

Bachelorkolloquium

# Beschleunigung von kräftebasierten Graphzeichenalgorithmen mittels wohlseparierten Paardekompositionen

von Johannes Zink

# Übersicht

## 1. Grundlagen

### 1.1 Kräftebasierte Graphzeitalgorithmen

1.2 Der kräftebasierte Graphzeitalgorithmus von Fruchterman und Reingold

1.3 Wohlseparierte Paardekompositionen

## 2. Algorithmus

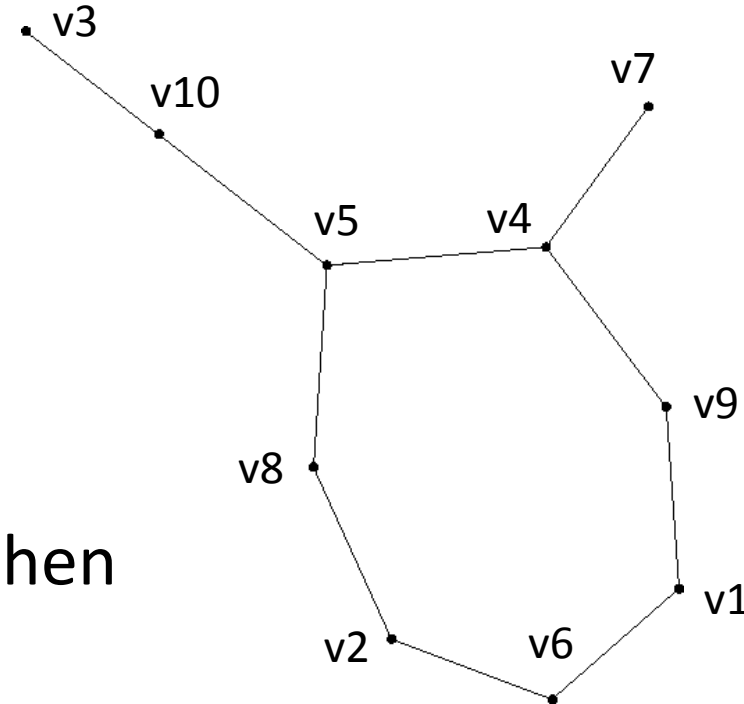
## 3. Ergebnisse

## 4. Zusammenfassung

# Kräftebasierte Graphzeichenalgorithmen

- Gegeben: Ein beliebiger (ungerichteter) Graph

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10
v1	0	0	0	0	0	1	0	0	1	0
v2	0	0	0	0	0	1	0	1	0	0
v3	0	0	0	0	0	0	0	0	0	1
v4	0	0	0	0	1	0	1	0	1	0
v5	0	0	0	1	0	0	0	1	0	1
v6	1	1	0	0	0	0	0	0	0	0
v7	0	0	0	1	0	0	0	0	0	0
v8	0	1	0	0	1	0	0	0	0	0
v9	1	0	0	1	0	0	0	0	0	0
v10	0	0	1	0	1	0	0	0	0	0

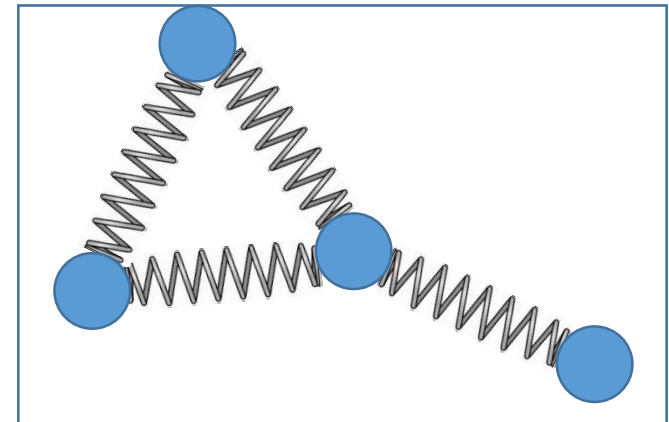
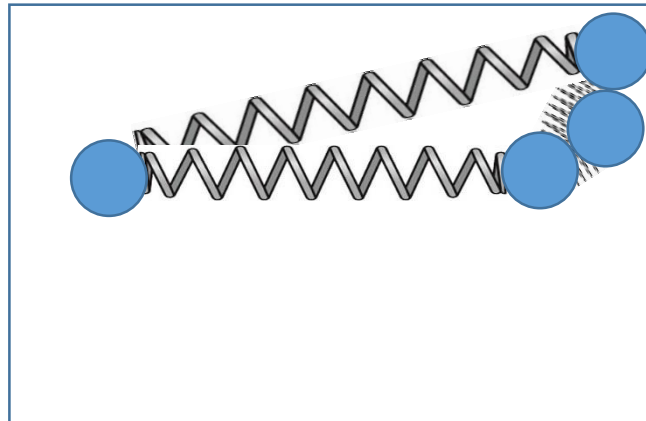


1	0		
2	0		
3	0		
4	0		
5	0		
6	0		
7	0		
8	0		
9	0		
10	0		
#			
1	0	4	9
2	0	9	1
3	0	1	6
4	0	7	4
5	0	2	8
6	0	8	5
7	0	5	10
8	0	10	3
9	0	4	5
10	0	6	2

- Gesucht: Eine Zeichnung zu diesem Graphen

# Kräftebasierte Graphzeichenalgorithmen

- Mögliche Lösung:
  - Gib eine Zeichenfläche vor
  - Weise jedem Knoten einen zufälligen Punkt in dieser Zeichenfläche zu
  - Modelliere jede Kante als Feder zwischen den Punkten ihrer inzidenten Knoten
  - Lass zwischen den Punkten der Knoten abstoßende Kräfte wirken
  - Lass dieses modellierte System einige Iterationen laufen
  - Nimm die so erhaltene Abbildung Knoten→Punkte als Grundlage für die gewünschte Zeichnung



# Kräftebasierte Graphzeichenalgorithmen

- In dieser Form erstmals 1984 von Eades eingeführt
- Zahlreiche verbesserte und veränderte Algorithmen von verschiedenen Autoren folgten
- Noch sehr nahe am Algorithmus von Eades: Der Algorithmus von Fruchterman und Reingold (1991)

## Unterschiede:

- Andere Formeln zur Kraftberechnung
- Verwendung eines Rahmens (frame) zur Begrenzung der Zeichenfläche
- Verwendung einer sehr einfachen Form von *Simulated Annealing* (Positionsänderungen werden mit steigender Iterationszahl verkleinert)

