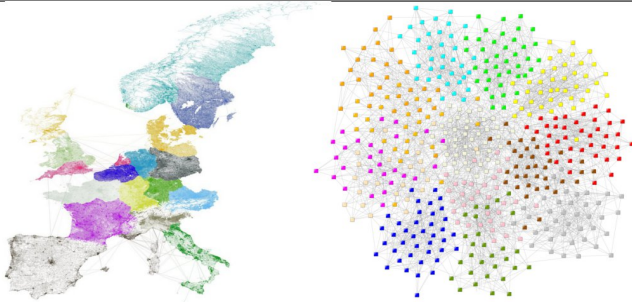


# Graphgeneratoren

Praktikum Algorithm Engineering | Sommer 2011

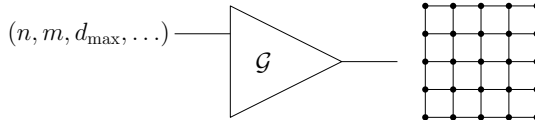
LEHRSTUHL FÜR ALGORITHMIK I+II



# Was sind Graphgeneratoren?

## Problemstellung

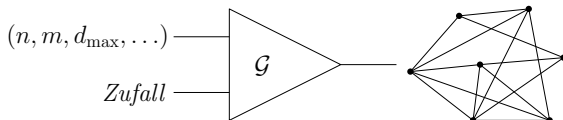
- Gegeben: Graphklasse  $\mathcal{C}$
- Gesucht: Algorithmus, der  $G \in \mathcal{C}$  erzeugt



# Was sind Graphgeneratoren?

## Problemstellung

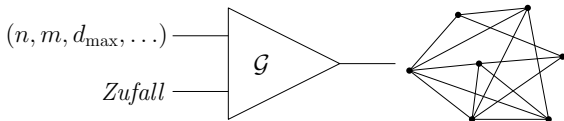
- Gegeben: Graphklasse  $\mathcal{C}$
- Gesucht: **randomisierter** Algorithmus, der  $G \in \mathcal{C}$  erzeugt



# Was sind Graphgeneratoren?

## Problemstellung

- Gegeben: Graphklasse  $\mathcal{C}$
- Gesucht: **randomisierter** Algorithmus, der  $G \in \mathcal{C}$  erzeugt



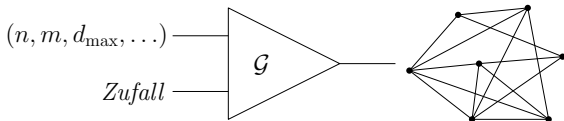
## Desiderata

- **vollständig**  $\mathbb{P}(X = G) > 0$  für alle  $G \in \mathcal{C}$
- **gleichverteilt**  $\mathbb{P}(X = G) = 1/|\mathcal{C}|$  für alle  $G \in \mathcal{C}$

# Was sind Graphgeneratoren?

## Problemstellung

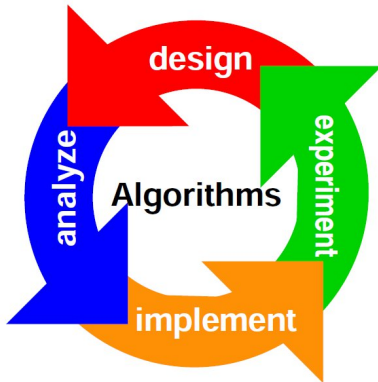
- Gegeben: Graphklasse  $\mathcal{C}$
- Gesucht: **randomisierter** Algorithmus, der  $G \in \mathcal{C}$  erzeugt



## Desiderata

- **vollständig**  $\mathbb{P}(X = G) > 0$  für alle  $G \in \mathcal{C}$
- **gleichverteilt**  $\mathbb{P}(X = G) = 1/|\mathcal{C}|$  für alle  $G \in \mathcal{C}$

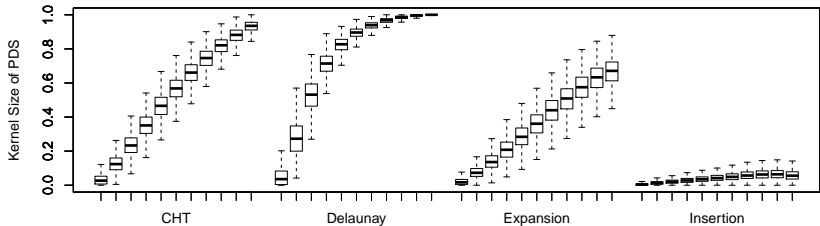
# Wozu Generatoren?



