



Praktikum – Beating the Worst Case

Jean-Pierre von der Heydt und Marcus Wilhelm | 08.11.2023



Fragen

Implementierung

Algorithmen

Ergebnisse



Fragen

Implementierung

- Umgang mit Framework?
- weitere Sprachen / Tools?
- Workflow?

Algorithmen

Ergebnisse



Fragen

Implementierung

- Umgang mit Framework?
- weitere Sprachen / Tools?
- Workflow?

Algorithmen

- asymptotische Laufzeit?
- verschiedene Varianten?

Ergebnisse

Fragen

Implementierung

- Umgang mit Framework?
- weitere Sprachen / Tools?
- Workflow?

Algorithmen

- asymptotische Laufzeit?
- verschiedene Varianten?

Ergebnisse

- Netzwerke identifiziert?
- Wie schnell / gut sind die Algorithmen?
- weitere Erkenntnisse? *Eigenschaften der Netzwerke, spannende Beobachtungen, ...*



Fragen

Implementierung

- Umgang mit Framework?
- weitere Sprachen / Tools?
- Workflow?

Algorithmen

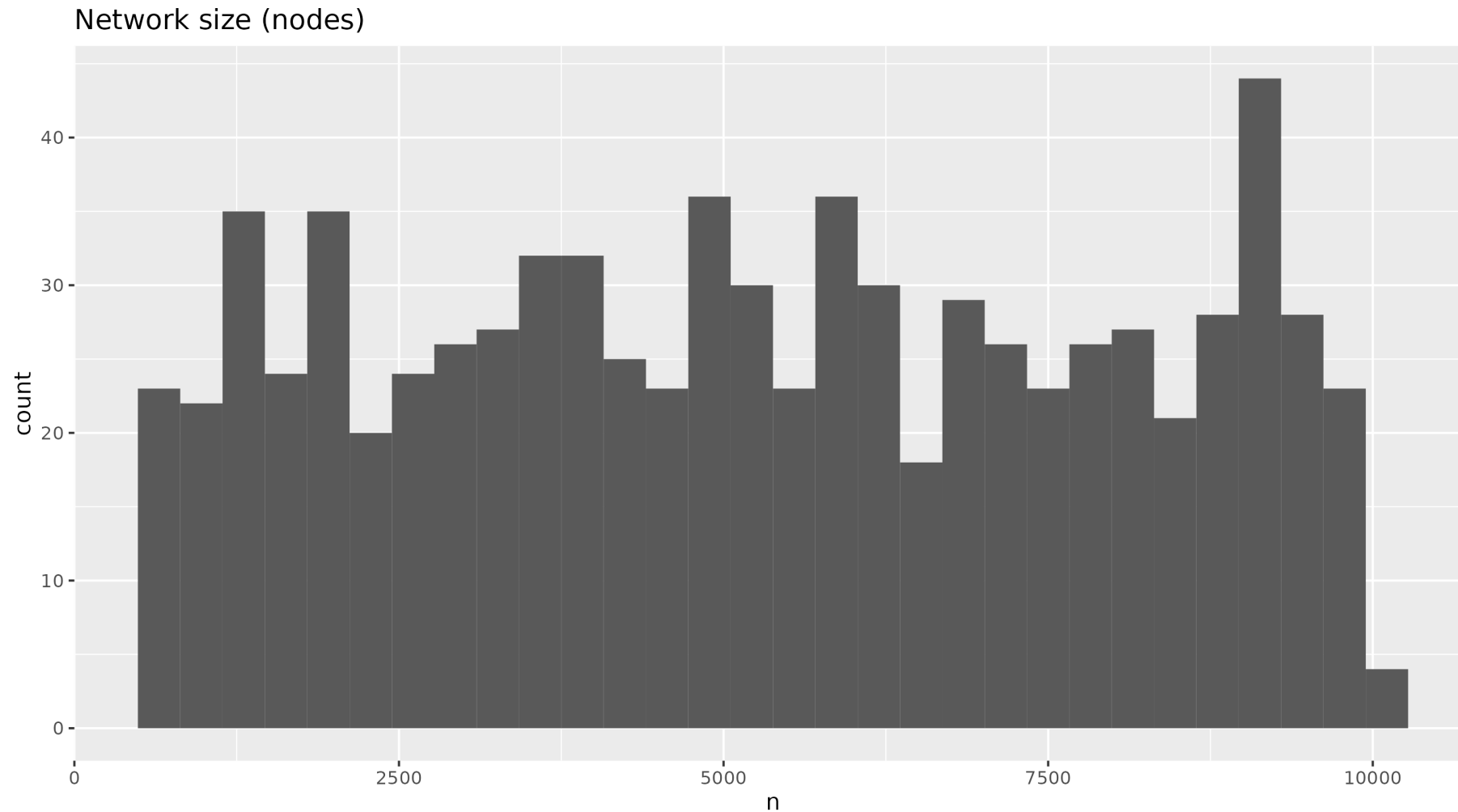
- asymptotische Laufzeit?
- verschiedene Varianten?