# Übersicht

## 1. Grundlagen

1.1 Kräftebasierte Graphzeichenalgorithmen

**1.2 Der kräftebasierte Graphzeichenalgorithmus von Fruchterman und Reingold**

1.3 Wohlseparierte Paardekompositionen

2. Algorithmus

3. Ergebnisse

4. Zusammenfassung

# Der Algorithmus von Fruchterman und Reingold

Eingabe: Ein Graph  $G = (V, E)$ , evtl. eine Zeichenfläche

Ausgabe: Eine Zuordnung  $V \rightarrow \mathbb{R}^2$

```

O(|V|) → 1  foreach  $v \in V$  do
           2    Weise  $v$  den zufällige Punkt  $p_v$  innerhalb der Zeichenfläche zu
O(1) → 3     $k = c\sqrt{A/|V|}$  //  $k$  ist der opt. Knotenabstand,  $c$  eine Konst.,  $A$  die Größe der Zeichenfläche
O(?) → 4    while Terminierungskriterium nicht erfüllt do
[O(|V|^2)]
O(|V|^2) → 5    //Abstoßende Kräfte
           6    foreach  $v \in V$  do
           7       $v.\text{disp} = \vec{0}$  //Das ist der iterationsinterne Verschiebungsvektor
           8      foreach  $u \in V$  do
           9        if  $v \neq u$  then
           9           $v.\text{disp} = v.\text{disp} + \frac{\overrightarrow{p_u p_v}}{\|\overrightarrow{p_v p_u}\|} \cdot \frac{k^2}{\|\overrightarrow{p_v p_u}\|}$ 
//Anziehende Kräfte
O(|E|) → 10   foreach  $\{u, v\} \in E$  do
           11      $u.\text{disp} = u.\text{disp} + \frac{\overrightarrow{p_u p_v}}{\|\overrightarrow{p_u p_v}\|} \cdot \frac{\|\overrightarrow{p_u p_v}\|^2}{k}$ 
           12      $v.\text{disp} = v.\text{disp} + \frac{\overrightarrow{p_v p_u}}{\|\overrightarrow{p_v p_u}\|} \cdot \frac{\|\overrightarrow{p_v p_u}\|^2}{k}$ 
//Neue Positionen berechnen
O(|V|) → 13   foreach  $v \in V$  do
           14      $p_v = p_v + v.\text{disp}$  //Beachte hierbei jedoch Cooling-Schema und Rahmen der Zeichenfläche
           15   return Errechnete Zuordnung  $V \rightarrow \mathbb{R}^2$ 

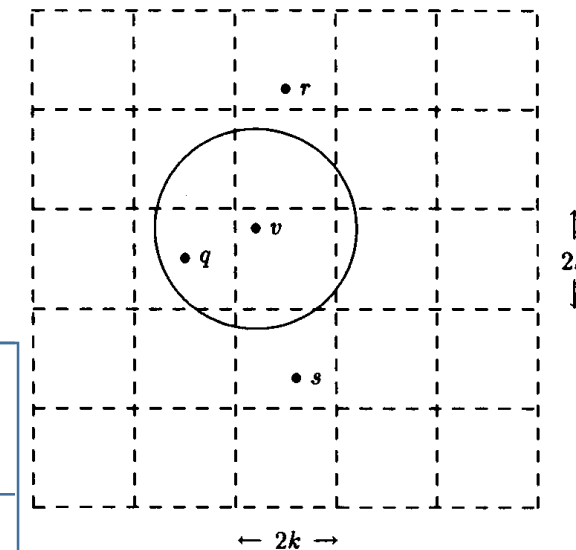
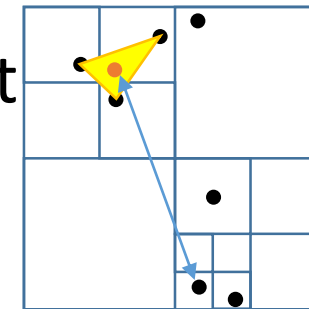
```

**Teurer Schritt  
bzgl. der  
Laufzeit!**

# Der Algorithmus von Fruchterman und Reingold

Möglichkeiten die quadratische Laufzeit (bzgl. der Knotenanzahl) beim Berechnen der abstoßenden Kräfte zu reduzieren:

- Grid-Variante von Fruchterman und Reingold
  - Bei gleichmäßiger Verteilung der Knoten Laufzeit in  $O(|V|)$
  - Knoten außerhalb des Bereichs werden ignoriert
- Verwendung des Ansatzes nach Barnes und Hut
  - Quadtree über Punktmenge aufbauen
  - Laufzeit in der Praxis in  $O(|V| \log|V| + |E|)$
- Neuer Ansatz: Verwendung von wohlseparierten Paardekompositionen (well-separated pair decompositions)
  - Laufzeit in  $O(|V| \log|V| + |E|)$





# Übersicht

## 1. Grundlagen

1.1 Kräftebasierte Graphzeichenalgorithmen

1.2 Der kräftebasierte Graphzeichenalgorithmus von Fruchterman und Reingold

### 1.3 Wohlseparierte Paardekompositionen

2. Algorithmus

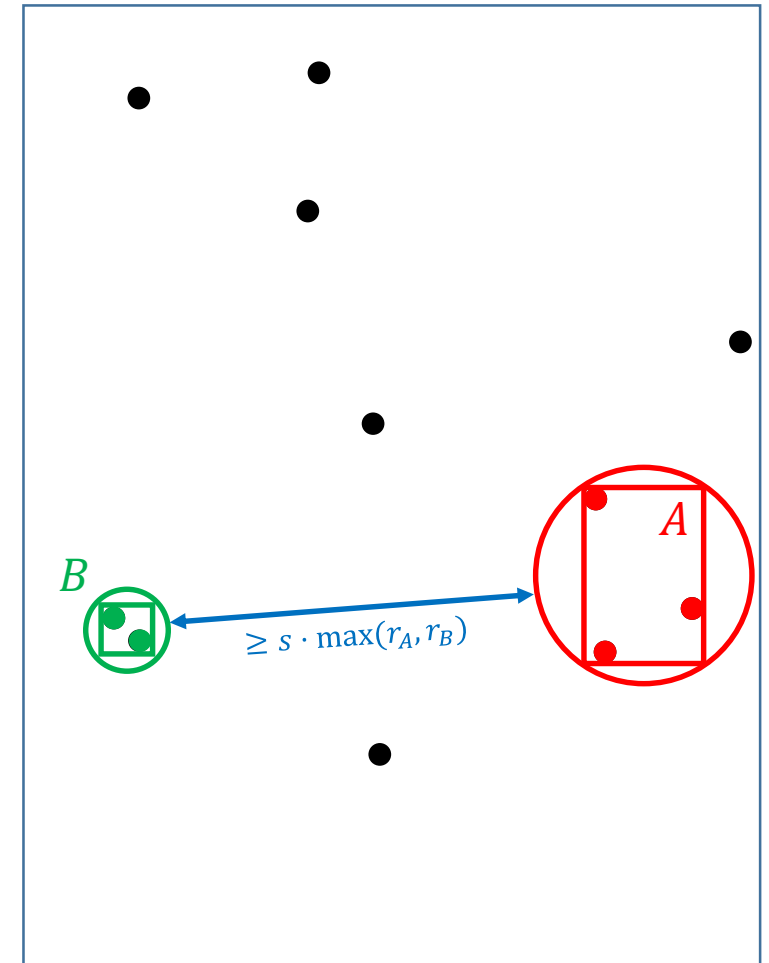
3. Ergebnisse

4. Zusammenfassung

# Wohlseparierte Paardekompositionen

(well-separated pair decompositions)

- Gegeben: Eine Punktmenge  $P$  (in der Ebene)
- Benötigt: Ein  $s > 0$
- Der engste Kreis um die Bounding-Box einer Punktmenge  $A$  heie  $C_A$  und habe Radius  $r_A$
- Zwei Punkt Mengen  $A$  und  $B$  bilden ein *wohlsepariertes Paar* bezglich  $s$ , wenn der Abstand zwischen  $C_A$  und  $C_B$  mindestens  $s \cdot \max(r_A, r_B)$  betrgt
- Eine *wohlseparierte Paardekomposition* (WSPD) bezglich  $s$  ist ein Folge von wohlseparierten Paaren  $\{A_1, B_1\}, \{A_2, B_2\}, \dots, \{A_m, B_m\}$  bzgl.  $s$ , sodass es fr alle  $p, q \in P$ ,  $p \neq q$  genau ein  $i$  gibt, sodass  $p \in A_i$  und  $q \in B_i$  oder  $p \in B_i$  und  $q \in A_i$



# Wohlseparierte Paardekompositionen

(well-separated pair decompositions)