## Experimente

- beschränkte Verfügbarkeit von Realwelt-Instanzen
- größere Varianz beim Testen, höhere Robustheit
- Skalierbarkeit
- Validierung von Algorithmen (berechnet der Algorithmus das generierte Ergebnis?)

# Vorsicht bei Auswahl/Design von Generatoren



aus: S.Meinert, D.Wagner, *An Experimental Study on Generating Planar Graphs*, AAIM 2011, to appear.

- Insertion: beweisbar kleine Baumbreite und beweisbar kleiner Kern, nicht vollständig
- Ergebnisse suggerieren extrem gutes Abschneiden der Algorithmen [Alber, Betzler, Niedermeier, 2006]

# Netzwerkanalyse

## Struktur-Analyse

- Generator als Modell
- Validierung von Modellen (je besser das Ergebnis der Modell-Simulation mit der wirklichen Welt übereinstimmt, desto valider das Modell)



# Ziele

## Zielstellung

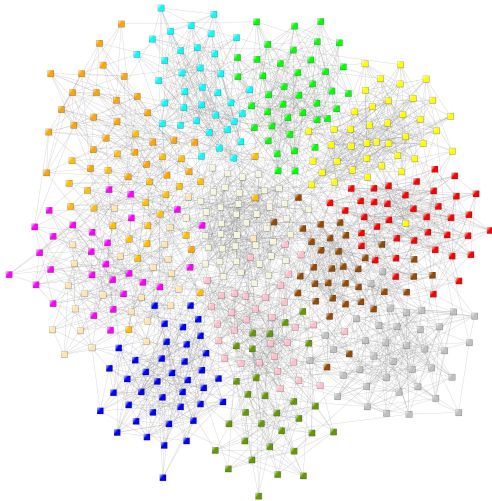
- Entwicklung von Graphgeneratoren / Analyse
- Implementierung / Evaluierung
- Algorithm Engineering
- Präsentation der Ergebnisse

## Zielgruppe

- Diplom- / Master-Studenten Informatik
- Bachelor- / Master-Studenten Informationswirtschaft
- 6 Leistungspunkte



# Generatoren für soziale Netzwerke



# Generatoren für soziale Netzwerke

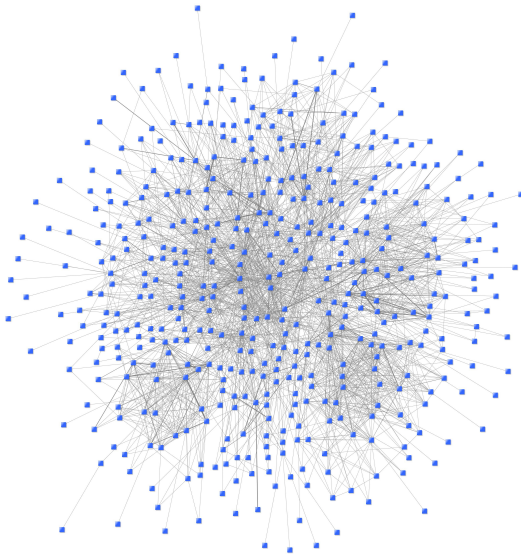
## Definition (Soziales Netzwerk)

Netzwerk bestehend aus Individuen (Knoten), die durch gemeinsame Merkmale miteinander assoziiert sind (Kanten).

## Beispiele

- Freundschaft (Facebook)
- Kaufverhalten (Payback)
- Co-Autorenschaft (Citeseer)

# E-Mail Netzwerk der Fakultät (Snapshot)



# Aufgabe

## Zentrale Fragestellungen

- Was macht diese Netzwerke aus?
- Wie kann man ähnliche Netzwerke erzeugen?
- Lassen sich daraus Erkenntnisse über die Entstehung der realen Netzwerke ableiten?