Ergebnisse

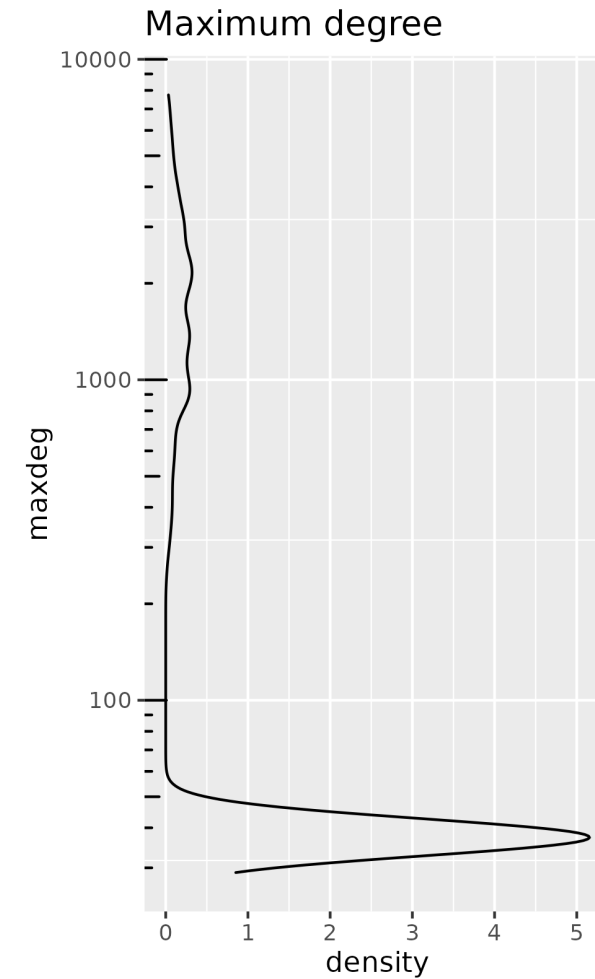
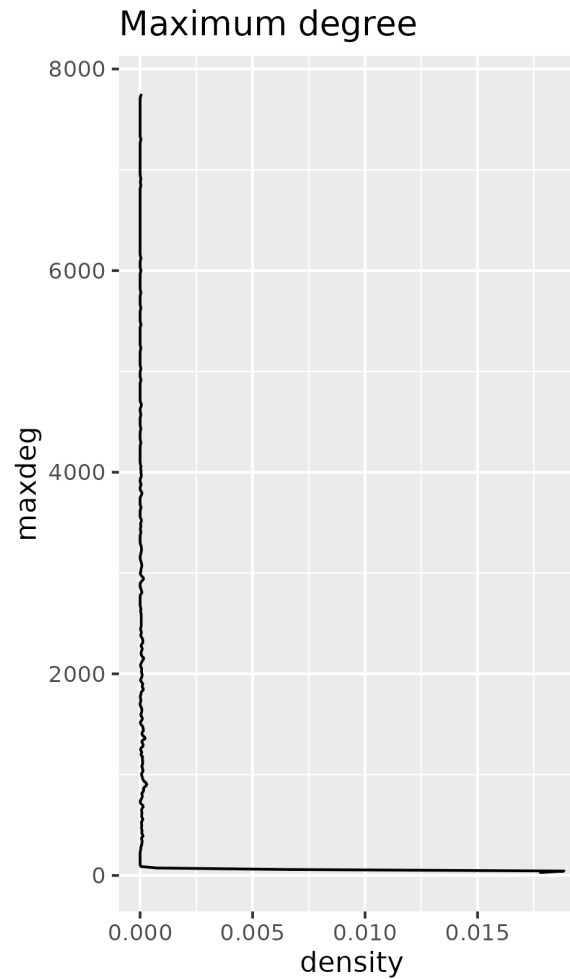
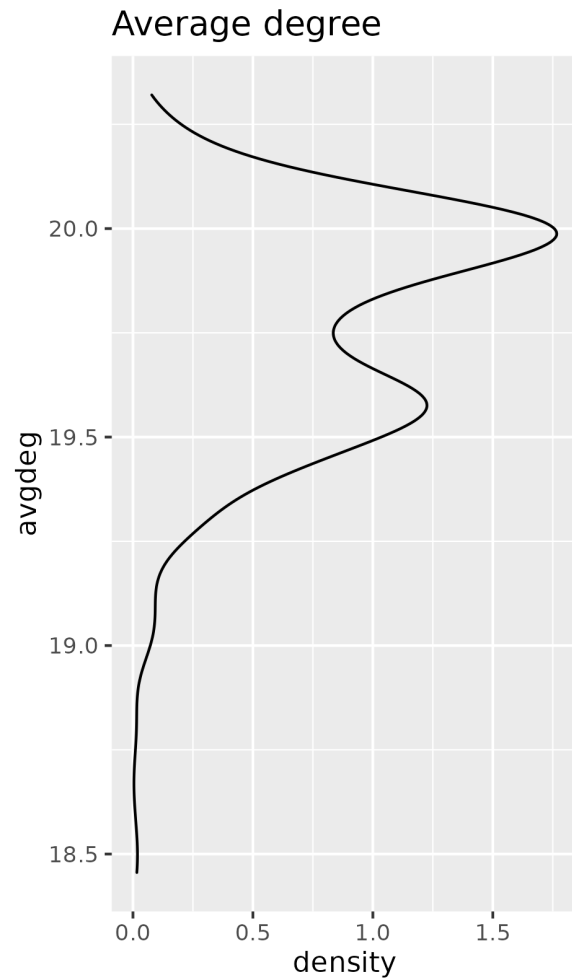
- Netzwerke identifiziert?
- Wie schnell / gut sind die Algorithmen?
- weitere Erkenntnisse? *Eigenschaften der Netzwerke, spannende Beobachtungen, ...*