- Eine WSPD zu einer zweidimensionalen Punktmenge  $P$  mit  $n$  Punkten und einem  $s > 0$  kann in  $O(s^2n)$  Zeit und mit  $O(s^2n)$  Paaren u. a. aus einem Split-Tree oder einem komprimierten Quadtree berechnet werden
- Jede Punktmenge, die eine der zwei Punkt Mengen eines Paares aus der WSPD ist, entspricht dabei einem Knoten des Baumes, auf dem die WSPD basiert
- Ein Split-Tree zu einer Punktmenge  $P$  mit  $n$  Punkten kann in  $O(n \log n)$  Zeit berechnet werden. Dieser hat eine Größe in  $O(n)$  und hängt nicht von  $s$  ab

# Übersicht

## 1. Grundlagen

1.1 Kräftebasierte Graphzeitalgorithmen

1.2 Der kräftebasierte Graphzeitalgorithmus von Fruchterman und Reingold

1.3 Wohlseparierte Paardekompositionen

## 2. Algorithmus

## 3. Ergebnisse

## 4. Zusammenfassung

# Der veränderte Algorithmus (FR mit WSPD)

Eingabe: Ein Graph  $G = (V, E)$ , evtl. eine Zeichenfläche, ein  $s > 0$

Ausgabe: Eine Zuordnung  $V \rightarrow \mathbb{R}^2$

$O( V )$	1	<b>foreach</b> $v \in V$ <b>do</b>
$O(1)$	2	Weise $v$ den zufällige Punkt $p_v$ innerhalb der Zeichenfläche zu
$O(?)$ bzw. $O( V \log V + E )$	3	$k = c\sqrt{A/ V }$ // $k$ ist der opt. Knotenabstand, $c$ eine Konst., $A$ die Größe der Zeichenfläche
$O( V \log V )$	4	<b>while</b> Terminierungskriterium nicht erfüllt <b>do</b>
$O( V )$	5	Berechne einen Split-Tree $st$ zu den aktuellen Positionen der Knoten
$O( V )$	6	Berechne eine WSPD $wspd$ bezüglich $s$ aus $st$
$O( V )$	7	Berechne die Schwerpunkte aller Split-Tree-Knoten aus $st$
$O( V )$	8	<b>foreach</b> $\{A, B\} \in wspd$ <b>do</b> // Abstoßende Kräfte
	9	$c_A := A.schwerpunkt, c_B := B.schwerpunkt$
	10	$A.abstoßendeKraft = A.abstoßendeKraft +  B  \frac{\overrightarrow{c_B c_A}}{\ \overrightarrow{c_A c_B}\ } \cdot \frac{k^2}{\ \overrightarrow{c_A c_B}\ }$
	11	$B.abstoßendeKraft = B.abstoßendeKraft +  A  \frac{\overrightarrow{c_A c_B}}{\ \overrightarrow{c_B c_A}\ } \cdot \frac{k^2}{\ \overrightarrow{c_B c_A}\ }$
$O( V )$	12	Propagiere berechneten Kräfte durch $st$ von den Split-Tree-Knoten zu $v.disp$ für alle $v \in V$
$O( E )$	13	<b>foreach</b> $\{u, v\} \in E$ <b>do</b> // Anziehende Kräfte
	14	$u.disp = u.disp + \frac{\overrightarrow{p_u p_v}}{\ \overrightarrow{p_u p_v}\ } \cdot \frac{\ \overrightarrow{p_u p_v}\ ^2}{k}$
	15	$v.disp = v.disp + \frac{\overrightarrow{p_v p_u}}{\ \overrightarrow{p_v p_u}\ } \cdot \frac{\ \overrightarrow{p_v p_u}\ ^2}{k}$
$O( V )$	16	<b>foreach</b> $v \in V$ <b>do</b> // Neue Positionen berechnen
	17	$p_v = p_v + v.disp$ // Beachte hierbei jedoch Cooling-Schema und Rahmen der Zeichenfläche
	18	<b>return</b> Errechnete Zuordnung $V \rightarrow \mathbb{R}^2$

# Eine heuristische Beschleunigungstechnik

- Jede Iteration wird ein neuer Split-Tree und eine neue WSPD berechnet
- Möglicherweise wären diese aus der vorherigen Iteration noch gut genug für die aktuelle Iteration
- Gegen Ende des Zeichenvorgangs nur noch kleine Veränderungen der Knotenpositionen

# Eine heuristische Beschleunigungstechnik

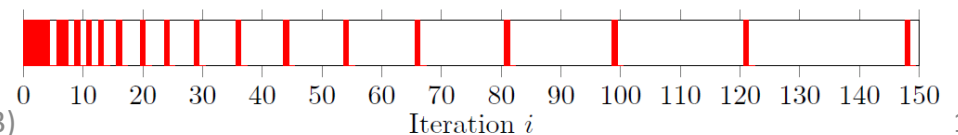
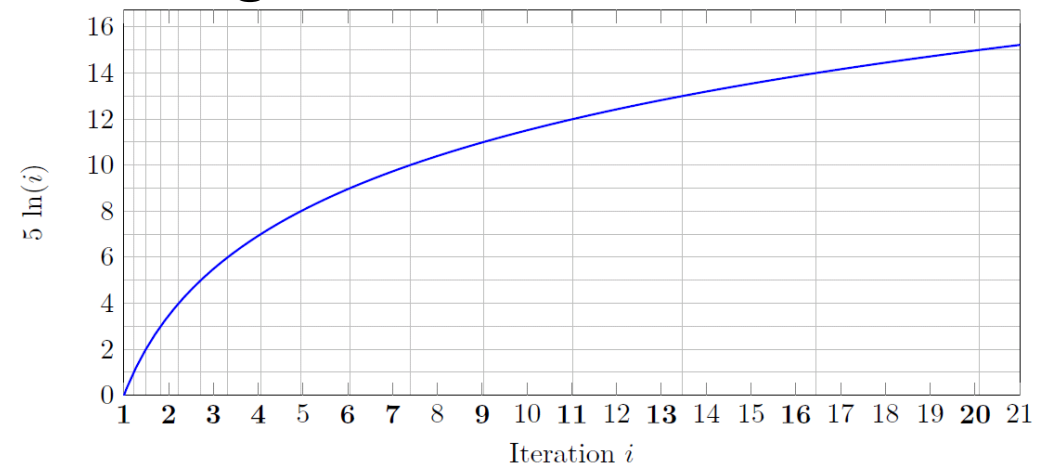
Daher:

- Berechne nur in manchen Iterationen Split-Tree und WSPD neu
- Verwende ansonsten den alten Split-Tree und die alte WSPD
- Berechne zu Beginn häufig neu und Richtung Ende immer seltener
- Mögliche Realisierung:

Berechne in den Iterationen neu,  
in denen  $\lfloor 5 \ln(i) \rfloor \neq \lfloor 5 \ln(i + 1) \rfloor$

