Praktikum Graphgeneratoren: Planare Graphen

Marcus Krug & Jochen Speck

Zielsetzung

- Generator für planare Graphen
- Eingabeparameter
  - n Anzahl der Knoten
  - k Anzahl der Kanten
- *Jeder planare Graph wird mit positiver Wahrscheinlichkeit erzeugt.*

Definitionen

Definition

Ein Graph $G = (V, E)$, der eine ebene Zeichnung besitzt, bei der sich keine Kanten überschneiden ist ein planarer Graph.

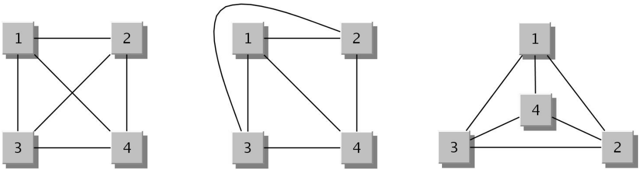


Abbildung: K_4 mit einer nicht ebenen und zwei ebenen Zeichnungen

Eigenschaften planarer Graphen

- $m \leq 3n - 6$
- Rektilineare Zeichnung eines planaren Graphen
  - Jeder planare Graph besitzt eine ebene, überschneidungsfreie Zeichnung, bei der die Kanten von G Strecken sind.
- Planaritätstest in $\Theta(n)$

Naiver Ansatz

Algorithmus

Erzeuge unabhängige Menge mit n Knoten.
while $m < k$ **do**
  Wähle zufällige Knoten u, v *mit* $\{u, v\} \notin E$
  if $G + \{u, v\}$ *planar* **then**
    füge $\{u, v\}$ *zu* G *hinzu*
  end if
end while

- *Aufwand:* $\Omega(kn)$

Grundidee für den Algorithmus

- Verzicht auf “teure” Planaritätstests.
- Erzeuge maximalen planaren Graphen ohne Verwendung von Planaritätstests.
  - Planarer Graph mit n Knoten und $3n - 6$ Kanten.
  - Jeder planare Graph ist Teilgraph eines maximalen planaren Graphen.
- Lösche zufällige Kanten, solange $m > k$.

Illustration: Maximale Planare Graphen

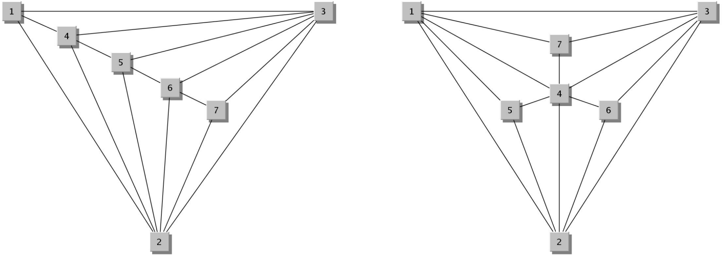


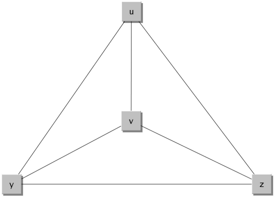
Abbildung: Zwei nicht isomorphe maximale planare Graphen mit 7 Knoten und 15 Kanten.

Erster Ansatz zur Erzeugung eines maximalen planaren Graphen

- Initialisiere Graph G als Dreieck.
- Füge rekursiv die restlichen $n-3$ Knoten wie folgt hinzu:

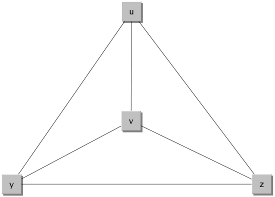
Erster Ansatz zur Erzeugung eines maximalen planaren Graphen

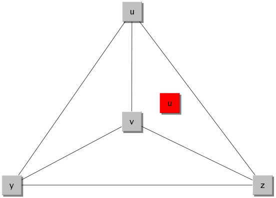
- Initialisiere Graph G als Dreieck.
- Füge rekursiv die restlichen $n - 3$ Knoten wie folgt hinzu:
- Wähle eine zufällige innere Facette F (Dreieck) aus.



