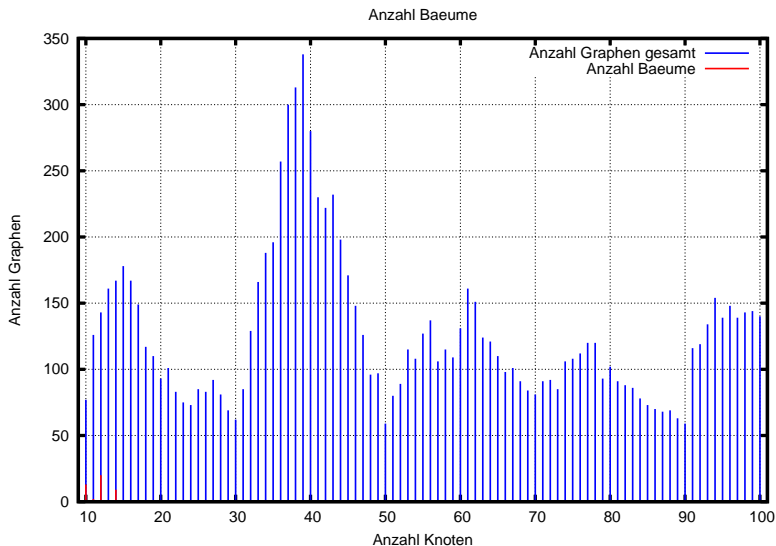


Korrektur Blatt 1



Schichtzuweisung mit Breitenbeschränkung AT&T Graphen

Dirk Ribbrock Nils Kriege Sophia Kardung

Universität Dortmund - LS 11 - Graphenzeichnen

Dortmund, 11. Dezember 2007

Algorithmus Coffman-Graham

- Phase 1: Berechne für jeden Knoten ein Label
 - Label 1,2,.. erhalten die Quellknoten
 - nächstes Label erhält der Knoten:
dessen Vorgänger schon ein Label haben
mit lexikographisch kleinster Labelmenge (=Menge der Label
der Vorgänger)
- Phase 2: Starte auf unterster Schicht(Rank) und wähle den Knoten mit größtem Label, der platziert werden darf.

Laufzeit Coffman-Graham

Laufzeit: $O(n^2)$

- Initialisierung: $O(n)$
- Phase 1: $O(n^2)$ (Vergebe n Label, je Zeit $O(n)$)
- Phase 2: $O(n^2)$ (Teile jedem Knoten ein Rank zu, finde nächsten Knoten, prüfe ob Knoten platzierbar)

Datenstrukturen Coffman-Graham

- NodeArray: zu jedem Knoten Label(int) und Labelmengen (Vektor) speichern
- Liste NodesToLabel: anfangs Quellknoten, nachher Nachfolger der gelabelten Knoten
- Liste NodesToRank: anfangs Senken, nachher Vorgänger der Knoten die Rank bekommen haben
- Liste NodesToRankLater: Knoten die noch nicht platzierbar sind, wird in jedem Durchlauf wieder zu NodesToRank hinzugefügt

Algorithmus TopSort mit Breitenbeschränkung

- Wähle immer eine Quelle und entferne sie dann
- Überprüfe nach jedem platzierten Knoten ob max. Breite erreicht ist, wenn ja erhöhe Schicht um eins
- Laufzeit: $O(|V| + |E|)$

Vergleich der Verfahren für AT&T Graphen

AT&T Graphen

- 1277 gerichtete Graphen (zwischen 10 und 100 Knoten)
- alle zusammenhängend und kreisfrei

→ ohne Preprocessing für Sugiyama Layout geeignet

Vergleich der Verfahren für AT&T Graphen

AT&T Graphen

- 1277 gerichtete Graphen (zwischen 10 und 100 Knoten)
- alle zusammenhängend und kreisfrei

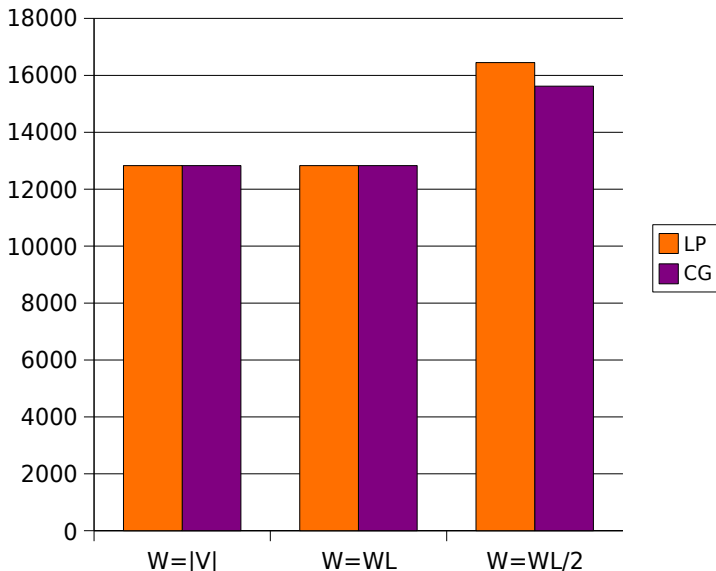
→ ohne Preprocessing für Sugiyama Layout geeignet