Dabei ist  $i$  die aktuelle Iterationszahl

Statt 5 größere Werte  $\rightarrow$  öfter neu



# Übersicht

## 1. Grundlagen

1.1 Kräftebasierte Graphzeichenalgorithmen

1.2 Der kräftebasierte Graphzeichenalgorithmus von Fruchterman und Reingold

1.3 Wohlseparierte Paardekompositionen

## 2. Algorithmus

## 3. Ergebnisse

## 4. Zusammenfassung

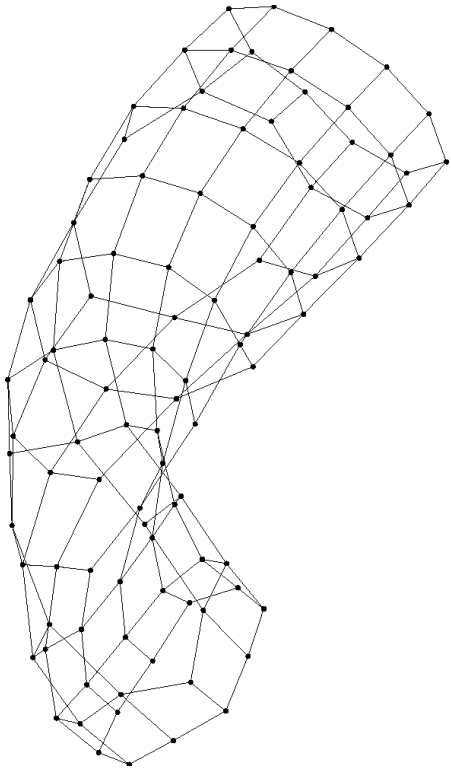


# Ergebnisse

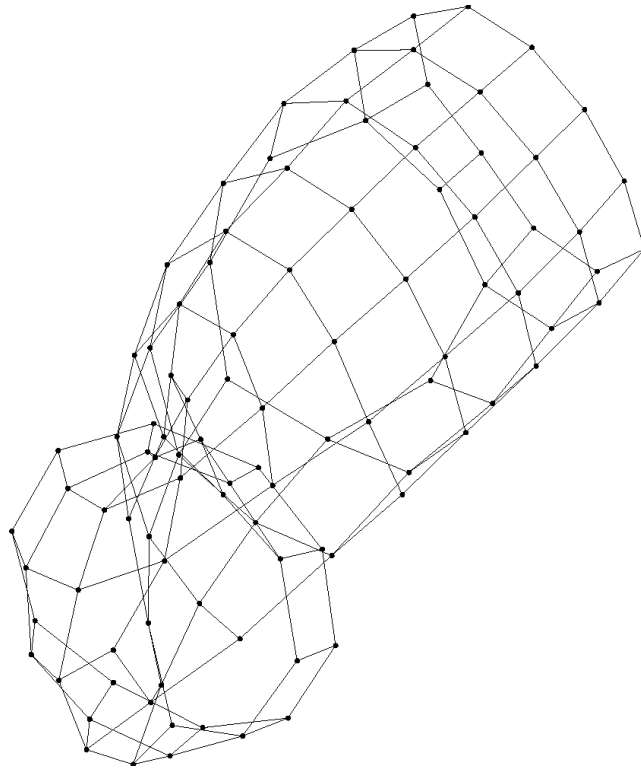
- Implementierung des Ansatzes
- Verwendet: JUNG-Bibliothek in Java
- Testgraphensätze:
  - Rome-Graphen (11.528 Graphen, 10–100 Knoten, <http://www.graphdrawing.org/download/rome-graphml.tgz>)
  - Hachul-Graphen (88 Graphen, ab 97 Knoten, [http://www.informatik.uni-koeln.de/public/gronemann/hachul\\_graphs.zip](http://www.informatik.uni-koeln.de/public/gronemann/hachul_graphs.zip))
  - Zufällige Graphen (20 Graphen, 4.565–96.855 Knoten, generiert mit dem EppsteinPowerLawGenerator aus JUNG)

# Beispiele für Zeichnungen

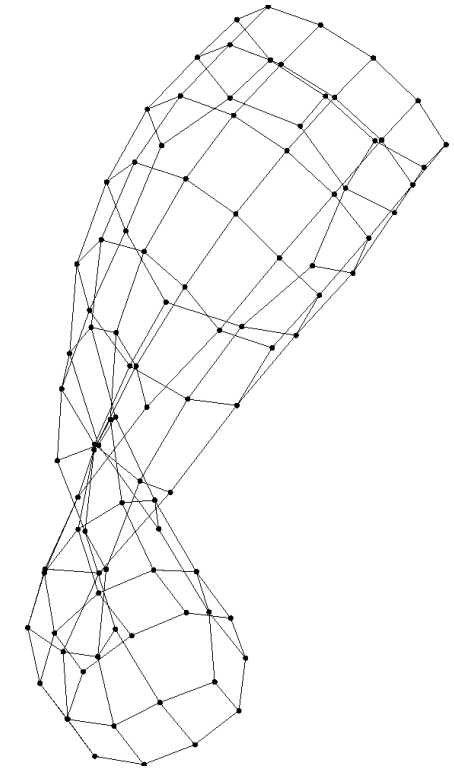
cylinder-rnd-010-010 (hachul\_graphs)



Fruchterman-Reingold  
(FR)



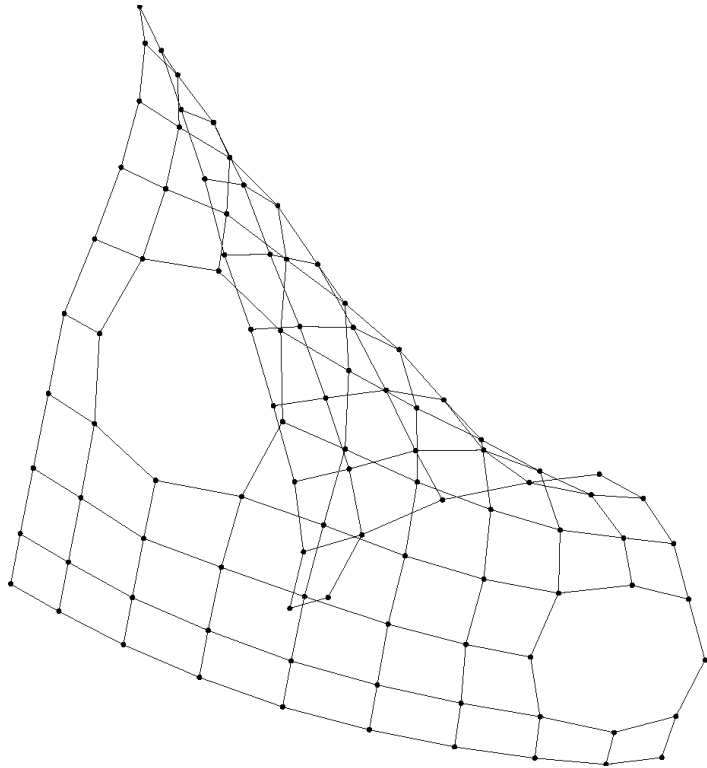
FR mit WSPD  
 $s = 1$ , neu gemäß  $5 \ln(i)$



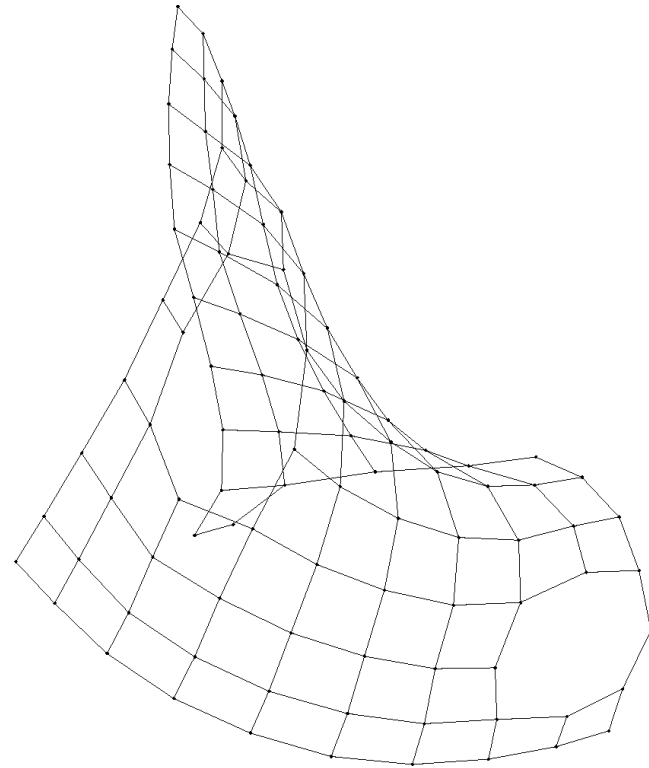
FR mit Quadtree  
(Barnes-Hut)

