Kap. 6: Kräftebasierte Verfahren
6.5 Verfahren für große Graphen

Prof. Dr. Petra Mutzel

Lehrstuhl für
Algorithm Engineering LS11
Universität Dortmund

27./28. VO | WS07/08 | 4./5. Februar 2008

Original-Literatur für diese VO

- Fruchterman, Reingold: Graph drawing by force-directed placement, Software-Practice and Experience, vol. 21 (11), 1129-1164, 1991.
- Quigley und Eades: FADE: Graph Drawing, Clustering, and Visual Abstraction, Graph Drawing 2000, LNCS, 1984, Springer, 197-210
- Tunkelang: JIGGLE: Java Interactive Graph Layout Environment, Graph Drawing 1998, LNCS 1547, Springer, 413-422, 1998.
- Aluru, Prabhu, Gustafson: Truly Distribution-Independent Algorithms for the N-body Problem, Proc. 1994 ACM/IEEE Conf. on Supercomputng, 420-428, 1994.

Original-Literatur für diese VO

- Callahan und Kosaraju: A decomposition of multidimensional point sets with applications to k-nearest-neighbors and n-body potential fields, Journal of the ACM 42 (1), 67-90, 1995.
- Hachul und Jünger: Drawing large graphs witha potential-field-based multilevel algorithm, Graph Drawing 2004, LNCS, vol. 3383, Springer, 285-295, 2005.
- C. Walshaw: A multilevel algorithm for force-directed graph drawing, Graph Drawing 2000, LNCS, vol. 1984, Springer, 197-210, 2001.

Überblick

6.5 Verfahren für große Graphen
6.5.1 Gitter Methode
6.5.2 Zerlegungsbaum Methoden
6.5.3 Multilevel Methoden
6.5.4 Algebraische Verfahren
6.5.5 Experimentelle Evaluation

- Abschlußbemerkungen zu GD VO
- Prüfungsmodalitäten
- Vorlesungsbewertung

6.5 Verfahren für große Graphen

- **Ziel hier:** Layout für Graphen mit mehreren hundert oder tausend Knoten bis zu 100.000 Knoten
- **Problem:** die klassischen kräfte-/energiebasierten Verfahren sind zu langsam
  - Berechnung der abstoßenden Kräfte in jeder Iteration: $\theta(|V|^2)$
  - Bsp: Laufzeit für Layout eines Graphen mit 100 Knoten 1 Sekunde, mit 10.000 Knoten 2:46 Stunden

Überblick

Lösung 1: Berechne die abstoßenden Kräfte nur approximativ
- ➡ Gitter Methode von Fruchterman & Reingold [FR91]
- ➡ Zerlegungsbaum Methode von Tunkelang [Tu98] und Quigley & Eades [QE00]

Lösung 2: Wende eine Multilevel Methode an
- ➡ Algorithmus von Walshaw [W00] (max. Matchings)
- Gajer, Goodrich & Kobourov [GGK00] (max. unabhängige Mengen)
- Harel & Koren [HK01] (k-center)
- Koren, Carmel & Harel [KCH02] (max. Matchings)
- Hachul & Jünger [HJ05] (Subgraphenzerlegung)

6.5.1 Gitter Methode

Idee:
- Unterteile die Zeichenfläche in ein regelmäßiges Gitter ein der Größe $\sqrt{|V|/k} \times \sqrt{|V|/k}$)
- FR91 wählten: $k=2$
- Berechne abstoßende Kräfte bei v nur für Knoten u, die sich im Feld von v und in direkt benachbarten Feldern befinden

Probleme:
- oft mangelnde Qualität, da die Kräfte weiter entfernt liegender Knoten vernachlässigt werden.
- Laufzeit kann immer noch quadratisch sein (wenn Knoten gehäuft in einem Feld sind)

6.5.2 Zerlegungsbaum Methoden

Idee:
- Aufbau eines Zerlegungsbaumes (Quad-Tree), der auf der momentanen Verteilung der Knoten in der Zeichenebene beruht
- Generiere zu jedem Knoten des Quad Trees Informationen die für die approximative Kräfteberechnung benötigt werden und speichere diese im Quad Tree ab
- Approximative Berechnung der abstoßenden Kräfte aufgrund der im Quad Tree gespeicherten Informationen