## Zielstellung

- Analyse bestehender Netzwerke
- Anpassung bestehender Modelle

# Analyse

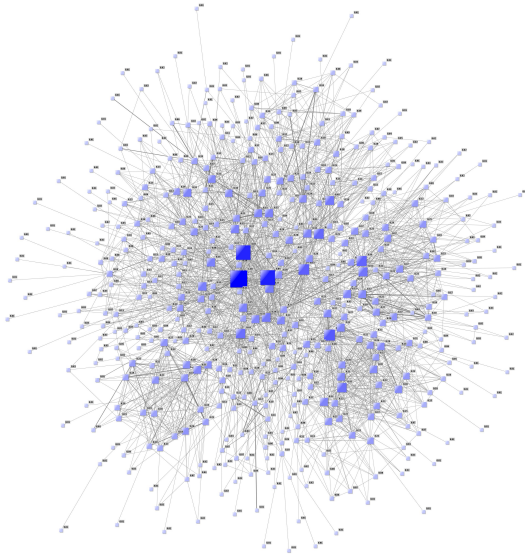
## Allgemeine Charakteristika

- Anzahl Knoten/Kanten, Dichte
- Durchmesser
- Gradverteilung
- Zentralitätsmaße

## Strukturelle Charakteristika

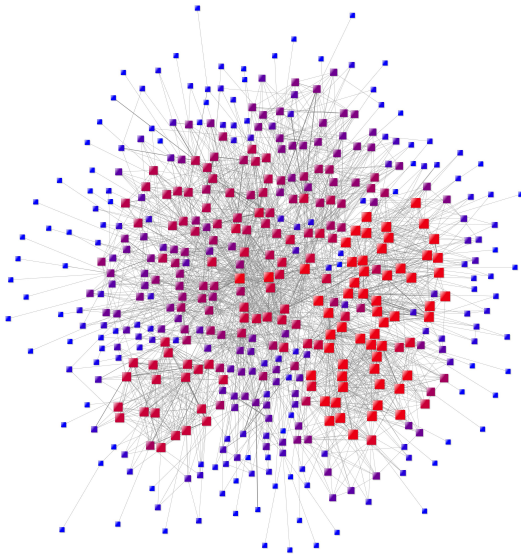
- Core-Struktur
- Cluster-Struktur

# Zentralität

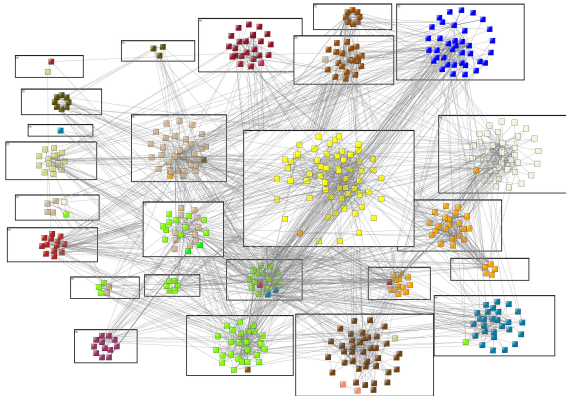




# Core-Struktur



# Cluster-Struktur



# Watts-Strogatz Modell<sup>1</sup>

Kennen wir nicht alle jemanden in der Ferne?

## Hypothese

Die durchschnittliche Pfadlänge in sozialen Netzwerken ist kurz (Experimente von Stanley Milgram, 1967).

## Small-World Graph

- starte mit regelmäßigem “Kreis”, in dem jeder Knoten mit seinen  $k$  nächsten Nachbarn verbunden ist
- biege Kanten mit Wahrscheinlichkeit  $p$  zufällig um

**Bemerkung:**  $p$  interpoliert zwischen Ordnung und Chaos

---

<sup>1</sup>[Watts und Strogatz, 1998]

# Barabási-Albert Modell<sup>2</sup>

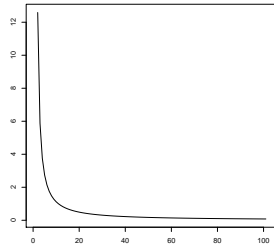
Lassen wir uns nicht alle (hin und wieder) vom Mainstream leiten?