Erster Ansatz zur Erzeugung eines maximalen planaren Graphen

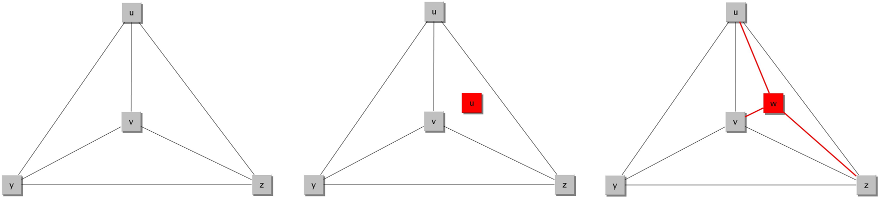
- Initialisiere Graph G als Dreieck.
- Füge rekursiv die restlichen $n - 3$ Knoten wie folgt hinzu:
- Wähle eine zufällige innere Facette F (Dreieck) aus.
- Füge einen neuen Knoten u in F ein.





Erster Ansatz zur Erzeugung eines maximalen planaren Graphen

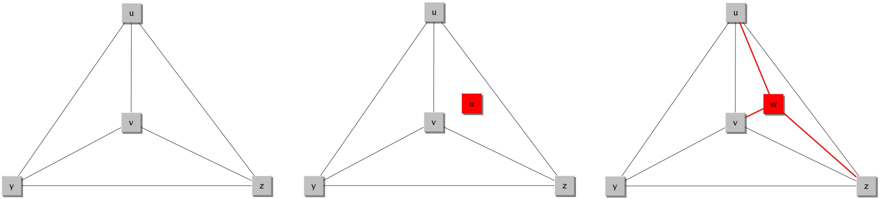
- Initialisiere Graph G als Dreieck.
- Füge rekursiv die restlichen $n - 3$ Knoten wie folgt hinzu:
- Wähle eine zufällige innere Facette F (Dreieck) aus.
- Füge einen neuen Knoten u in F ein.
- Verbinde u mit allen Knoten aus $V(F)$.



- Im i-ten Schritt hat G $3i - 6$ Kanten, ist also maximal.

Erster Ansatz zur Erzeugung eines maximalen planaren Graphen

- Initialisiere Graph G als Dreieck.
- Füge rekursiv die restlichen $n - 3$ Knoten wie folgt hinzu:
- Wähle eine zufällige innere Facette F (Dreieck) aus.
- Füge einen neuen Knoten u in F ein.
- Verbinde u mit allen Knoten aus $V(F)$.



- Im i-ten Schritt hat G $3i - 6$ Kanten, ist also maximal.
- *Aufwand:* $O(n)$

Problem

- Vorgehen nicht vollständig.
- Es gibt immer einen Knoten mit Grad 3.
- Folgender 4-regulärer kann so nicht konstruiert werden.

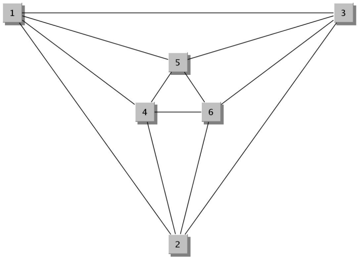


Abbildung: Ein 4-regulärer maximaler planarer Graph.

Neuer Ansatz: Kontraktion

- $G = (V, E)$ Graph, $e = \{u, v\} \in E$
- $G/e = (V', E')$ Graph, der durch Kontraktion der Kante e entsteht

$$\begin{aligned} V' &= (V - \{u, v\}) \cup \{v_e\} \\ E' &= \{\{x, y\} \in E \mid x, y \notin \{u, v\}\} \cup \\ &\quad \{\{v_e, x\} \mid x \in V', \{u, x\} \in E \vee \{v, x\} \in E\} \end{aligned}$$

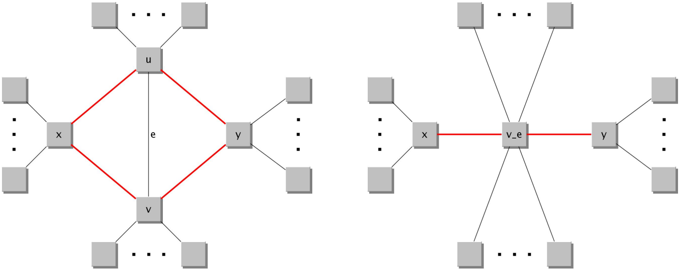


Abbildung: Kontraktion schematisch

Hinführung

Lemma

- *Sei* $G = (V, E)$ *ein maximaler planarer Graph mit* $n \geq 4$ *Knoten*
- *dann existiert* $e = \{u, v\} \in E$ *mit* $|N(u) \cap N(v)| = 2.$

Hinführung

Lemma

- *Sei* $G = (V, E)$ *ein maximaler planarer Graph mit* $n \geq 4$ *Knoten*
- *dann existiert* $e = \{u, v\} \in E$ *mit* $|N(u) \cap N(v)| = 2$.