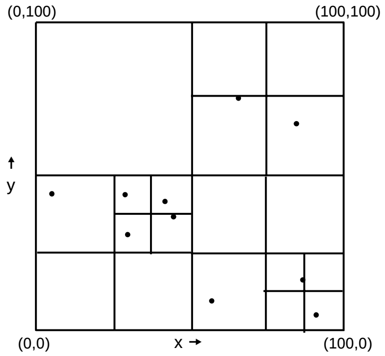
Exkurs: Datenstruktur Quad Tree

Point-Region Quadtrees

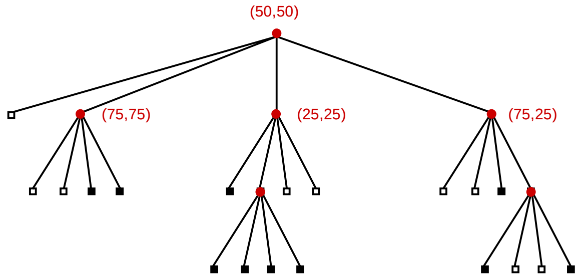
- Repräsentation von Punkten in einem k-dim. Bereich
- Rekursive Teilung eines quadratischen Bereiches in vier gleich große Quadranten, STOP falls Block nur 0 oder nur 1 Punkt enthält
- Jedem Feld wird ein Knoten in einem Suchbaum mit Maximal- Grad 4 zugeordnet
- Jedes Kind eines Knotens repräsentiert Quadranten (NW,NE,SW,SE)
- Blätter ← Aufteilung nicht weiter notwendig
- Blätter sind entweder ``weiß´´ (falls kein Punkt enthalten ist) oder ``schwarz´´ (sonst), innere Knoten sind ``grau´´

Point-Region Quadtrees: Beispiel

N=10 Punkte



PR Quadtree für Beispiel



Aufbau eines PR Quadtrees

Top-Down Aufbau:

- Starte mit Feld B, das alle Knoten enthält
- Sei v_B der zugehörige Knoten im Suchbaum
- Falls B mehr als einen Knoten enthält, dann
  - erzeuge 4 Kinder von v_B im Suchbaum
  - weise jedem Kind-Feld B^i alle Knoten aus B zu, welche in B^i enthalten sind
  - entferne leere Kind-Felder $v_B{}^i$ im Baum

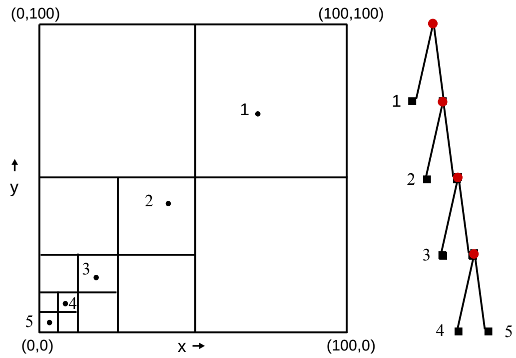
Alternativ: Insert-Aufbau:

- Starte mit leerem Feld B und füge iterativ die Knoten ein
- Einfügen geht ähnlich wie bei binären Suchbäumen: suche das richtige Feld, Suche endet an Blatt, füge ein.

Laufzeit ?

Baum hat Tiefe N

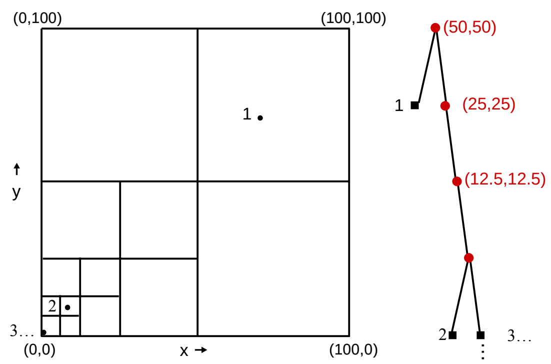
→ Laufzeiten beider Aufbau-Algorithmen: $O(N^2)$



Laufzeit: noch schlimmer ?

