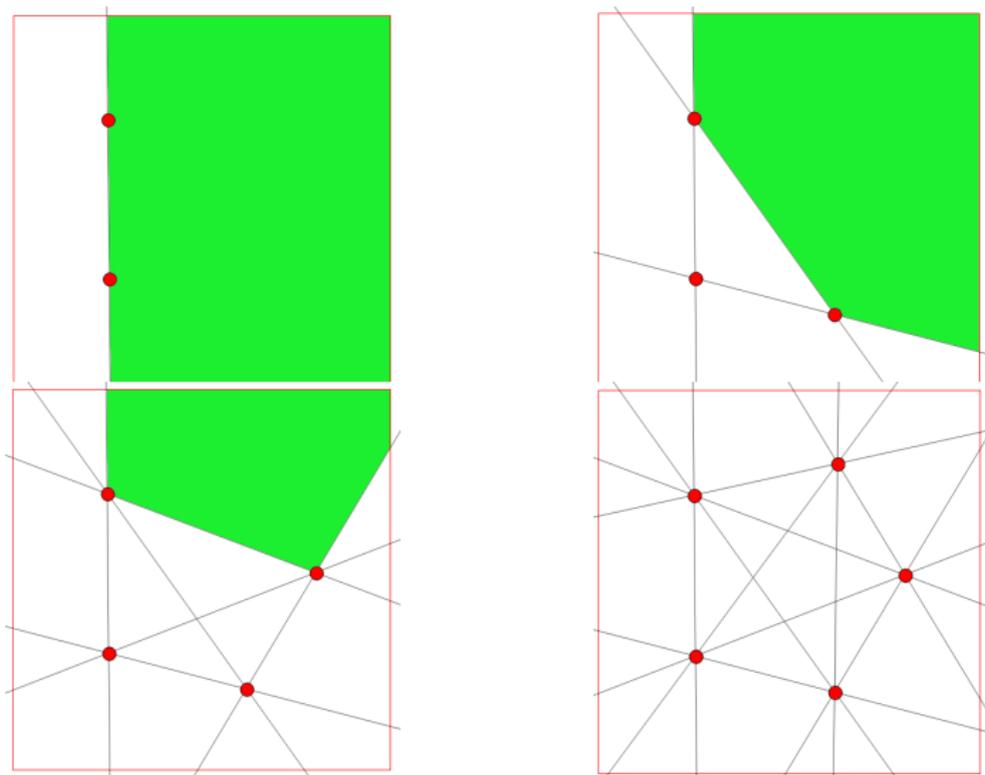
# Convex 5-Gon: Zählmethode



# Convex 5-Gon: Zählmethode



# Convex 5-Gon: Zählmethode



# Convex 5-Hole: Definition

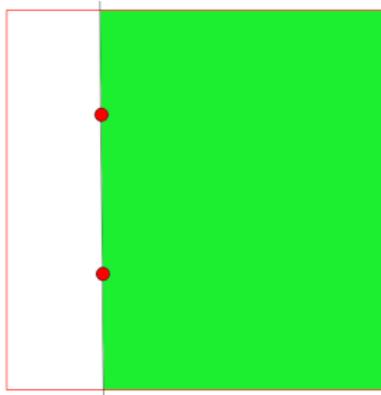
## Leeres, konvexes 5-Eck (convex 5-hole)

Ein konvexes 5-Eck ist leer, wenn kein Punkt im Inneren<sup>1</sup> liegt.

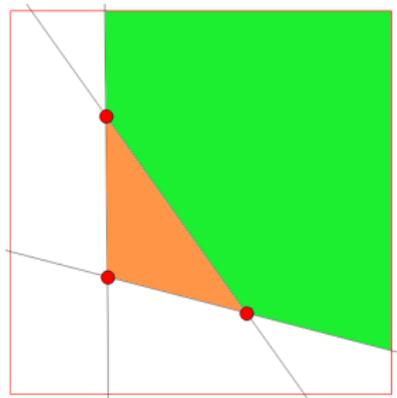
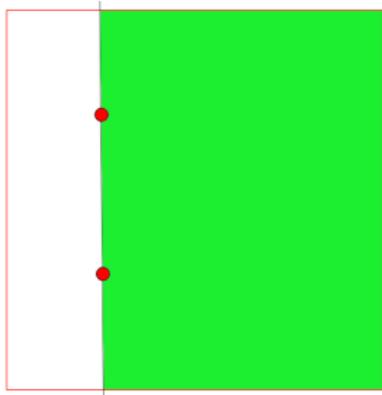
---

<sup>1</sup> $p$  liegt im Inneren, wenn  $p$  links von den Geraden  $a - b$ ,  $b - c$ ,  $c - d$ ,  $d - e$  und  $e - a$  liegt.