Sonstiges?

Eigenschaften der Netzwerke

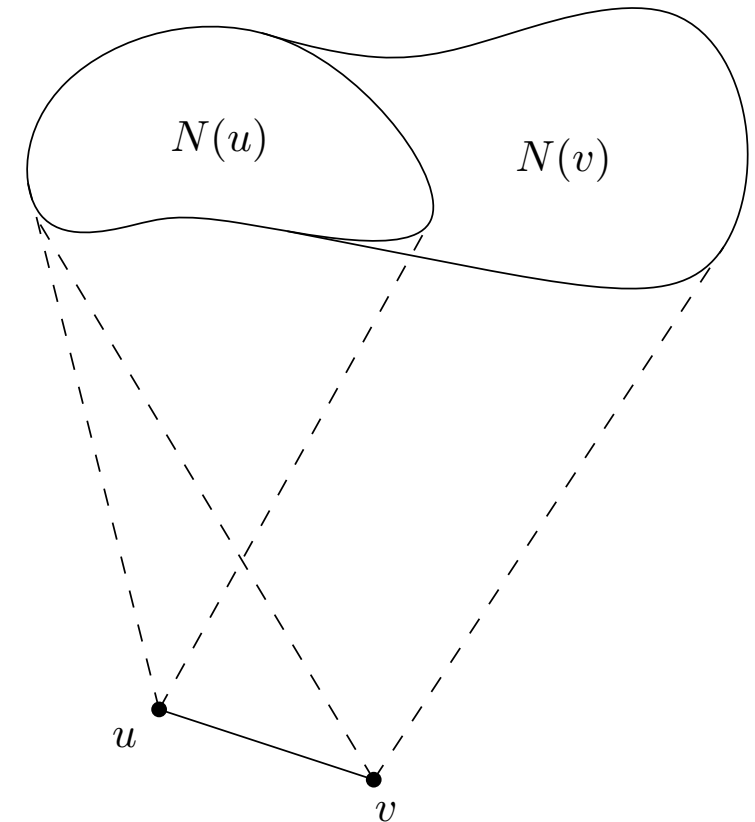


Eigenschaften der Netzwerke



Vertex Cover Dominanzregel

- Gegeben einen Graphen $G = (V, E)$
- $X \subseteq V$ ist *Vertex Cover*, falls $\forall \{u, v\} \in E: u \in X \vee v \in X$
- Für $u, v \in V$: v *dominiert* u , wenn $N[u] \subseteq N[v]$
- v ist *dominierend*, wenn ein $u \in V$ existiert, mit v dominiert u
- **Dominanzregel:** Ein dominierender Knoten kann ins Vertex Cover aufgenommen werden



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare

Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare

Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?
 - Sortierte Listen, Hashtabelle, Zeile aus Adjazenzmatrix



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?
 - Sortierte Listen, Hashtabelle, Zeile aus Adjazenzmatrix
- Wielöscht man den Knoten u aus dem Graphen?



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?
 - Sortierte Listen, Hashtabelle, Zeile aus Adjazenzmatrix
- Wielöscht man den Knoten u aus dem Graphen?
 - Markiere Knoten nur als gelöscht



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?
 - Sortierte Listen, Hashtabelle, Zeile aus Adjazenzmatrix
- Wielöscht man den Knoten u aus dem Graphen?
 - Markiere Knoten nur als gelöscht
- Wiederholt einzelnen Knotenlöschen $\rightarrow \mathcal{O}(nm)$ Iterationen



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?
 - Sortierte Listen, Hashtabelle, Zeile aus Adjazenzmatrix
- Wielöscht man den Knoten u aus dem Graphen?
 - Markiere Knoten nur als gelöscht
- Wiederholt einzelnen Knoten löschen $\rightarrow \mathcal{O}(nm)$ Iterationen
 - Lösche alle dominierenden Knoten in einem Schritt



