Kap. 5: Planaritätsbasierte Verfahren

Prof. Dr. Petra Mutzel

Lehrstuhl für
Algorithm Engineering LS11
Universität Dortmund

23. VO WS07/08 21. Januar 2008

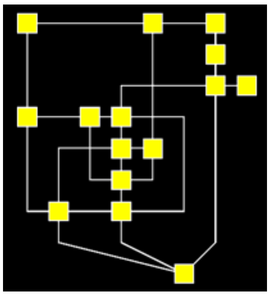
Literatur für diese VO

- M. Kaufmann, D. Wagner (Eds.): Drawing Graphs: Methods and Models, Lecture Notes in Computer Science, Tutorial, Vol. 2025, 2001, ISBN 3-540-42062-2.

- M. Jünger, P. Mutzel (Eds.): Kapitel 2: Technical Foundations aus: Graph Drawing Software, Mathematics and Visualization, Springer Verlag, 2004, ISBN 3-540-00881-0.

Kap. 5: Planaritätsbasierte Verfahren

- Zustandsdiagramme
- State Charts
- Datenmodelle
- ER-Diagramme
- UML Diagramme
- ...



Batini, Nardelli, Tamassia 1986

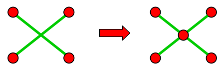
Planarisierungsverfahren - Idee

(1) Finde kreuzungsminimale Zeichnung

Kreuzungsminimierungsproblem

➡ Größtes Planares Untergraphenproblem

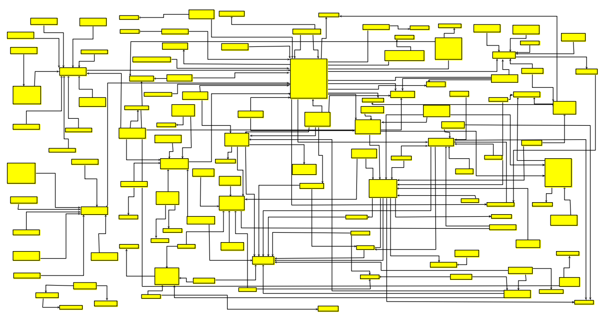
(2) Ersetze die Kreuzungen durch Knoten



(3) Verwende planares Zeichenverfahren

Animation des Planarisierungsverfahrens mittels größter planarer Untergraph

Datenbank Versicherung



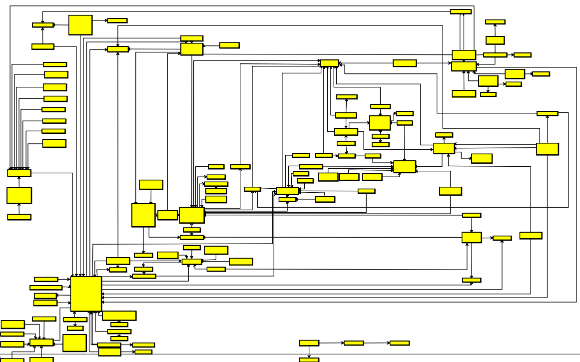
Datenbank Versicherung

Datenmodell
- 120 Objekte
- 161 Relationen

Originalzeichnung
- 122 Kreuzungen
- nicht-hierarchisch
- Anomalien

Unsere Zeichnungen
- Hierarchie
- Planarisierung
  (7 Kreuzungen)

Datenbank Versicherung



Planarisierungsverfahren

1. Grösstes Planares Untergraphenproblem
   - ➡ Graphenalgorithmen
   - NP-schwierige Optimierung
2. Berechnung einer kombinatorischen Einbettung
   - Planaritätstest ✓, Einbettung mit maximaler Außenfläche ✓, Einbettung mit Constraints ✓
3. Wiedereinfügen der entfernten Kanten
   - Kürzeste Wege Algorithmen ✓, Optimales Kanteneinfügen ✓, Kanteneinfügen mit Constraints ✓
4. Anwendung planarer Zeichenalgorithmen
   - Planare straightline Verfahren mittels kanonischer Ordnung ✓
   - ➡ Planare orthogonale Verfahren mittels Netzwerkflüssen

zu 1) Bestimmung eines größten planaren Untergraphen

- **Definition:** Ein Untergraph $H=(W,P)$ mit $W \subseteq V$, $P \subseteq E$, eines Graphen G heisst (ungewichteter) **größter planarer Untergraph** (**maximum planar subgraph**, **MPS**) von G, wenn für jeden planaren Untergraphen $H'=(W',P')$ gilt: $|P'| \leq |P|$.