Testläufe

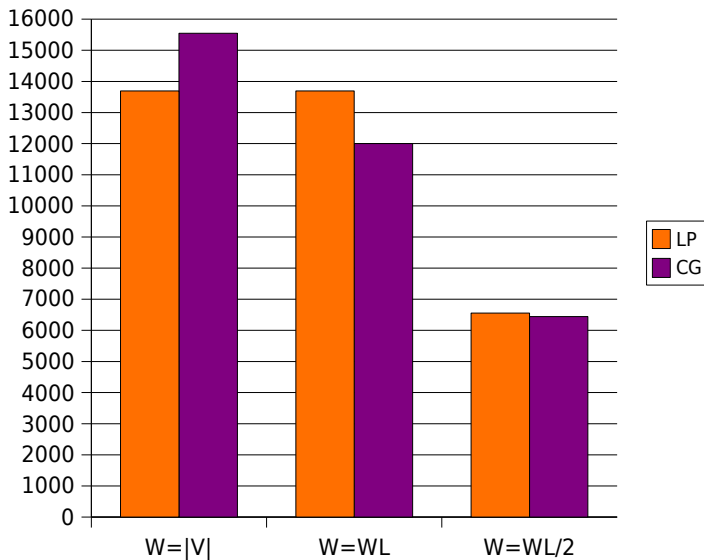
- ① Maximalbreite $W = |V|$
- ② Maximalbreite $W = W_L$ (Breite der Instanz mit Longest-Path)
- ③ Maximalbreite $W = W_L/2$

Ermittelte Werte: Breite, Höhe, tatsächliche Breite, Anzahl Dummyknoten

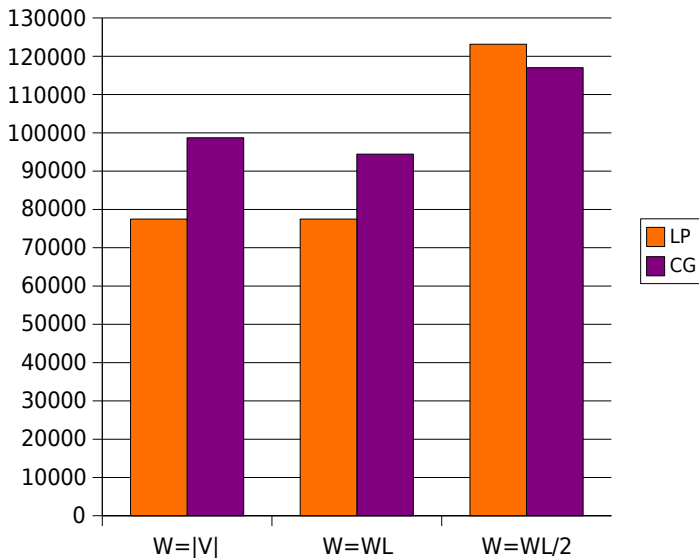
Höhe (Summe über alle Graphen)



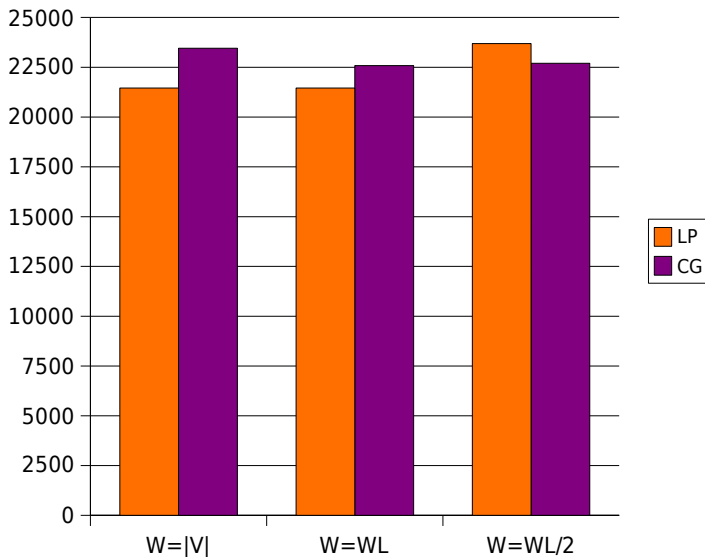
Breite



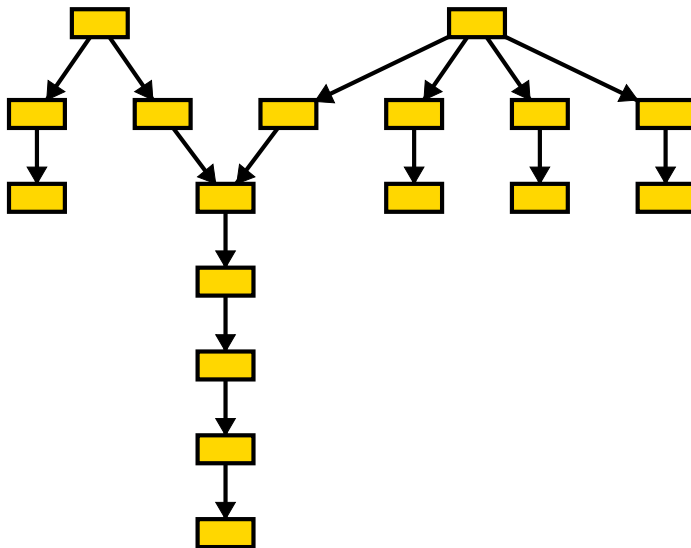
Anzahl Dummyknoten (Summe)