Baum hat Tiefe $d \gg N$

Lösung: Reduced Quad Trees (z.B. in HJ)



Zerlegungsbaum Methode

von Tunkelang 98, Quickley & Eades 2001

hier: PR Quad Trees

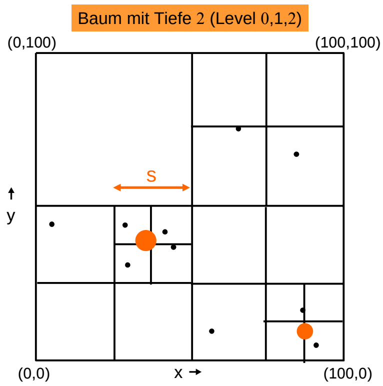
1. Aufbau eines Zerlegungsbaumes Quad-Tree T
2. In einem Knoten des Quad Trees wird die Gesamtladung aller in seinem Feld enthaltenen Knoten gespeichert. Diese Clusterladung wird auf dem gemeinsamen Schwerpunkt der Ladungen platziert.
3. Berechne Knoten-Cluster Interaktionen:
   - Berechne Kraft, die von nahe gelegenen Partikeln auf v ausgeübt wird direkt
   - Approximiere die Kraft, die von weiter entfernten Partikeln auf v ausgeübt wird durch Kräfte der Clusterladungen auf geeigneten Ebenen des Quad Trees

Zerlegungsbaum Methode Bsp

N=10 Punkte

Baum mit Tiefe 2 (Level 0,1,2)

Level 2:

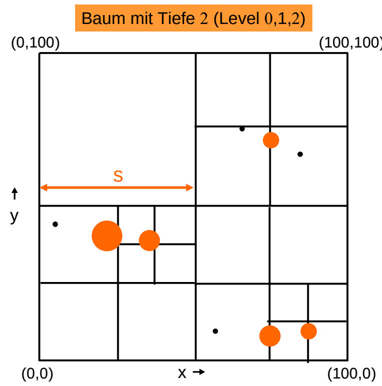


Zerlegungsbaum Methode Bsp

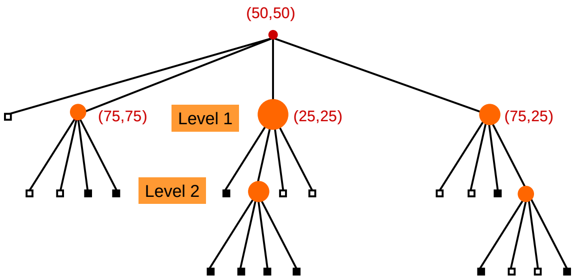
N=10 Punkte

Baum mit Tiefe 2 (Level 0,1,2)

Level 1:



PR Quadtree für Beispiel



Algorithmus

von Tunkelang 98, Quickley & Eades 2001

1. Wähle einen Genauigkeitsparameter t
2. Für jedes $v \in V(G)$ mache folgendes:
3. Setze $F_{rep}(v)=0$
4. Starte mit dem Wurzelknoten w von T
5. Ist v in dem an w wurzelnden Teilbaum enthalten, so gehe zu jedem der maximal vier Kindknoten von w
6. Ist v nicht in dem an w wurzelnden Teilbaum enthalten
   - und w ein Blattknoten, so addiere zu $F_{rep}(v)$ die Kraft, welche c auf v ausübt
   - und $s/d<t$, so addiere die Kraft, welche c auf v ausübt
   - Ansonsten gehe zu den maximal vier Kindern von w und verfahre analog.

Bemerkungen

- Laufzeit: $O(|V| \log |V|)$ für uniforme Verteilungen
- Bei uniformen Verteilungen und ca. v=10.000 Knoten ist diese Methode um Faktor 100 schneller als das exakte Verfahren (bei vernachlässigbarem Fehler durch Approximation)
- i.A. aber quadratisch: $O(|V|)^2$
- Es gibt Zerlegungsbaum-Methoden, die eine Laufzeit von $O(|V| \log |V|)$ garantieren (z.B. Reduced Quad Tree, 2D Trees). Diese sind jedoch algorithmisch komplexer [Aluru, Prabhu, Gustafson 1994, Callahan, Kosaraju 1995, Hachul, Jünger 2005]

6.5.3 Multilevel Methoden

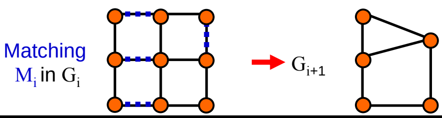
Idee:

1. Erzeuge zu $G=(V,E)=:G_0$ eine Familie von Graphen $\{G_0,G_1,\ldots,G_k\}$ mit absteigender Größe
2. Zeichne den Graphen G_k mit einem Kräfteverfahren
3. Für alle $i = k,\ldots,1$:
   – erzeuge ein Anfangslayout für G_{i-1} aus dem Layout von G_i
   – wende das Kräfteverfahren auf G_{i-1} an → Layout für G_{i-1}