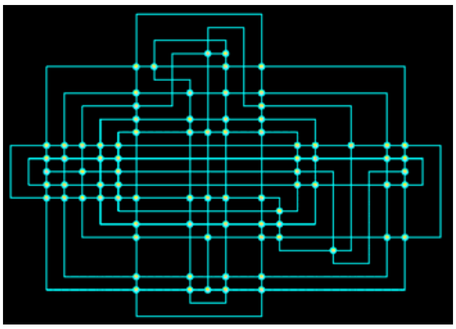
Bemerkungen:
- MPS ist äquivalent zum Problem in G die kleinste Kantenmenge zu entfernen, so dass der Restgraph planar ist.
- Berechnung eines MPS ist NP-schwer.

Maximal planarer Untergraph

- **Definition:** Ein Untergraph $H=(V,P)$ mit $P \subseteq E$ eines Graphen G heisst **maximal planarer Untergraph** (**maximal** planar subgraph) von G, wenn keine Kante $e \in E \setminus P$ existiert, so dass $H=(V,P \cup \{e\})$ planar ist.

Bemerkungen:
- Beispiel:
- Ein maximal planarer Untergraph kann in polynomieller Zeit berechnet werden.

Maximum vs. Maximal Planar Subgraph



Verfahren zur Bestimmung eines maximal planaren Untergraphen

Gegeben: G=(V,E), Greedy-Algorithmus:
1. Initialisierung: P=ø
2. Für alle e∈E:
   - Falls G´=(V,P∪{e}) planar ist, dann: P= P∪{e}

Behauptung: H=(V,P) ist ein maximal planarer Untergraph von G.

Beweis: Untergraph: ✓ planar: ✓ maximal: existiert eine Kante, so dass P∪{e} planar? nein ✓

Laufzeit: naiv implementiert: quadratisch: |E||V|, aber es gibt auch inkrementelle Planaritätstests, dann geht es (theoretisch) fast linear. Mit SPQR-Baum: |E| log |V|.

Qualität des Greedy-Verfahrens

Behauptung: Die optimale Lösung ist höchstens 3 Mal besser als die gefundene Lösung, d.h. der obige Greedy-Algorithmus ist ein 1/3-Approximationsalgorithmus für das MPS Problem.

Beweis: Die berechnete Greedy-Lösung enthält mindestens V -1 Kanten. Die optimale Lösung höchstens 3|V|-6 Kanten.
Damit: $|P_{ALG}| \geq |V|-1 \geq |V|-2 = 1/3\,(3|V|-6) \geq 1/3|P_{OPT}|$

s. Beispiel

Bemerkung: Das ist zwar theoretisch schön (konstanter Approximationsfaktor), aber bezüglich der entfernten Kanten in der Praxis nicht gut genug.

Alternative Verfahren

- Planarer Untergraph basierend auf Planaritätstest von Lempel-Even-Cederbaum
  – Untergraphen sind zwar nicht maximal, aber dafür relativ groß (besser als Greedy, obwohl nicht maximal)
  – Es gibt eine Reihe von Publikationen, die jeweils versucht haben, die Untergraphen maximal zu machen, die aber alle fehlerhaft sind
  – wir konnten dann zeigen, dass es grundsätzlich nicht möglich ist, mit dem Verfahren maximale Untergraphen zu erhalten

- **Maximal** planare Untergraphen basierend auf dem Planaritätstest von Hopcroft-Tarjan

Offenes Problem: Planare Untergraphen basierend auf dem Planaritätstest von Boyer-Myrvold.

Verfahren zur Bestimmung eines maximum planaren Untergraphen

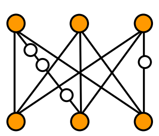
- Es existiert ein 4/9-Approximationsalgorithmus
  [Calinescu, Fernandes, Finkler, Karloff: A better approximation algorithm for finding planar subgraphs, Journal of Algorithms, vol. 27, no. 2, 269-302, 1998]

- **Branch-and-Cut Algorithmus für MPS: →**

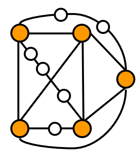
ILP-Formulierung für das Maximum Planar Subgraph Problem

Basiert auf:

Kuratowski´s Theorem: Ein Graph ist **planar** if genau dann wenn er keine Unterteilung von $\mathbf{K_{3,3}}$ oder $\mathbf{K_5}$ enthält.