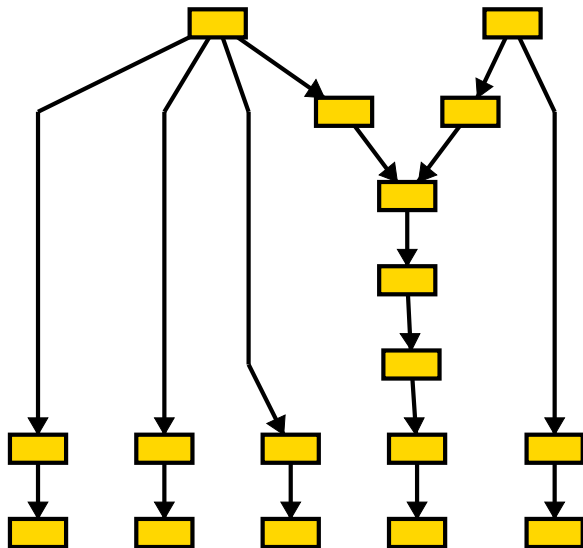
Breite (inkl. Dummyknoten)



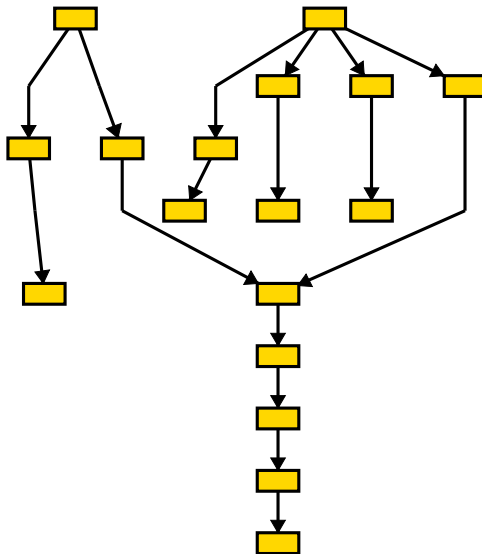
g.17.7.gml, Longest-Path ohne Breitenbeschränkung



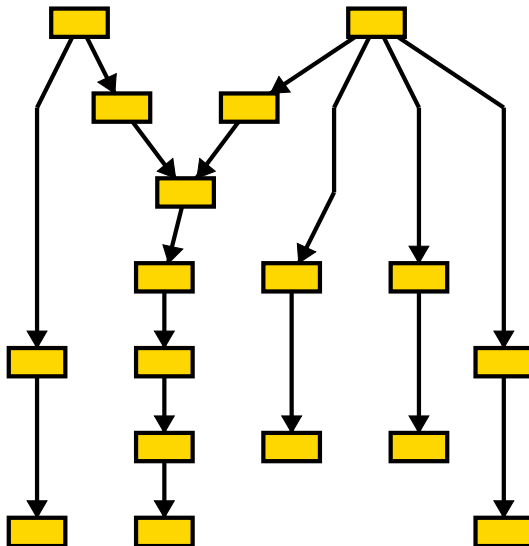
g.17.7.gml, Coffman-Graham ohne Breitenbeschränkung



g.17.7.gml, Longest-Path und Breite 3



g.17.7.gml, Coffman-Graham und Breite 3



Fazit (1/2)

Longest-Path mit Breitenbeschränkung

- Liefert akzeptable Ergebnisse bezüglich der Höhe
- Anzahl der Dummyknoten akzeptabel
- einfach implementierbar, effizient

Fazit (1/2)

Longest-Path mit Breitenbeschränkung

- Liefert akzeptable Ergebnisse bezüglich der Höhe
- Anzahl der Dummyknoten akzeptabel
- einfach implementierbar, effizient

Coffman-Graham Algorithmus

- Liefert gute Ergebnisse bezüglich der Höhe
- neigt zu “langen” Kanten → viele Dummyknoten (kann durch zusätzliches Postprocessing vermieden werden)
- Vorteile besonders bei starker Breitenbeschränkung
- aufwendiger als Longest-Path

Fazit (2/2)

- Breitenbeschränkung reduziert Breite inkl. Dummyknoten nicht → Kanten müssen dicht nebeneinander gezeichnet werden können
- Breite der Zeichnung hängt stark von Phase 3 (Knotenpositionierung) ab
- Qualität der Verfahren schwankt von Instanz zu Instanz

Ende

Vielen Dank für die Aufmerksamkeit