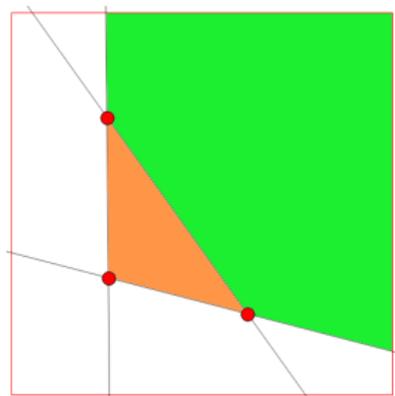
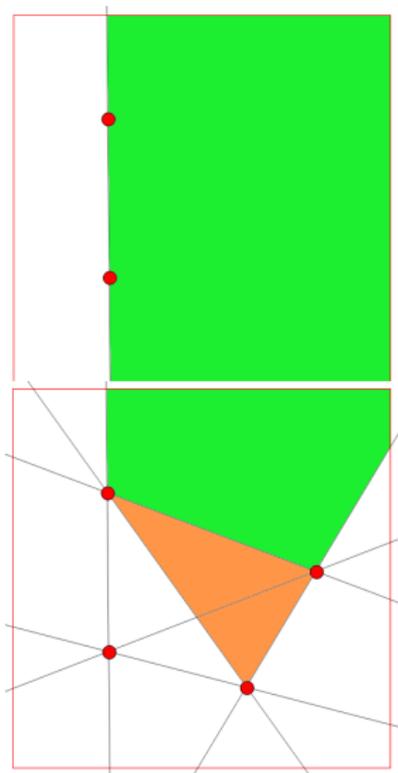
# Convex 5-Hole: Zählmethode