Vertex Cover Dominanzregel

- Jedes Knotenpaar überprüfen $\rightarrow \mathcal{O}(n^2)$ Paare
- Nur Nachbarn überprüfen $\rightarrow \mathcal{O}(m)$ Paare
- Wie überprüfen wir $N[v] \subseteq N[u]$?
 - Sortierte Listen, Hashtabelle, Zeile aus Adjazenzmatrix
- Wielöscht man den Knoten u aus dem Graphen?
 - Markiere Knoten nur als gelöscht
- Wiederholt einzelnen Knoten löschen $\rightarrow \mathcal{O}(nm)$ Iterationen
 - Lösche alle dominierenden Knoten in einem Schritt
 - Zeige: a und b sind dominierend in $G \Rightarrow b$ ist dominierend in $G - a$



Vertex Cover Dominanzregel

Algorithm 1: Vertex Cover Reduction

```
while  $\exists u \in V$  dominating do  
  for  $\{a, b\} \in E$  do  
    if  $N[a] \subseteq N[b]$  then  
      mark  $b$  as dominating  
    end  
  end  
  mark all dominating nodes as deleted  
end  
mark all isolated vertices as deleted
```



Vertex Cover Dominanzregel

Algorithm 1: Vertex Cover Reduction

```

while  $\exists u \in V$  dominating do
  | for  $\{a, b\} \in E$  do
  | | if  $N[a] \subseteq N[b]$  then
  | | | mark  $b$  as dominating
  | | end
  | end
  | mark all dominating nodes as deleted
end
  mark all isolated vertices as deleted
  