Lemma

- *Sei* $G = (V, E)$ *ein maximaler planarer Graph mit* $n \geq 4$ *Knoten und*
- *sei* $e = \{u, v\} \in E$ *mit* $|N(u) \cap N(v)| = 2$,
- *dann ist* G/e *maximal planar.*

Hinführung

Lemma

- *Sei* $G = (V, E)$ *ein maximaler planarer Graph mit* $n \geq 4$ *Knoten*
- *dann existiert* $e = \{u, v\} \in E$ *mit* $|N(u) \cap N(v)| = 2$*.*

Lemma

- *Sei* $G = (V, E)$ *ein maximaler planarer Graph mit* $n \geq 4$ *Knoten und*
- *sei* $e = \{u, v\} \in E$ *mit* $|N(u) \cap N(v)| = 2$*,*
- *dann ist* G/e *maximal planar.*

- Es gibt eine Folge von maximalen planaren Graphen $G = G_0, G_1, \ldots, G_{n-4}, G_{n-3} = K_3$ mit
- $G_i = G_{i-1}/e$ für eine Kante e von G_{i-1}

Kontraktion mit mehr als zwei gemeinsamen Nachbarn

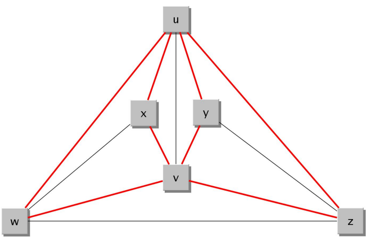


Abbildung: Kontraktion mit mehr als zwei gemeinsamen Nachbarn: G/e nicht maximal planar

Extraktion

- “Umkehrung” einer Kontraktion

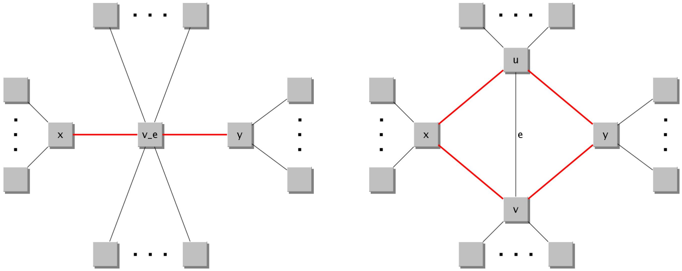


Abbildung: Extraktion schematisch

Algorithmus: Maximaler Planarer Graph

Algorithmus

- *Eingabe: Anzahl der Knoten n*
- *Ausgabe: ein maximaler planarer Graph mit n Knoten*

Initialisiere $G = (V, E)$ *als Dreieck.*
for $i = 4 \ldots n$ **do**
    Wähle zufälligen Knoten $v_e \in V$.
    Wähle zufällige Kanten $\{v_e, x\}, \{v_e, y\} \in E$.
    Erzeuge neue Knoten u, v.
    $V \leftarrow (V - \{v_e\}) \cup \{u, v\}$
    Füge Kanten gemäß Zeichnung ein.
end for

- *Aufwand:* $O(n)$

Algorithmus: Planarer Graph

Algorithmus

- *Eingabe: Anzahl der Knoten n, Anzahl der Kanten k*
- *Ausgabe: ein zufälliger planarer Graph mit n Knoten und k Kanten*

Erzeuge maximalen planarer Graphen $G = (V, E)$ *(*$m = 3n - 6$*).*
while $m > k$ **do**
  Lösche zufällige Kante.
end while

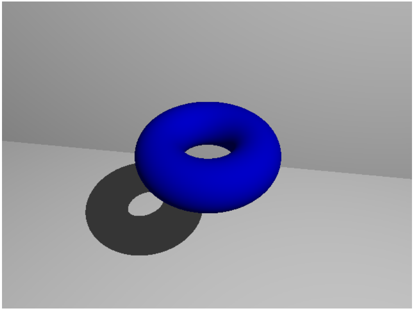
- *Aufwand:* $O(n + n) = O(n)$

Vollständigkeit

Theorem

- *Der Algorithmus erzeugt jeden planaren Graphen mit positiver Wahrscheinlichkeit.*

Geschlecht



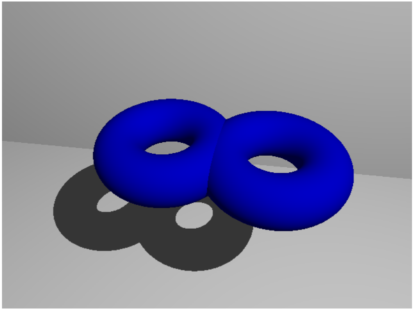


Abbildung: Fläche vom Geschlecht eins bzw. zwei.