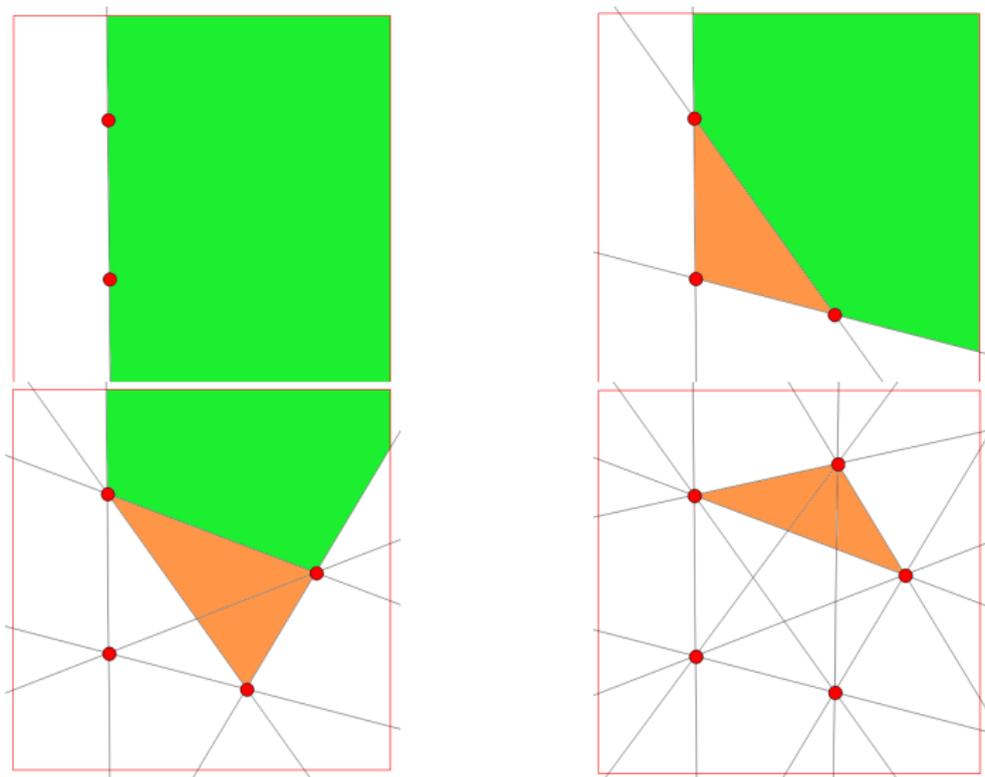
# Convex 5-Hole: Zählmethode



# Convex 5-Hole: Zählmethode



# Convex 5-Hole: Zählmethode



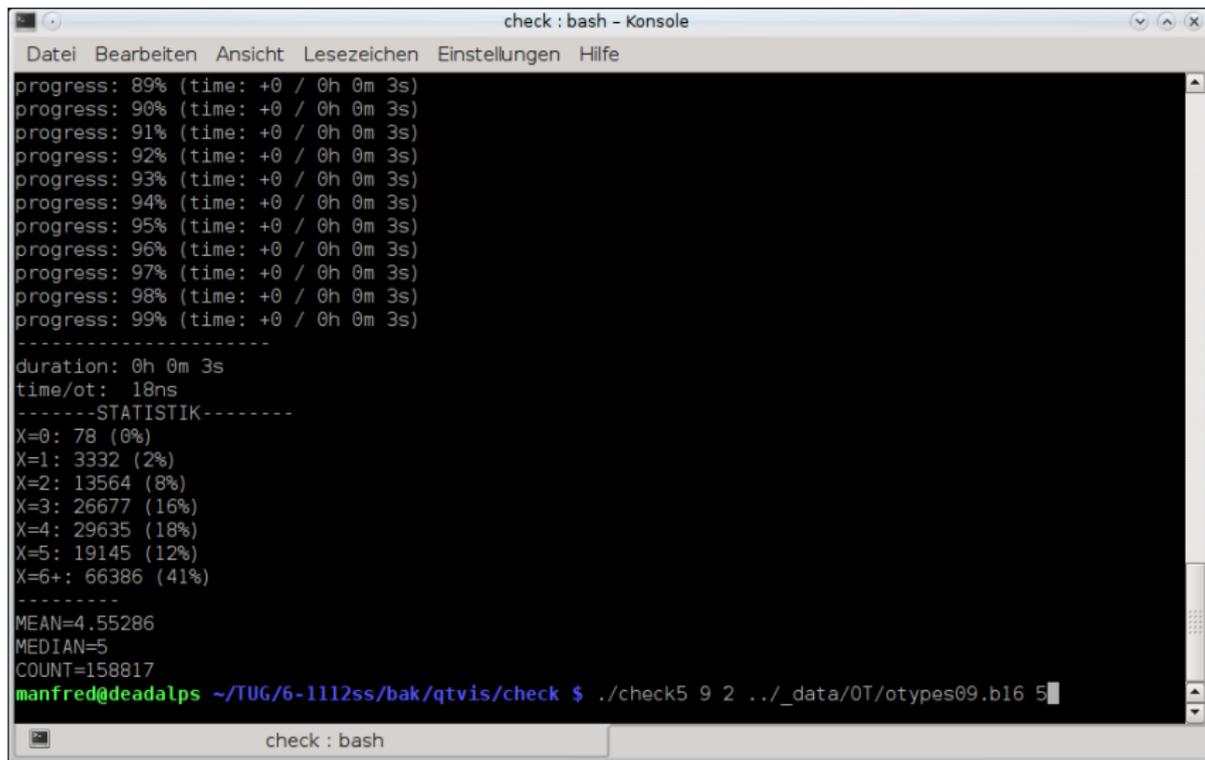
# Tool: check5

- Zum Zählen von c5h
- Einfache Zählmethode + Merged Cape  
⇒  $h_5(13) = 3$ 
  - ▶ Input: ~50GB
  - ▶ Output: 168(a)OTs (~60KB)
  - ▶ langsam/ineffizient!