Multilevel Methode von Walshaw

Idee: basierend auf maximalen Matchings

1. **Erzeuge Familie von Graphen:** Für i=0 bis k-1:
   – Finde ein maximales Matching M_i in G_i. Schrumpfe jede Kante aus M_i zu einem Knoten → G_{i+1}
   – Merke dabei zu jedem Knoten aus G_{i+1} den dazugehörigen Subgraphen aus G_i
   – Entferne Multikanten und Schleifen aus G_{i+1}



Multilevel Methode von Walshaw

2. Zeichnen von G_i

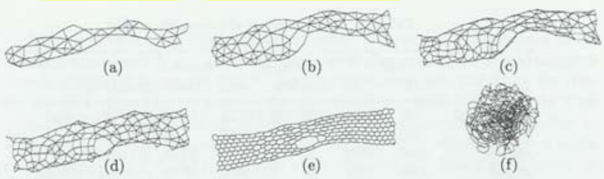
- FR zum Zeichnen von G_i zusammen mit der Gittermethode („some modifications" zur Beschleunigung)
- FR benutzte $R=2l$ (l natürliche Federlänge) für Abstand, ab dem die abstoßenden Kräfte ignoriert werden; hier: R abhängig von Feinheitsgrad i: $R_i = 2(i+1)k_i$ (wird kleiner)
- $\mathcal{E}$ (Displacement Vektor) auch abhängig von i, maximal 66 Iterationen

3. Anfangslayout für G_{i-1}

- Walshaw setzt die beiden zu einer Matchingkante assoziierten Knoten auf die Position des dazugehörigen Clusterknotens im Layout von G_i.

Beispiel aus Walshaw

Layout von G_4 | Initial-Layout von G_2 | Layout von G_2 nach erster Iteration



Final Layout von G_2 | Final Layout von G_0 | Gleiches Kräfteverfahren ohne Multilevel

Bemerkungen

- Laufzeitanalyse:
  - Matching und Coarsening: $O(|V_i|+|E_i|)$
  - Laufzeit wird dominiert durch Kräfteverfahren
  - Best Case: $O(|V|)$
  - Worst Case: $(|V|(|V|^2+|E|))$
- Bei praktischen Instanzen ist dieser Ansatz oft sehr schnell: optimale Zeichnungen für $|V|$ = 10.000 in weniger als einer Minute
- Vorteil gegenüber klassichen Methoden: deutlich bessere und schnellere Layouts aufgrund guter Initiallösungen für Kräfteverfahren

Layout Beispiel aus Walshaw

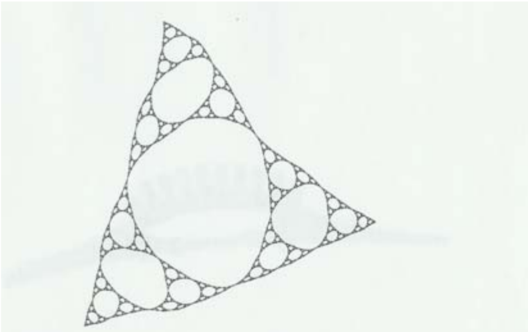


Fig. 5. The layout of sierpinski10 computed with the multilevel placement algorithm

Layput Beispiel aus Walshaw

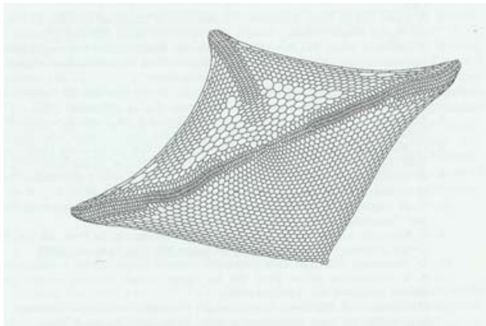


Fig. 3. The layout of 4970 computed with the multilevel placement algorithm

Alternativen für Schritt 1

- Maximal independent set filtration [Gajer, Goodrich & Kobourov]
- k-center [Harel & Koren]
- Subgraphenzerlegung (Solarsysteme) [Hachul & Jünger]

Diplomarbeit

6.5.4 Algebraische Verfahren

- **ACE von Koren et al:**
- Def. quadratisches Optimierungsproblem dessen Minimum beim Eigenvektor angenommen wird, der zum kleinsten positiven Eigenwert von L (Laplace-Matrix) gehört
- in 2 Dimensionen: Bestimme die zwei Eigenvektoren von L, die zu den kleinsten Eigenwerten gehören
- Für die Bestimmung benutze *algebraische multigrid Algorithmen:* ähnlich wie multilevel: löse zuerst Probleme im niederdim. Raum, gehe dann schrittweise höher….