```

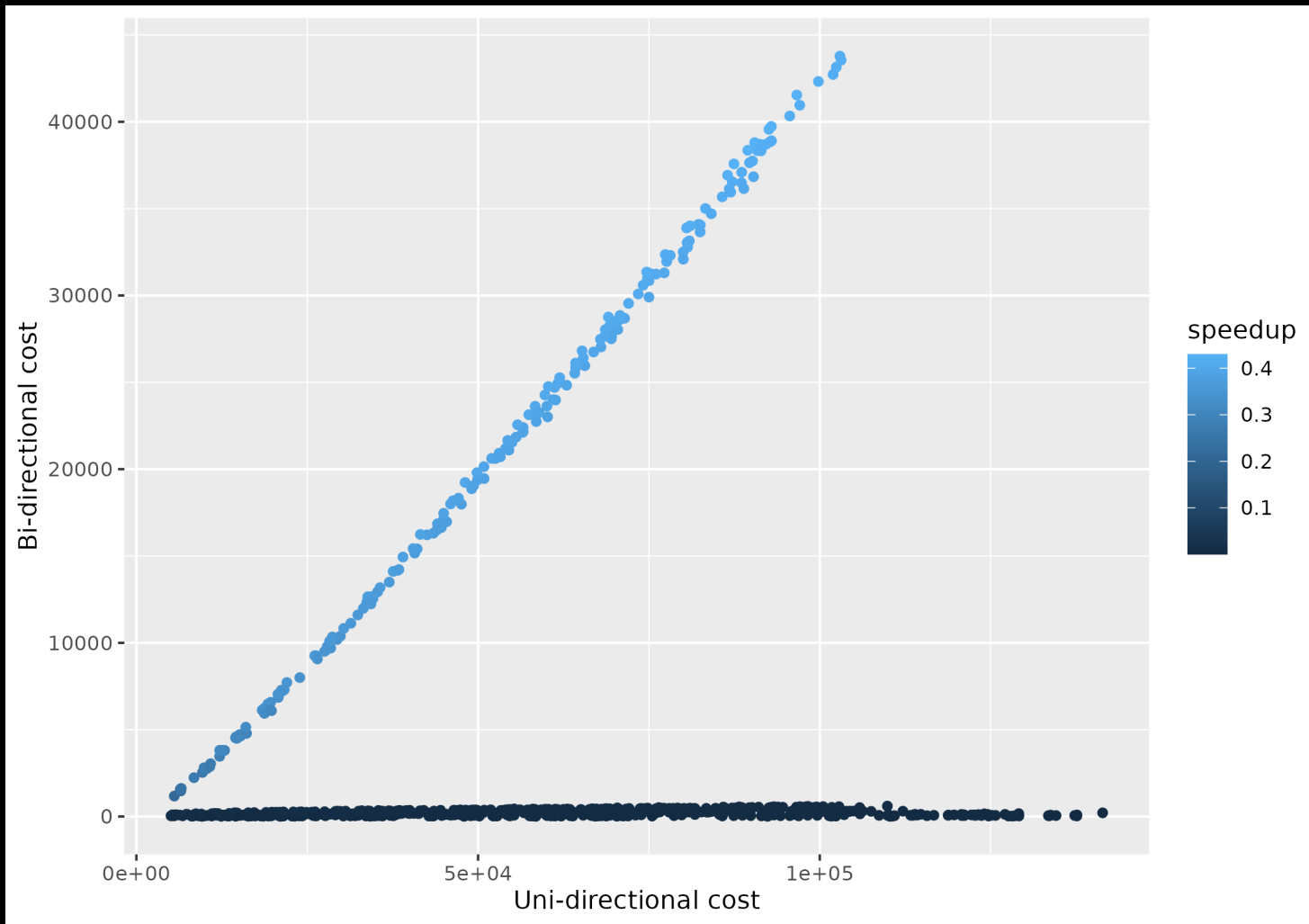
Nur für nicht gelöschte a, b

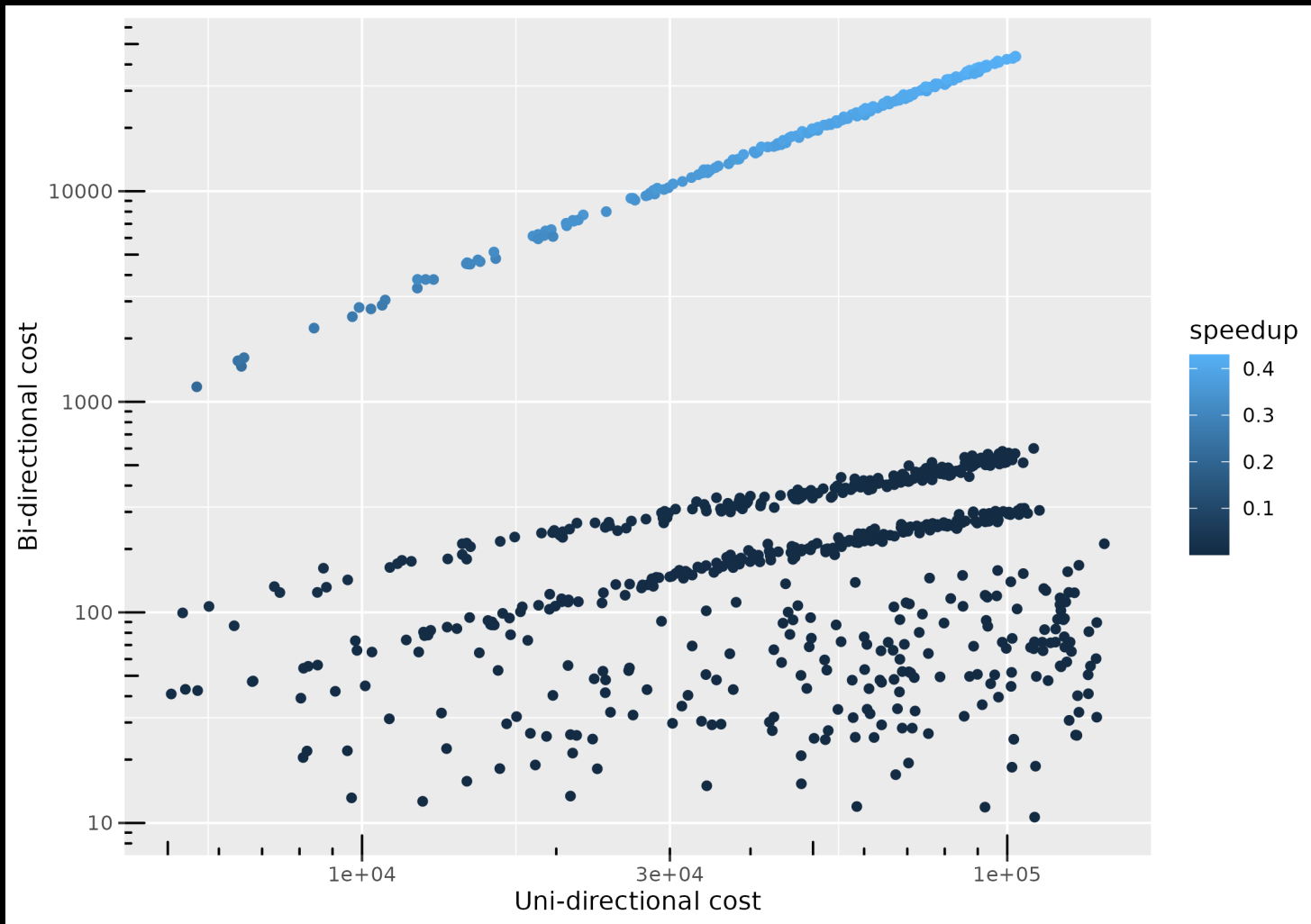
Sortierte Listen, Hashtabelle,
 $\mathcal{O}(n)$ Array