Planare Graphen Höherer Ordnung

Definition

- $G = (V, E)$ planarer Graph der *Ordnung* g, gdw.
- G lässt sich auf einer Fläche vom Geschlecht g so zeichnen, dass seine Kanten *überschneidungsfrei* sind.

Warum Planare Graphen Höherer Ordnung?

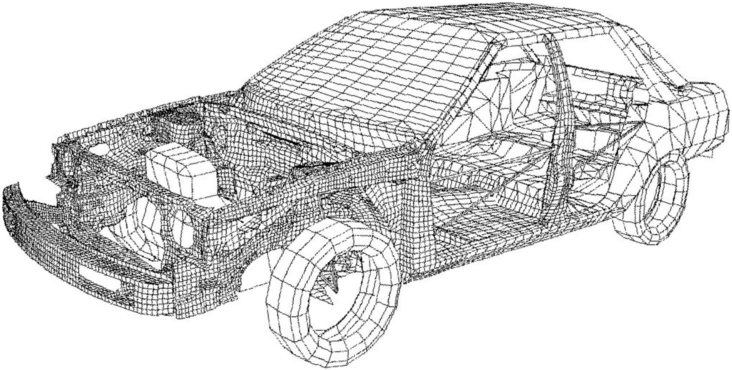
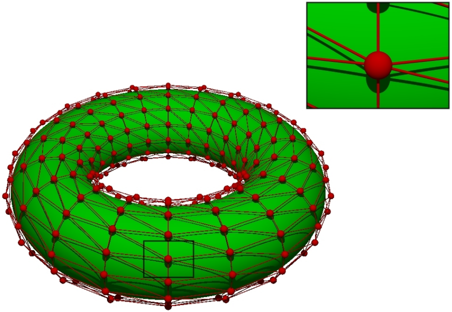


Illustration: Ein planarer Graph Höherer Ordnung

- Triangulierter Torus mit n Knoten: 6-regulär $\Rightarrow 3n$ Kanten



Algorithmus: Planarer Graph Höherer Ordnung

Algorithmus

Erzeuge planaren Graphen mit n Knoten und k Kanten.
for $i = 1, \ldots, g$ **do**
  Wähle zufällige Facetten F_1, F_2.
  Verbinde die Facetten mit einer Kante (durch den "Tunnel") $\Rightarrow$ neue "Tunnelfacette" F^*.
  Wähle zufälligen Parameter l für die Anzahl der "Tunnelkanten" aus $\{1, \ldots, |V(F_1)| + |V(F_2)|\}$.
  for $j = 2, \ldots, l$ **do**
    Wähle zufällige "Tunnelfacette" und splitte diese an zwei geeigneten Knoten.
    Aktualisiere Facetten-Information.
  end for
end for

- *Aufwand:* $O(n + gn)$.

Zeichnungen von “torusplanaren” Graphen

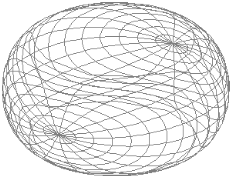


Abbildung: Parametrisierung

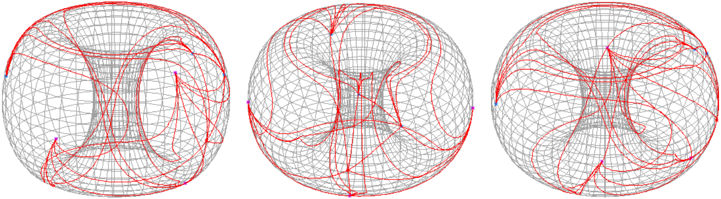


Abbildung: “torusplanarer” K_7

Zusammenfassung

- Effizienter und vollständiger Algorithmus zur Erzeugung von planaren Graphen.
- Reduktion auf "einfacheres" Problem: Erzeugung eines maximalen planaren Graphen.
- Vorgehen auch auf andere Generatoren übertragbar: z.B. Graphen mit vorgegebener Core-Struktur
- Planare Graphen höheren Geschlechts, Erweiterung des Algorithmus

Ausblick

- Welche Klasse von Graphen liefert erster Ansatz?
- Verteilung der Graphen?
- Vollständigkeit und Effizienz des Algorithmus zur Erzeugung von planaren Graphen höherer Ordnung?