# Beispiele für Zeichnungen

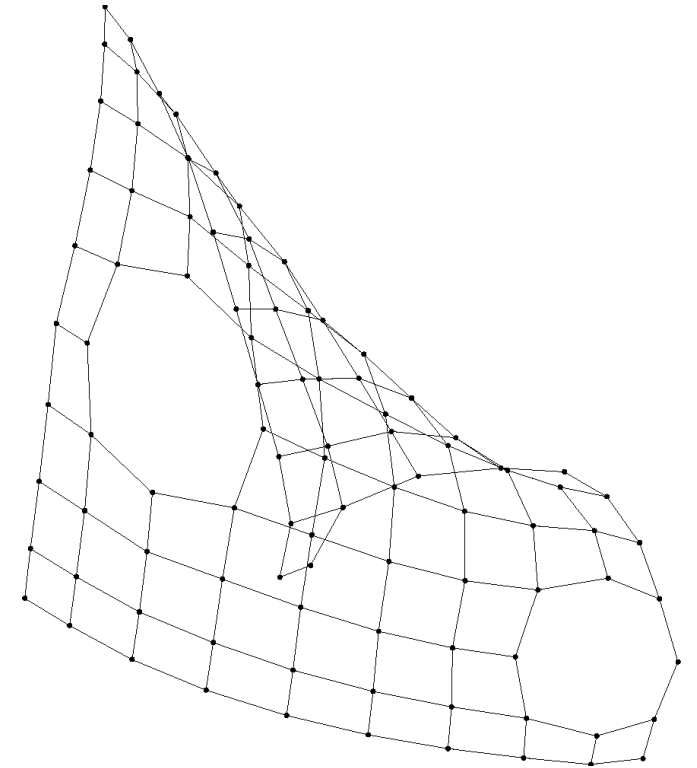
grid-rnd-010 (hachul\_graphs)



Fruchterman-Reingold  
(FR)



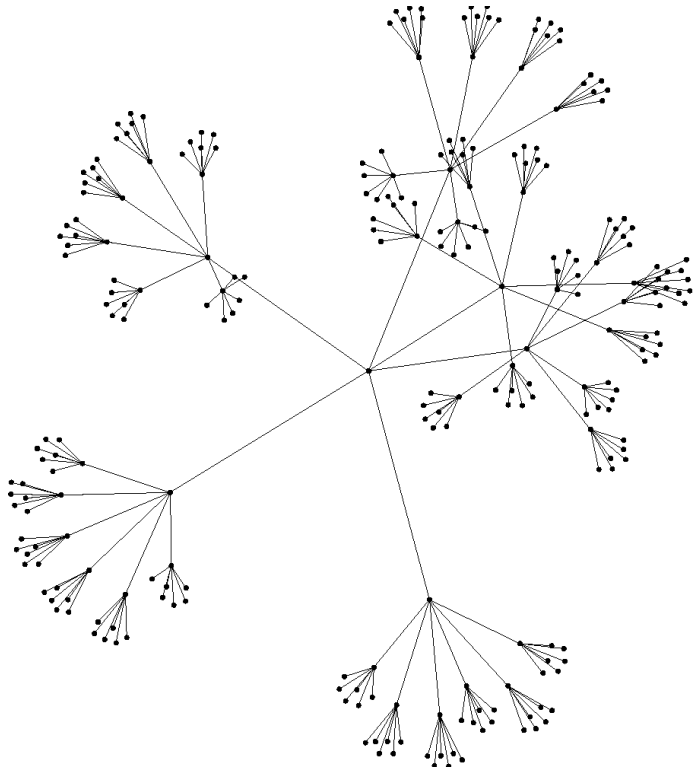
FR mit WSPD  
 $s = 1$ , neu gemäß  $5 \ln(i)$



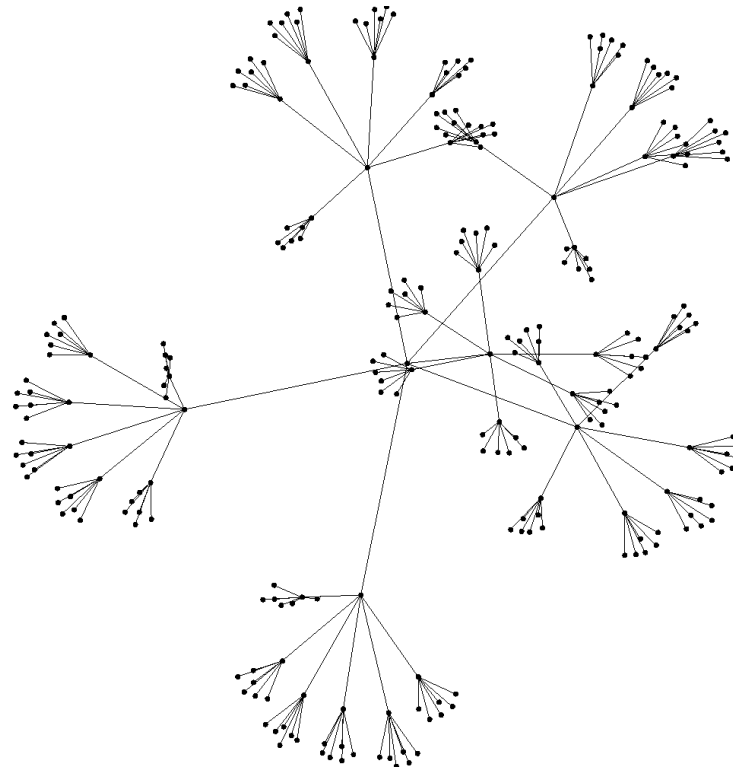
FR mit Quadtree  
(Barnes-Hut)

# Beispiele für Zeichnungen

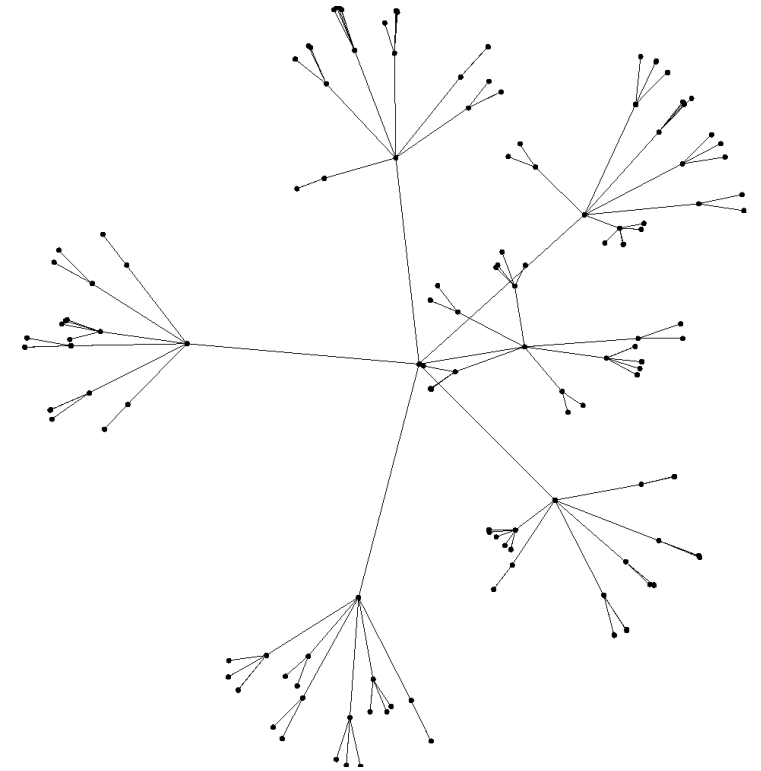
tree-06-03 (hachul\_graphs)