# Tool: check5

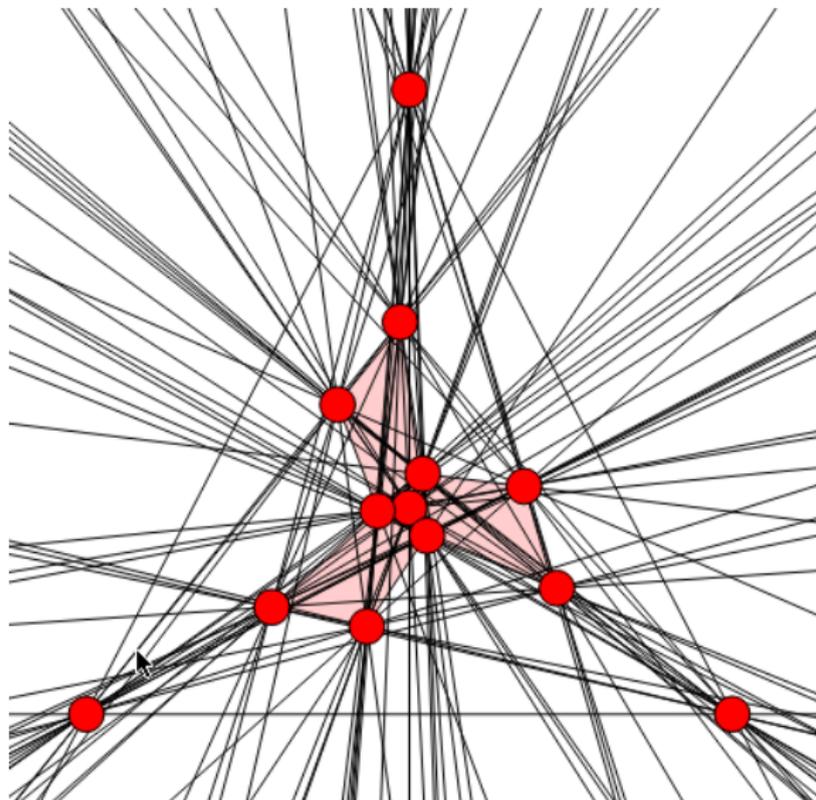
- Zum Zählen von c5h
- Einfache Zählmethode + Merged Cape  
⇒  $h_5(13) = 3$ 
  - ▶ Input: ~50GB
  - ▶ Output: 168(a)OTs (~60KB)
  - ▶ langsam/ineffizient!

# Tool: check5



```
check : bash - Konsole
Datei Bearbeiten Ansicht Lesezeichen Einstellungen Hilfe
progress: 89% (time: +0 / 0h 0m 3s)
progress: 90% (time: +0 / 0h 0m 3s)
progress: 91% (time: +0 / 0h 0m 3s)
progress: 92% (time: +0 / 0h 0m 3s)
progress: 93% (time: +0 / 0h 0m 3s)
progress: 94% (time: +0 / 0h 0m 3s)
progress: 95% (time: +0 / 0h 0m 3s)
progress: 96% (time: +0 / 0h 0m 3s)
progress: 97% (time: +0 / 0h 0m 3s)
progress: 98% (time: +0 / 0h 0m 3s)
progress: 99% (time: +0 / 0h 0m 3s)
-----
duration: 0h 0m 3s
time/ot: 18ns
-----STATISTIK-----
X=0: 78 (0%)
X=1: 3332 (2%)
X=2: 13564 (8%)
X=3: 26677 (16%)
X=4: 29635 (18%)
X=5: 19145 (12%)
X=6+: 66386 (41%)
-----
MEAN=4.55286
MEDIAN=5
COUNT=158817
manfred@deadalps ~/TUG/6-1112ss/bak/qtvis/check $ ./check5 9 2 ../_data/OT/otypes09.b16 5
```

$n = 13$  Order Type mit 3 Convex 5-Holes



# Merged Cape: Pentagon Functionality

- Zählen **während** dem Erweitern  
⇒ deutlich weniger Output
- Alte c5h vom Vorgänger übernehmen  
⇒ Nur neue c5h zu zählen  
⇒  $O(n^4)$  anstatt von  $O(n^5)$  Zeit
- ⇒  $h_5(14) = 6$

# Merged Cape: Pentagon Functionality

- Zählen während dem Erweitern  
⇒ deutlich weniger Output
- Alte c5h vom Vorgänger übernehmen  
⇒ Nur neue c5h zu zählen  
⇒  $O(n^4)$  anstatt von  $O(n^5)$  Zeit
- ⇒  $h_5(14) = 6$

# Merged Cape: Pentagon Functionality

- Zählen während dem Erweitern  
⇒ deutlich weniger Output
- Alte c5h vom Vorgänger übernehmen  
⇒ Nur neue c5h zu zählen  
⇒  $O(n^4)$  anstatt von  $O(n^5)$  Zeit
- ⇒  $h_5(14) = 6$