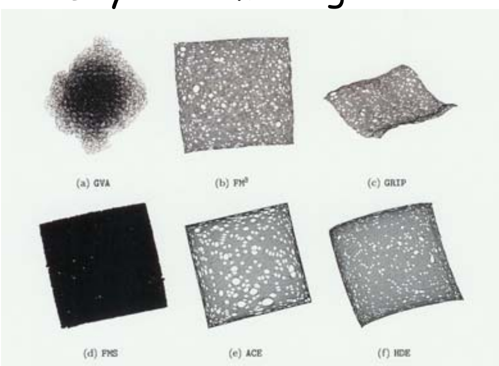
Algebraische Verfahren ff

- **HDE von Harel und Koren:**
- Generiere Einbettung im hochdimensionalen Raum:
- approximiere k-center Problem, k=50
- Projeziere diese Zeichnung nach 2D: Kovarianz Matrix, Eigenvektoren

6.5.5 Experimentelle Evaluation in HJ

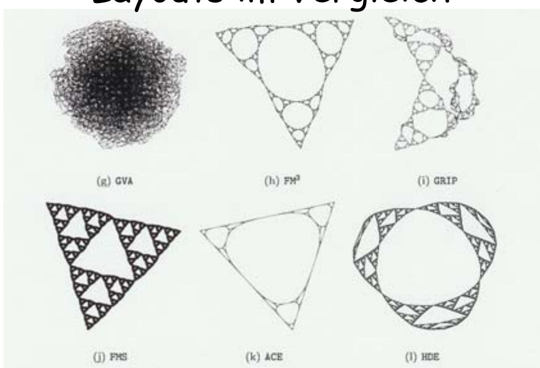
- GVA: Fruchterman & Reingold mit Gitter
- GRIP: Gajer & Kobourov: multilevel, maximum independent set filtration
- FMS: fast multi-scale method von Harel & Koren, multilevel, k-center
- FM3: force-directed multilevel, Hachul & Jünger
- ACE: Algebraic multigrid method, Koren,Karmel & Harel
- HDE: High-dimensional embedding, Harel & Koren

Layouts im Vergleich



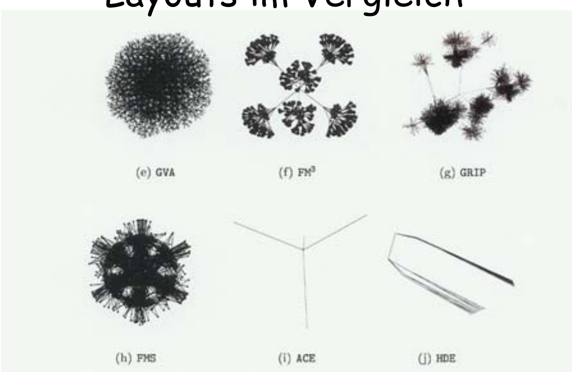
rnd_grid_100

Layouts im Vergleich



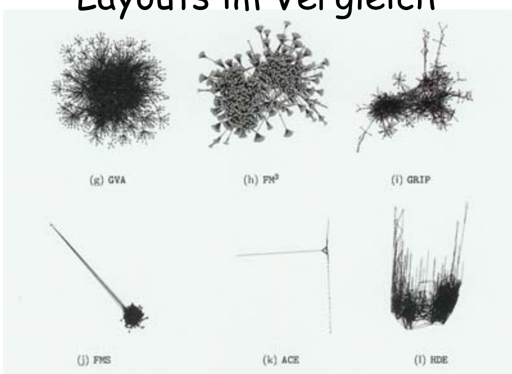
Sierpinski_08

Layouts im Vergleich



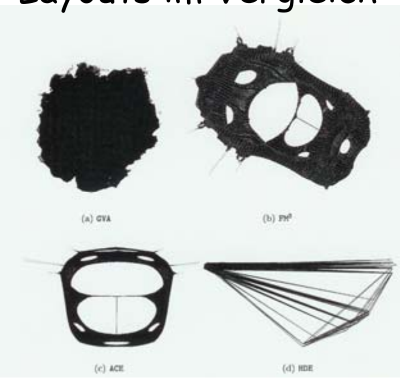
tree_06_05

Layouts im Vergleich



Soziale Netzwerke: Esslingen

Layouts im Vergleich



besstk_31

Abschluss-Bemerkungen

Themen, die wir nicht behandelt haben:

- **Cluster Planarität:** Erweiterung der Planarität auf c-Planarität: OFFENES PROBLEM: bisher nur Lösungen für Spezialgraphen bekannt
- **Cluster Planarisierung:** Praktisches Verfahren hierfür? Dasselbe für **Level-Planarität**
- **Upward Planarisierung:** Test auf Upward-Planarität ist NP-schwierig, aber für 1-sink Graphen polynomiell; trotzdem: Planarisierungsansatz?
- **Simultaneous Embeddings:** Layout von 2 oder mehr Graphen gleichzeitig, so dass z.B. Kreuzungen nur innerhalb des gleichen Graphen zählen

Abschluss-Bemerkungen

Themen, die wir nicht behandelt haben:

- **Dynamisches Zeichnen, Mental Map:** Änderung einer Zeichnung (Einfügen neuer Knoten/Kanten), wobei die Mental Map erhalten werden sollte.
- **Navigation:** Collapse, Uncollapse von Teilgraphen, Rest der Zeichnung sollte sich nicht allzu sehr ändern
- **Exploration großer Graphen:** Erwandere nach und nach einen Graphen (Web-Links)
- …vieles mehr

Prüfungsmodalitäten

- Mündliche Fachprüfung:
  - Über VO 4 inkl Ü 2: 9LP
  - Anforderungen:
    - Zusammenhänge des Gebiets
    - Spezielle Fragestellungen einordnen und bearbeiten
    - (Regelmäßige aktive Mitarbeit in Übungen, u.a. mind. drei erfolgreiche Präsentationen)
    - Mündliche Prüfung: Stoff der VO und Ü (20 Min.)
    - Ein Unter-Thema streichbar (s. Themen)

Prüfungselemente

- Leistungsnachweis:
  - Über VO 4 inkl. Ü 2: 9LP
  - Anforderungen:
    - Regelmäßige aktive Mitarbeit in Übungen, u.a. mind. drei erfolgreiche Präsentationen

Themen der Vorlesung

1. Einführung*
2. Tree Drawing
3. Sugiyama: Überblick, Layering, Koordinatenzuweisung
4. Sugiyama: Kreuzungsminimierung (Heuristiken und Exakt)
5. Kreuzungszählen
6. Planaritätstest
7. SPQR-Bäume und planare Einbettungen, max. Außenfläche
8. SPQR-Bäume: ec-Constraints, optimales Kanteneinfügen
9. Planarisierungsverfahren
10. Planare Straightline Zeichenalgorithmen
11. Orthogonale planare Verfahren
12. Kräftebasierte Verfahren
13. Kräftebasierte Vefahren für große Graphen

Fachprüfung: Ein Thema streichbar; * = nicht streichbar!

Ihre Lieblingsthemen sind:

1. Einführung*
2. Tree Drawing **1**
3. Sugiyama: Überblick, Layering, Koordinatenzuweisung **2**
4. Sugiyama: Kreuzungsminimierung (Heuristiken und Exakt) **2**
5. Kreuzungszählen **2**
6. Planaritätstest **5**
7. SPQR-Bäume und planare Einbettungen, max. Außenfläche **1**
8. SPQR-Bäume: ec-Constraints, optimales Kanteneinfügen
9. Planarisierungsverfahren
10. Planare Straightline Zeichenalgorithmen **1**
11. Orthogonale planare Verfahren **1**
12. Kräftebasierte Verfahren **3**
13. Kräftebasierte Vefahren für große Graphen **3**

Diese Themen fanden Sie gar nicht schön:

Auswertungen der anonymen VO Umfrage

Noten von
1: sehr gut – 7 sehr schlecht

Anzahl der Antworten

		1	2	3	4	5	6	7	
1. Wie motivierend ist die VO?		1	8	2					
2. Wie wird der Stoff erklärt?		1	8	1	1				
3. Modalität der Übungen?		3	2	5					
4. Schwierigkeitsgrad der Übungen?	leicht				2	8	1		schwer
5. etwas mitgenommen/gelernt?		1	5	4		1			
6. nützlich für später?		1	4	4	1	1			
7. VO Bewertung insgesamt?		2	8	1					
8. UE Bewertung insgesamt?		1	2	5	2	1			

…und wie geht es weiter?:

- **SS 2008:**
  - DAP2 4VO+2UE+2PR
  - Seminar über Netzwerkalgorithmen (Flüsse), Vorbesprechung am Mi 9.4., 10:15 Uhr, Interessensbekundung bei: Markus Chimani

Bis Bald!