Fruchterman-Reingold  
(FR)



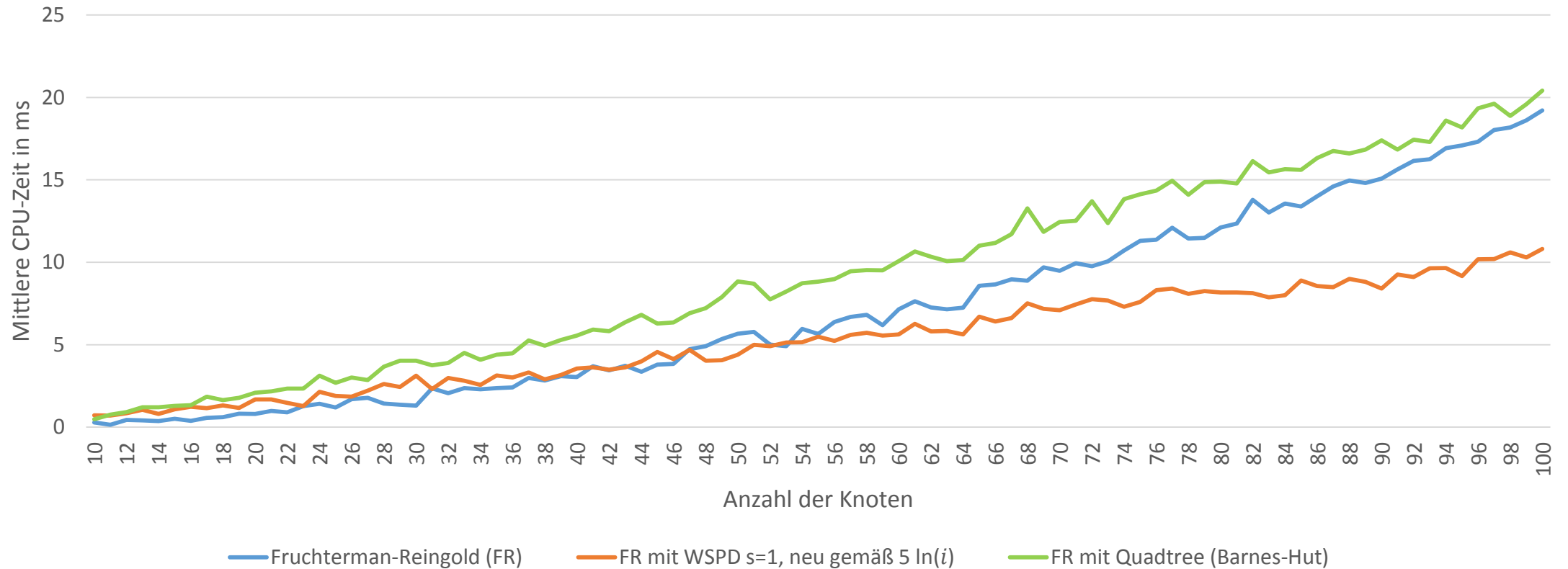
FR mit WSPD  
 $s = 1$ , neu gemäß  $5 \ln(i)$



FR mit Quadtree  
(Barnes-Hut)