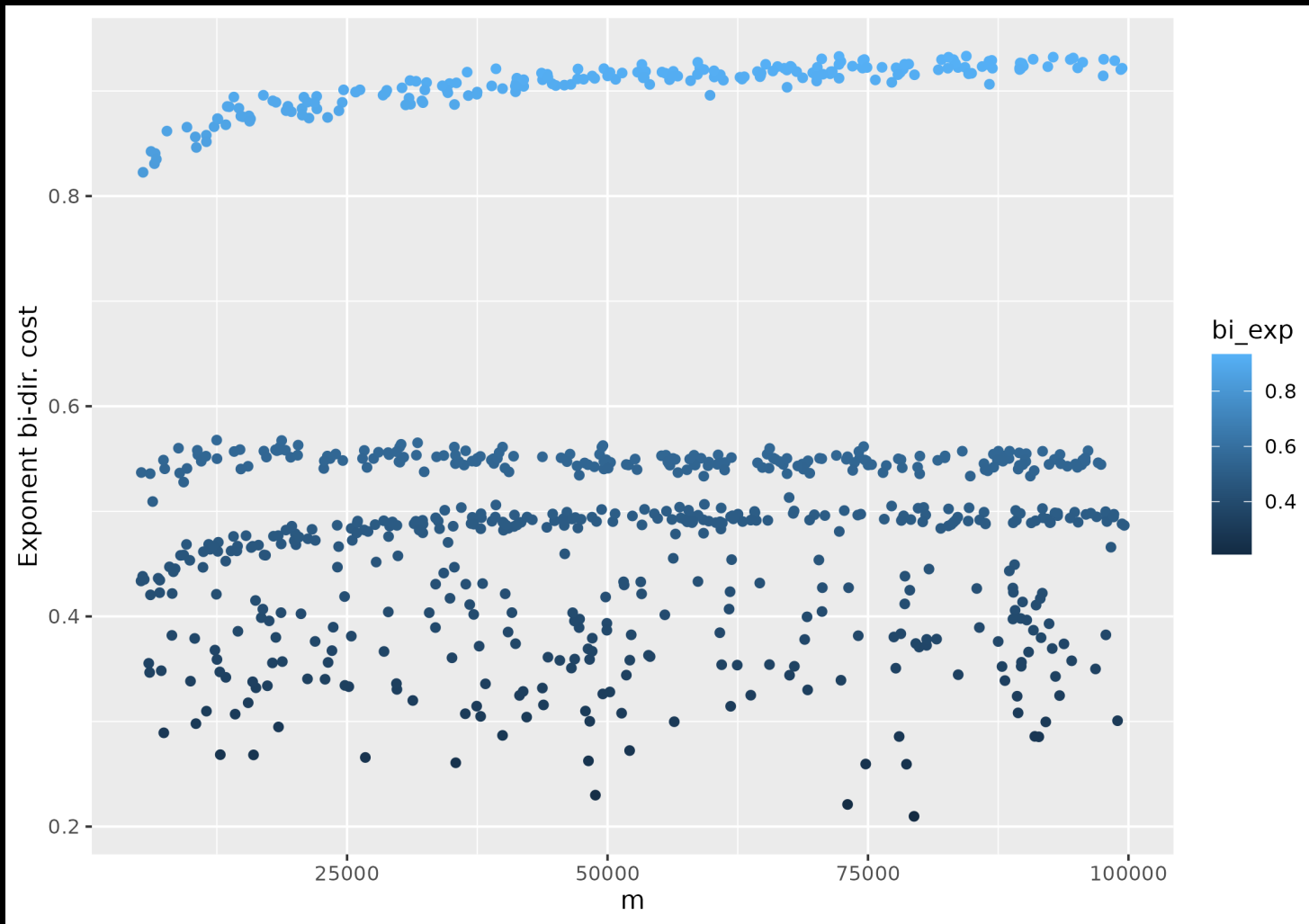
nur markieren, nicht löschen

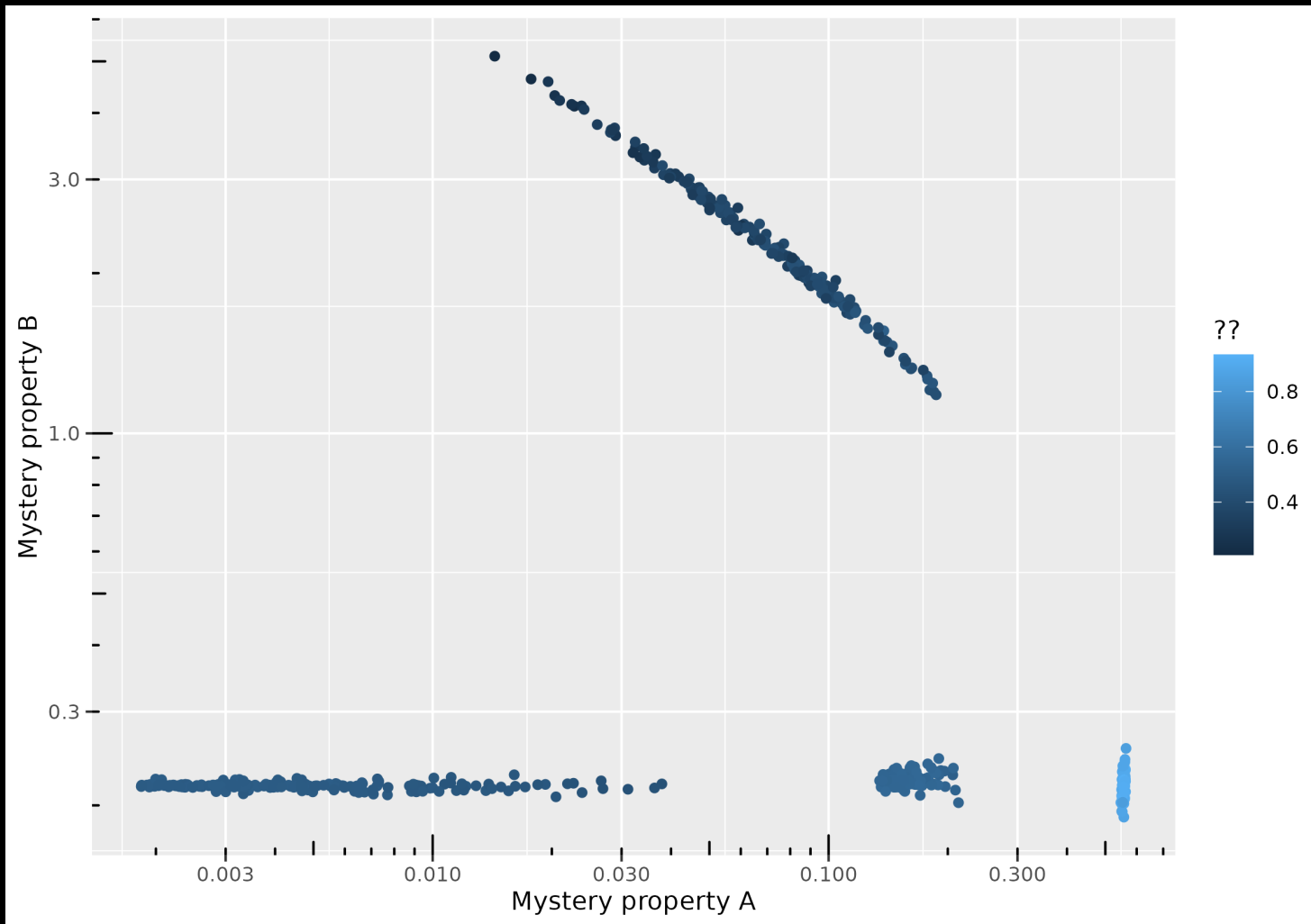


Demo

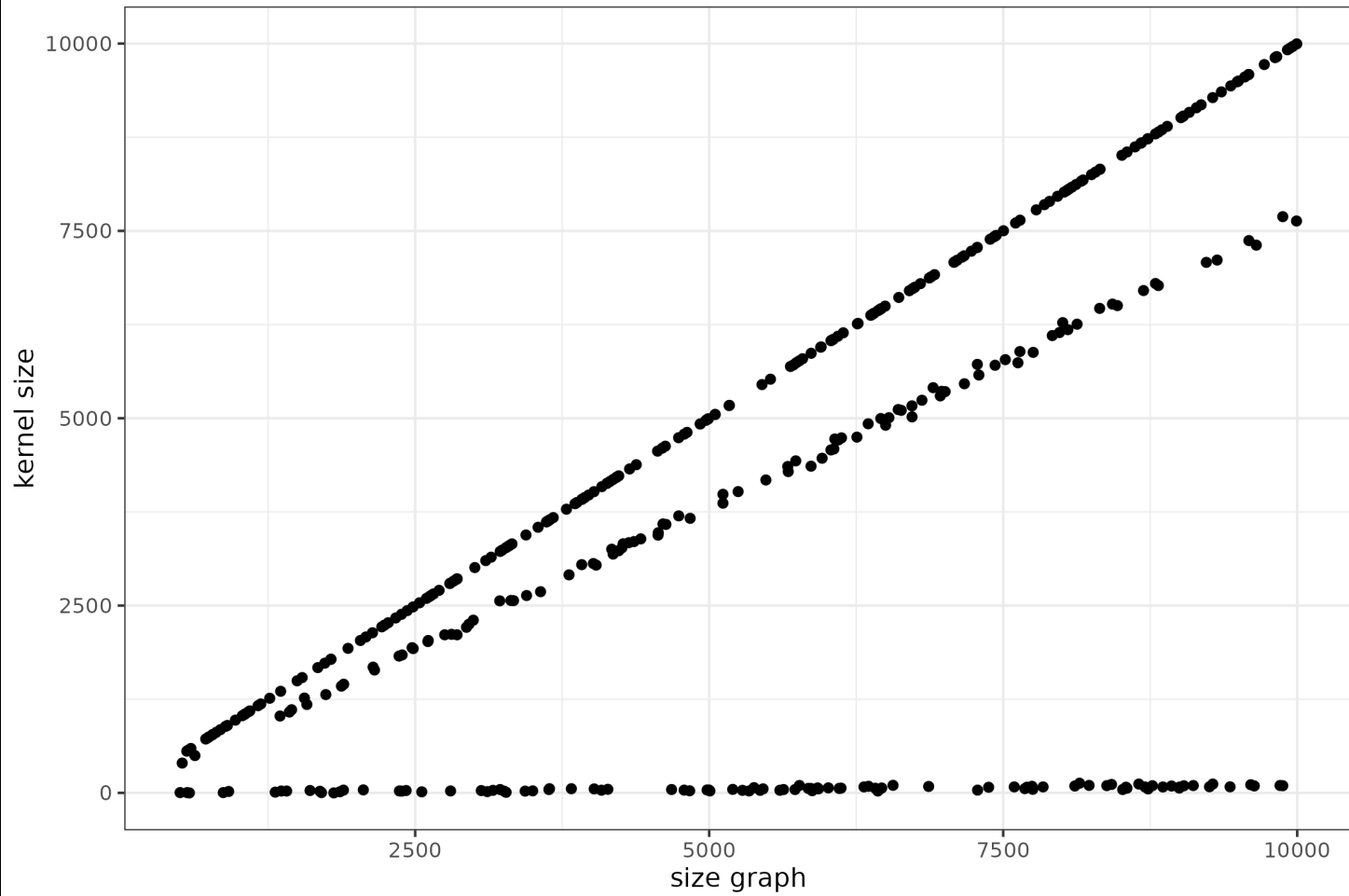