# Tool: paracape

- Einfaches Parallelisierungstool
- $\Rightarrow$  Vermutung:  $h_5(15) = 9$

# Tool: randcape

- Modifizierung von paracape
- Zufälliges Erweitern
- Bessere Minima (nur) bis  $n = 20$
- Problem: isFather-Test (Merged Cape)

# Tool: randcape

- Modifizierung von paracape
- Zufälliges Erweitern
- Bessere Minima (nur) bis  $n = 20$
- Problem: isFather-Test (Merged Cape)

# Tool: check4/check6 (Experimentell)

- Zum Zählen von 4 und 6 Ecken  
(später auch zum Visualisieren)
- split-Feature

# Tool: meta\_randcape

- Automatische Suche
- Zufälliges Erweitern (+Greedy)
- + ignoreFatherTest (Merged Cape)
- $\Rightarrow$  bessere Minima (auch für  $n > 20$ )

# Tool: meta\_randcape

- Automatische Suche
- Zufälliges Erweitern (+Greedy)
- + ignoreFatherTest (Merged Cape)
- $\Rightarrow$  bessere Minima (auch für  $n > 20$ )

# Tool: qtviz

- OT Visualisierungs-/Bearbeitungs-Tool
- Händische Suche
- $\Rightarrow$  bessere Minima für  $n \geq 16$

# Tool: qtvis

The screenshot displays the qtvis software interface, titled "Qtvis - PointEditor". The main window shows a network graph with 13 red nodes and numerous black edges. A red rectangular box highlights a central cluster of nodes. The interface includes several control panels:

- Top Left Panel:**
  - number of points: 13
  - Status: OK
  - #ec5gons: 3
  - #ec6gons: 0
  - NodeSize: [slider]
  - Selected NodeSize: [slider]
  - EdgeSize: [slider]
- Options Panel:**
  - count convex 5 holes
  - color convex 5 holes
  - count convex 6 holes
  - color convex 6 holes
  - fix current OT (restrict point movement)
  - hide edges of current node
- Operations Panel:**
  - centering and scaling:**
    - range of points (relative): 0,800
    - scale only by integer numbers
    - center and scale
  - optimize: move points to center of cell:**
    - number of iterations: 1
    - optimize [Space]
  - find minimum (randomized):**
    - count 6gons instead of 5gons
    - number of trials: 3
    - max iterations (all): 6
    - max fail-iterations: 2
    - alpha: 1,05
    - epsilon: 3
    - only move last point
    - try to minimize [Enter]
- Bottom Panel:** save and continue with next ordertype

# Tool: reduce

- Vorgänger OTs mit  $(n-k)$  Punkten berechnen
- $n = 29$  OT mit keinem convex 6-hole
- reduce + qtviz  $\Rightarrow$  Verbesserung:  $n = 20..36$

# Vollständige Erweiterung auf $n = 15$

- $h_5(15) = 9$
- $h_5(16) > 9$
- OT mit 11 c5h bekannt  $\Rightarrow h_5(16) \in \{10, 11\}$

# Vollständige Erweiterung auf $n = 15$

- $h_5(15) = 9$
- $h_5(16) > 9$
- OT mit 11 c5h bekannt  $\Rightarrow h_5(16) \in \{10, 11\}$

# Vollständige Erweiterung auf $n = 15$

- $h_5(15) = 9$
- $h_5(16) > 9$
- OT mit 11 c5h bekannt  $\Rightarrow h_5(16) \in \{10, 11\}$

# Neue Ergebnisse

$n$	$h_5(n)$
$\leq 9$	$0^*$
10	$1^*$
11	$2^*$
12	$3^*$
13	3
14	6
15	9
16	$\in \{10, 11\}$

---

\* ... bereits bekannt

# Neue Ergebnisse

n	$h_5(n) = \min \#c5h$ (realisierbar)	$\min \#c5h$ (abstrakt)
17	$\leq 16$	
18	$\leq 21$	
19	$\leq 28$	$\leq 27$
20	$\leq 36$	$\leq 34$
21	$\leq 43$	$\leq 40$
22	$\leq 52$	$\leq 48$
23	$\leq 63$	$\leq 55$
24	$\leq 71$	
25	$\leq 82$	
26	$\leq 90$	
27	$\leq 103$	
28	$\leq 117$	
29	$\leq 134$	
30	$\leq 155$	
⋮		
36	$\leq 330$	

Vielen Dank für Ihre Aufmerksamkeit!