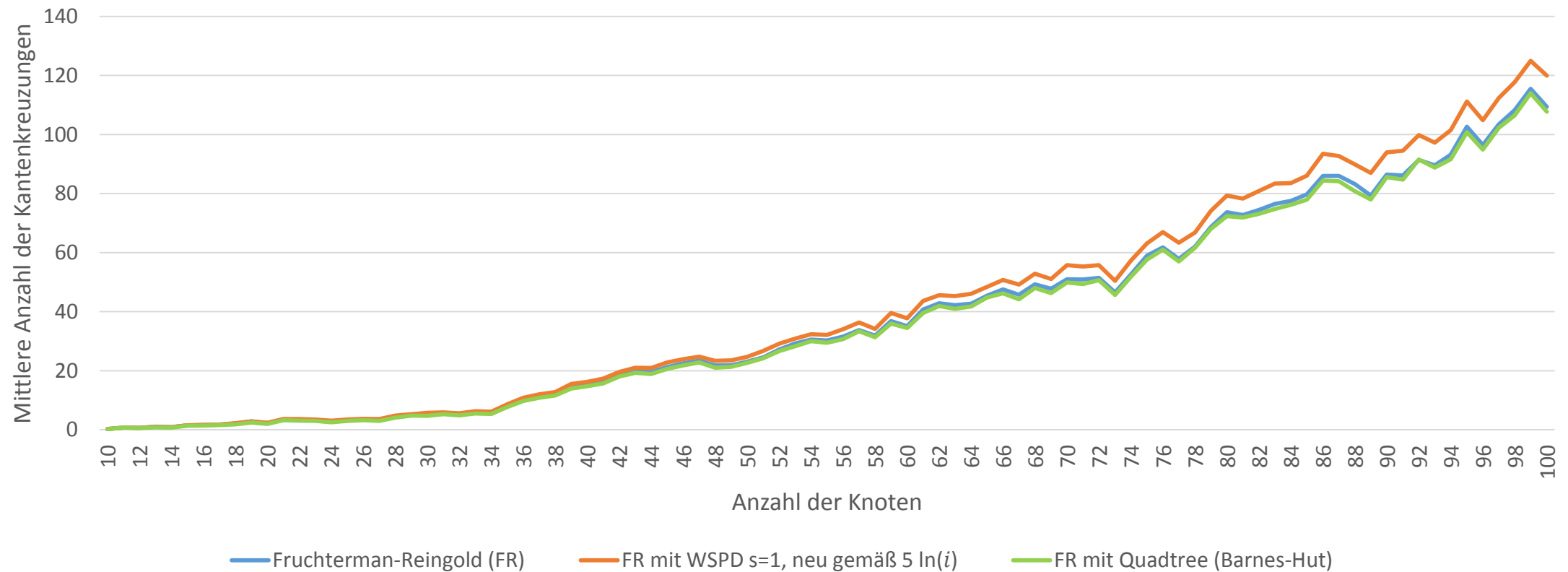
# Laufzeitvergleich

5-facher Durchlauf der 11.528 Rome-Graphen



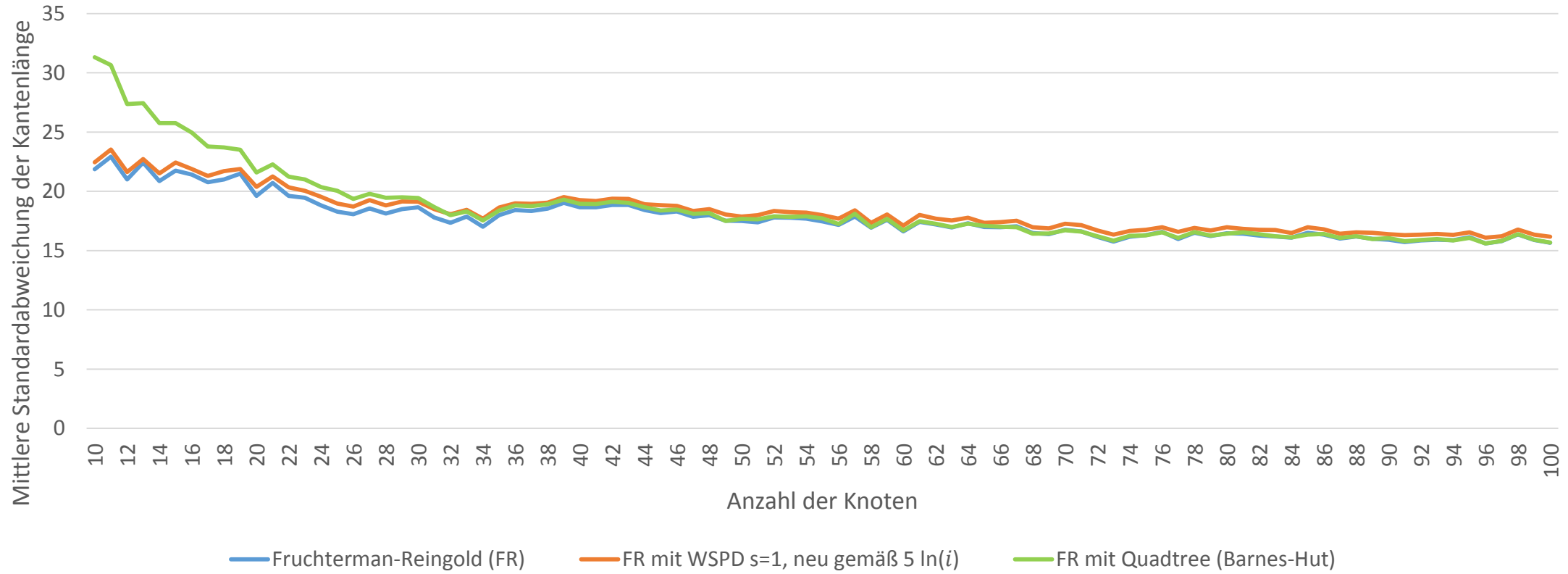
# Vergleich der Anzahl der Kantenkreuzungen

5-facher Durchlauf der 11.528 Rome-Graphen



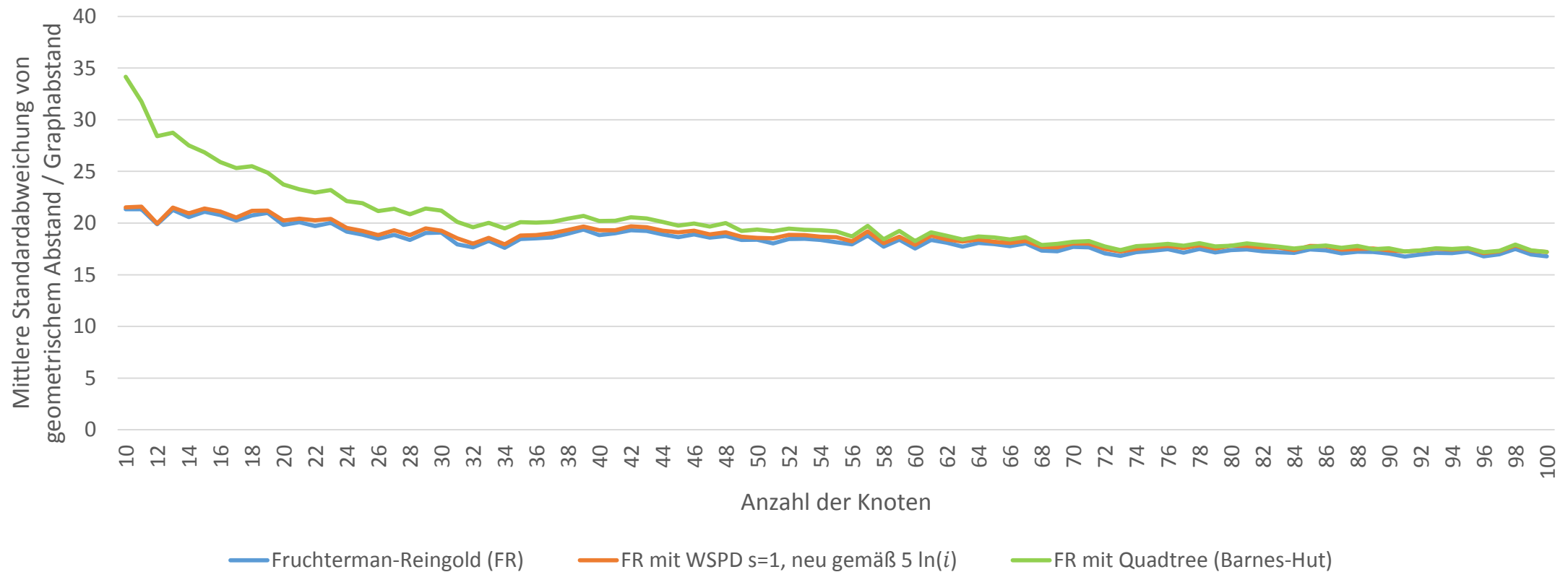
# Vergleich der Standardabweichung der Kantenlänge

5-facher Durchlauf der 11.528 Rome-Graphen



# Vergleich des Verhältnisses von geometrischem Abstand zu Graphabstand

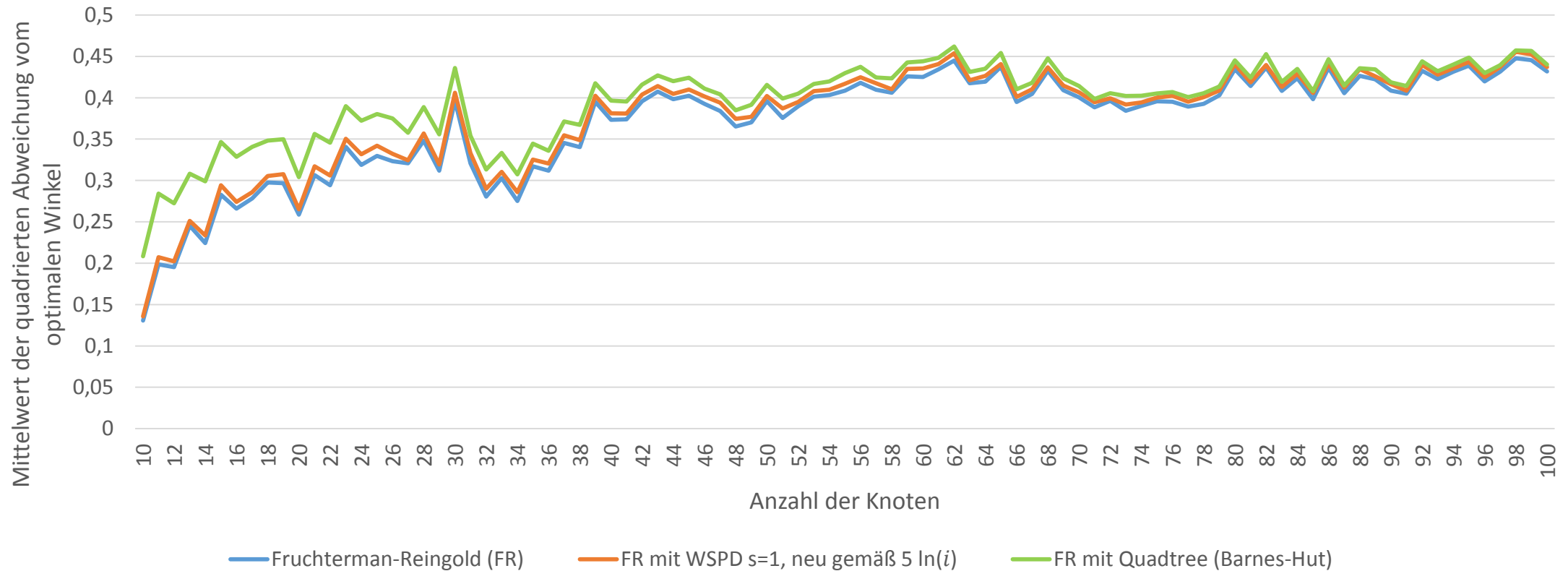
5-facher Durchlauf der 11.528 Rome-Graphen