## Beobachtung

Viele große Netzwerke haben eine Power-Law Gradverteilung, d.h. die Anzahl der Knoten mit Grad  $x$  ist etwa  $cx^k$  ( $k < 0$ ).

## Preferential Attachment

- Beginne mit  $m_0$  Knoten
- Füge iterativ Knoten hinzu
- Jeder Knoten wählt  $m$  Nachbarn proportional zu deren Grad
- beliebte Knoten werden noch beliebter



<sup>2</sup>[Barabási und Albert, 1999]

# Random Intersection Graphs<sup>3</sup>

Entsteht Freundschaft nicht meistens aus gemeinsamen Interessen?

## Idee

Wir lassen uns bei unseren Entscheidungen nicht so sehr vom Zufall, sondern mehr von einem komplexen Geflecht von Interessen leiten.

## Random Intersection Graph

- Ziehe für jeden Knoten  $v$  zufällige Menge  $S_v$
- $uv \in E$  genau dann, wenn  $S_u \cap S_v \neq \emptyset$

---

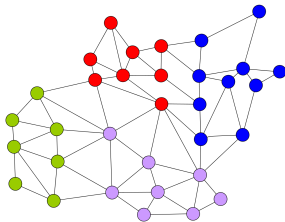
<sup>3</sup>[Karoński et al., 99]



# $\epsilon$ -Balanced Graph Partitioning

Partition graph  $G = (V, E, c : V \rightarrow \mathbf{R}_{>0}, \omega : E \rightarrow \mathbf{R}_{>0})$   
into  $k$  disjoint blocks s.t.

- total **node weight** of each block  $\leq \frac{1 + \epsilon}{k}$  total node weight
- total weight of **cut** edges as small as possible



## Applications:

sparse matrix reordering, VLSI design, graph compression, ...

# $|V| > 1\,000\,000\,000$ Filmgraphen

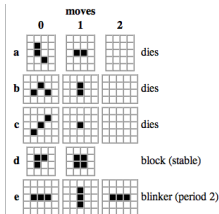


- riesige Graphen aus Filmen
  - 1 Knoten: Pixel im Raum
  - 2 Nachbarschaft?
  - 3 Kantengewichte?
  - 4 Framework?
- $|G| >$  Arbeitsspeicher
- Streaming Model
- Partitionierung der Graphen
  - ⇒ Externe Algorithmen
  - ⇒ Parallele Algorithmen



# 3d Cellular Automata Graphs

- Grid of cells with a finite number of states, such as “On“ and ”Off“
- For each cell we define a set of neighbors
- A new generation is created, according to some fixed rule determining the new state of each cell in terms of the current state of the cell and the states of the cells in its neighborhood.



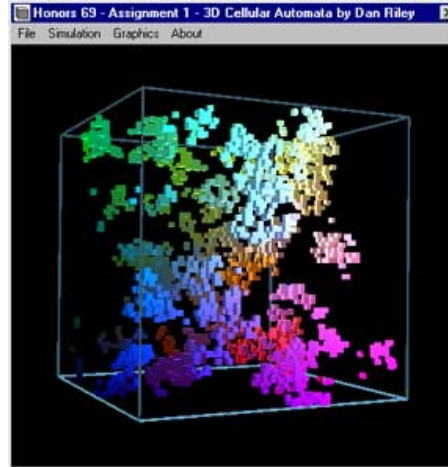
## 2d Conway's Game of Life

- if a live cell has 2 or 3 live neighbors it stays alive
- if a dead cell has 3 live neighbors it becomes alive



# 3d Cellular Automata Graphs

- In 3d case each cell has 26 neighbors
- Task - Generate a graph based on cellular automaton states
  - Nodes - cells that are "ON"
  - Edges connect two "ON" nodes if they are neighbors
- Streaming computation of largest connected component components



# Themenvergabe

## Generatoren für soziale Netzwerke

- Small World
- Preferential Attachment
- Intersection Graphs

## Generatoren für Graphpartitionierung

- Filmgraphen
- 3D Zellular-Automaten