$K_{3,3}$



K_5

Unterteilungen

Integer Linear Programming Formulierung

Variablen x_e für alle Kanten e

$$x_e \in \begin{cases} 0 & \text{Kante } e \text{ ist nicht im Subgraph} \\ 1 & \text{Kante } e \text{ ist im Subgraph} \end{cases}$$

Zielfunktion: $\max \sum_{e \in E} x_e$

Kuratowski Ungleichungen:

$$\sum_{e \in K} x_e \leq |K| - 1 \qquad \forall \text{ Kuratowski Subgraphen } K \text{ in } G$$

$K_{3,3}$ and K_5 Unterteilungen

Branch-and-Cut Algorithmus für Maximum Planare Subgraphen

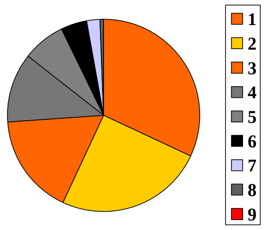
- Planares Untergraphenpolytop: [Jünger, Mutzel ´93, ´96]
  - Kuratowski Ungleichungen, Euler Ungleichungen, s-chorded cycle inequalities, even Moebius ladders, ...

- Branch-and-cut code (ABACUS) [Fialko ´98, Jansen 2007]
  - Heuristische Separationsroutinen für Kuratowski und Euler Ungleichungen

Leider keine Zeit mehr uns das genauer anzuschauen schade ☹

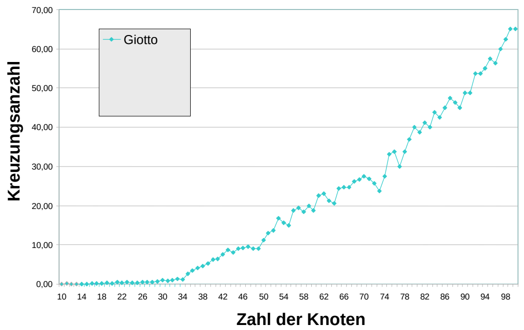
Experimentelle Resultate aus dem Jahr 1997

Anzahl der entfernten Kanten



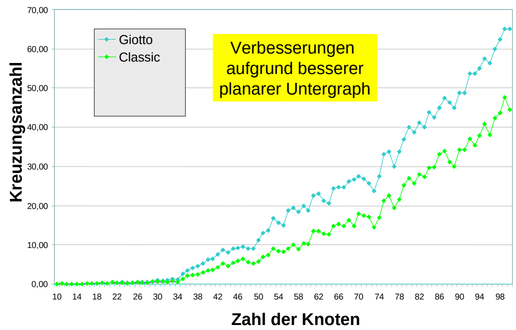
- 11529 Graphen (Rom-Lib)
  - 3280 planar
  - 8249 nicht-planar
- 10-100 Knoten
- Ergebnisse bis zu 65 Knoten:
  - insgesamt 7925 Graphen
  - 4654 nicht-planar
  - 4460 (96%) optimal innerhalb 1 hour CPU

Fortschritte bei der Planarisierungsmethode Anzahl der Kreuzungen für Benchmarks

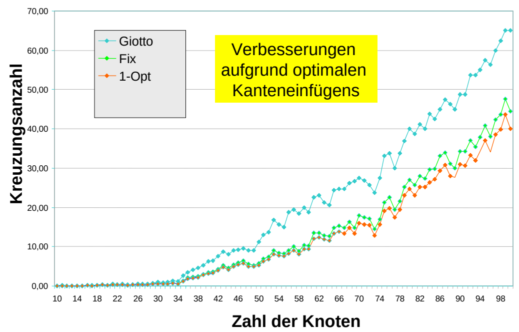


Anzahl der Kreuzungen für Benchmarks

(Unsere experimentelle Werte 2004)



Anzahl der Kreuzungen für Benchmarks



Anzahl der Kreuzungen für Benchmarks