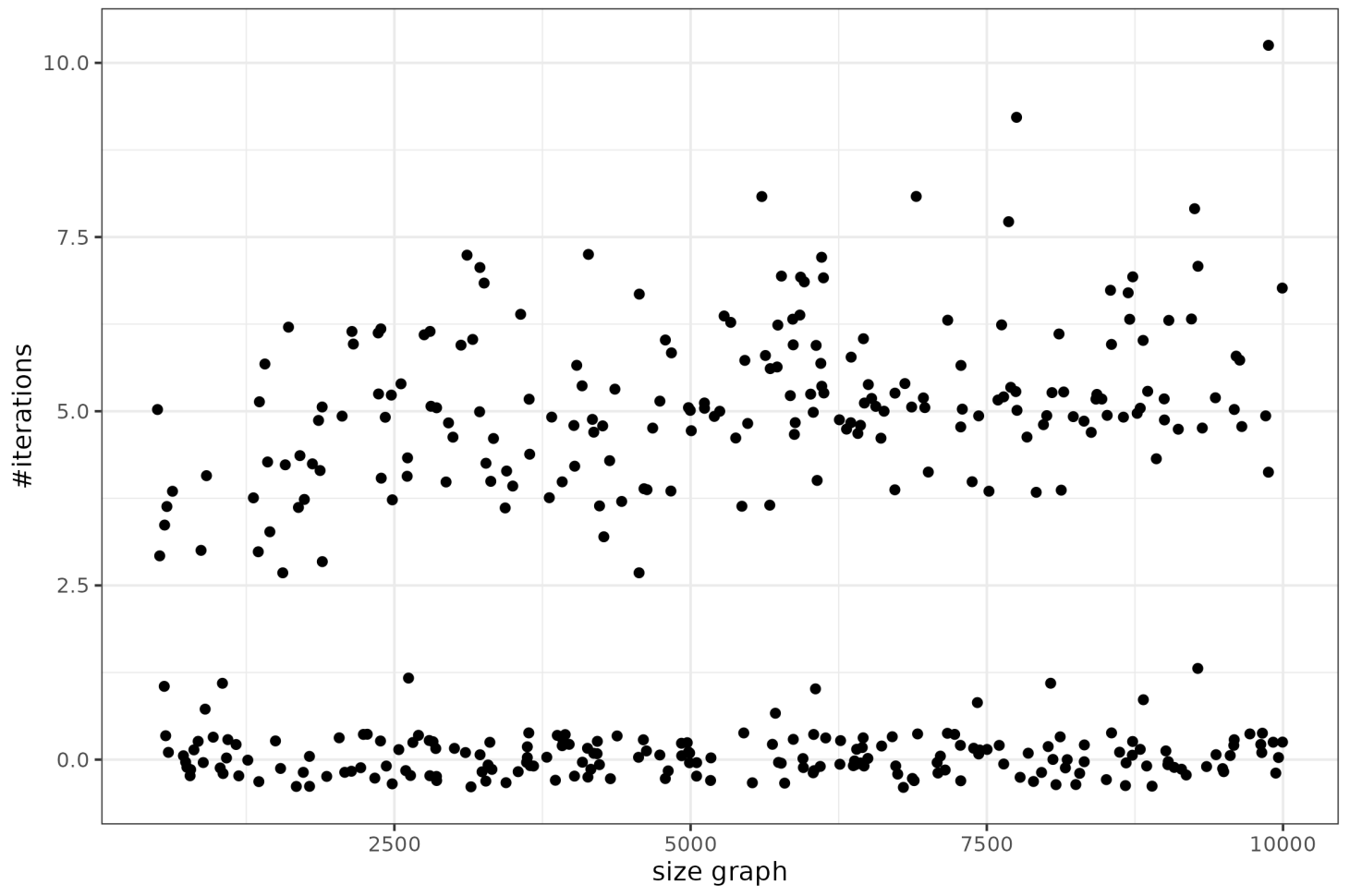




Vertex Cover Kerne Size



Vertex Cover #iterations



Plan bis nächste Woche

- Übungsblatt fertig bearbeiten
- Vorbereiten: Ergebnisse / Prozess kurz vorstellen
- Bonus: woran erkennt man, ob auf geg. Netzwerk VC-kernel klein / Bi-BFS schnell ist?