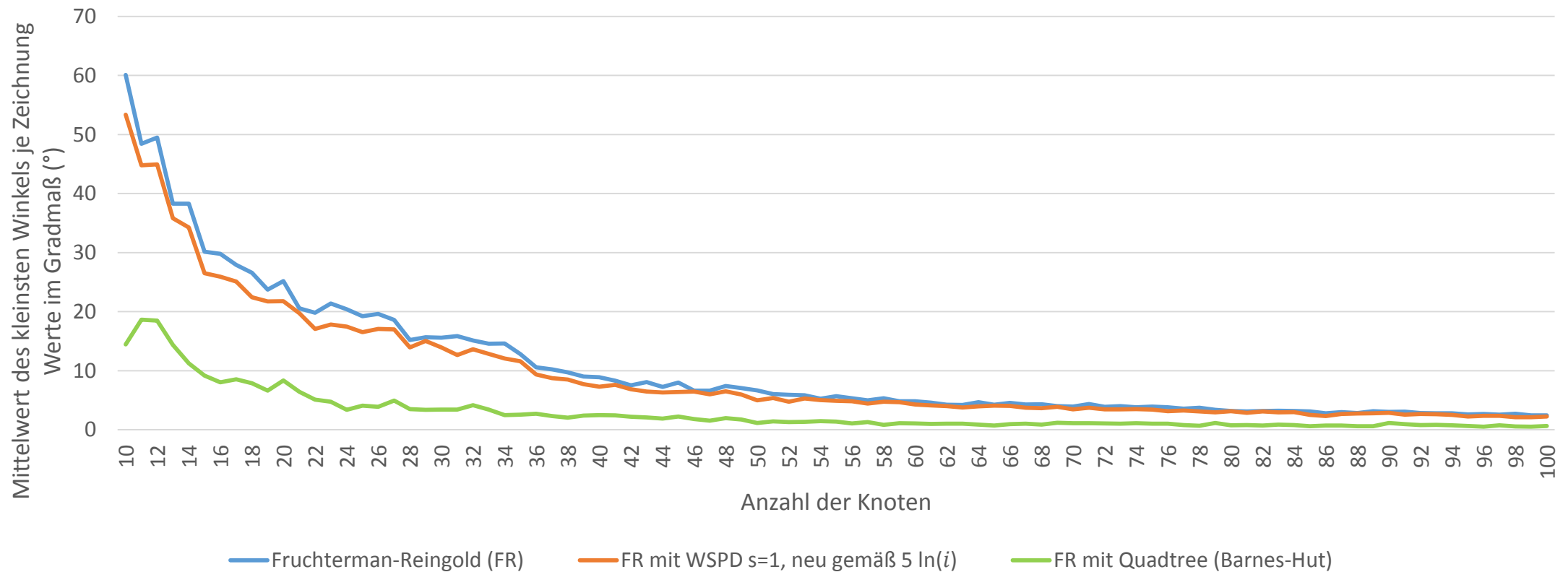
# Vergleich der Winkelauflösung (Abweichung vom optimalen Winkel)

5-facher Durchlauf der 11.528 Rome-Graphen



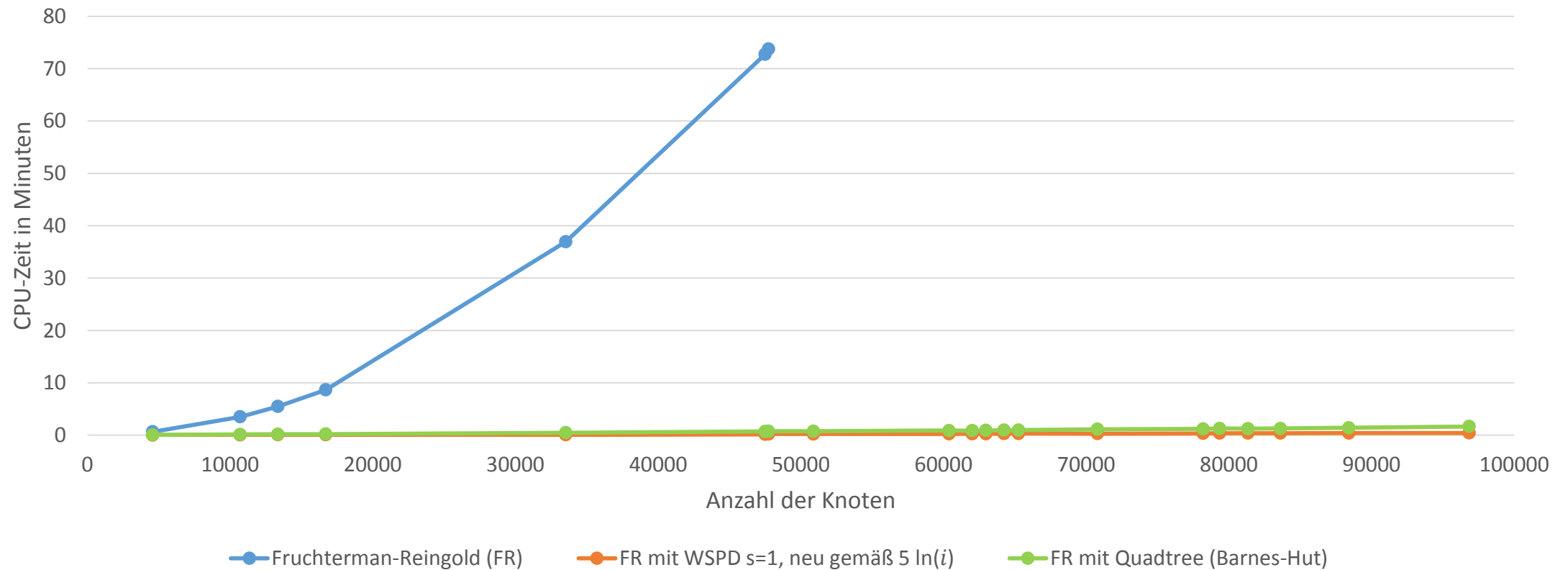
# Vergleich der Winkelauflösung (Kleinster Winkel)

5-facher Durchlauf der 11.528 Rome-Graphen



# Laufzeitvergleich

Durchlauf der 20 zufälligen Graphen mit 1.000-100.000 Knoten



# Übersicht

## 1. Grundlagen

1.1 Kräftebasierte Graphzeichenalgorithmen

1.2 Der kräftebasierte Graphzeichenalgorithmus von Fruchterman und Reingold

1.3 Wohlseparierte Paardekompositionen

## 2. Algorithmus

## 3. Ergebnisse

## 4. Zusammenfassung

# Zusammenfassung

- Mit wohlseparierten Paardekompositionen (well-separated pair decompositions) können die abstoßenden Kräfte eines klassischen kräftebasierten Graphzeichenalgorithmus approximiert werden
- Die Dauer der Berechnung dieser Kräfte sinkt dabei von  $O(|V|^2)$  auf  $O(|V| \log|V|)$
- Die Laufzeitklasse des gesamten Algorithmus sinkt dadurch bei konstanter Iterationszahl von  $O(|V|^2)$  auf  $O(|V| \log|V| + |E|)$
- Die Qualität der gelieferten Zeichnungen liegt bei den durchgeführten Testreihen annähernd bei der Qualität der Zeichnungen des Algorithmus mit exakter Berechnung