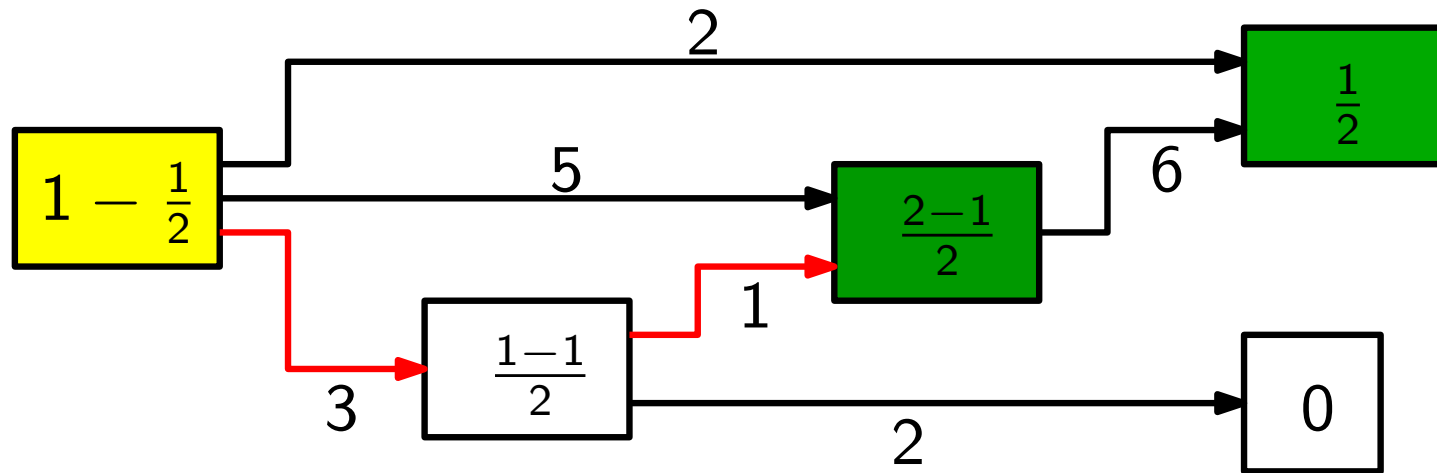


Masterarbeit

Zeichnen von Rechengraphen

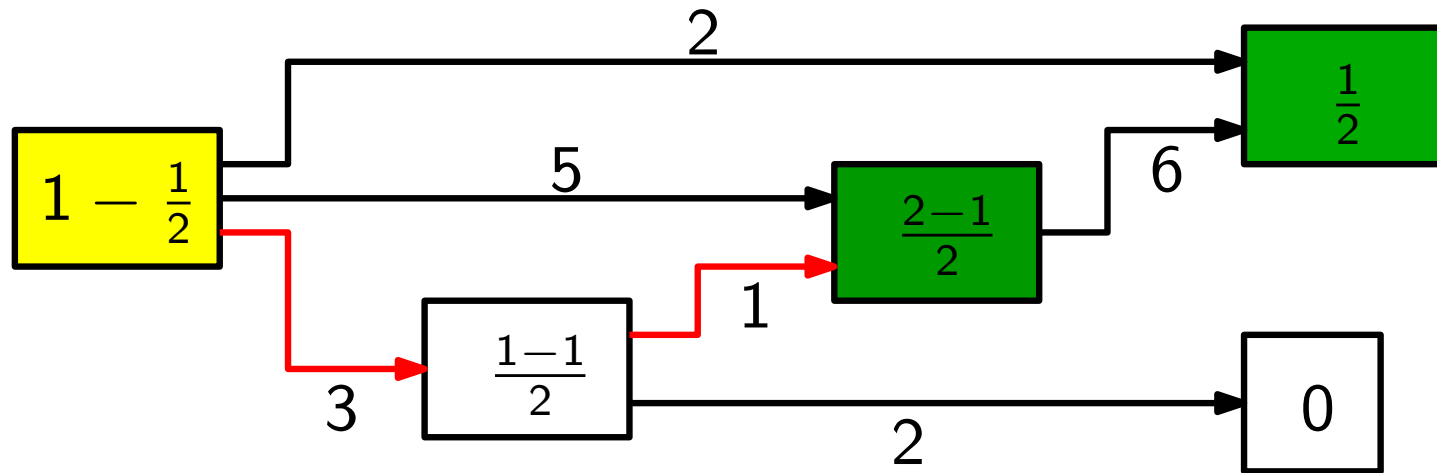
Julian Schuhmann

Einleitung



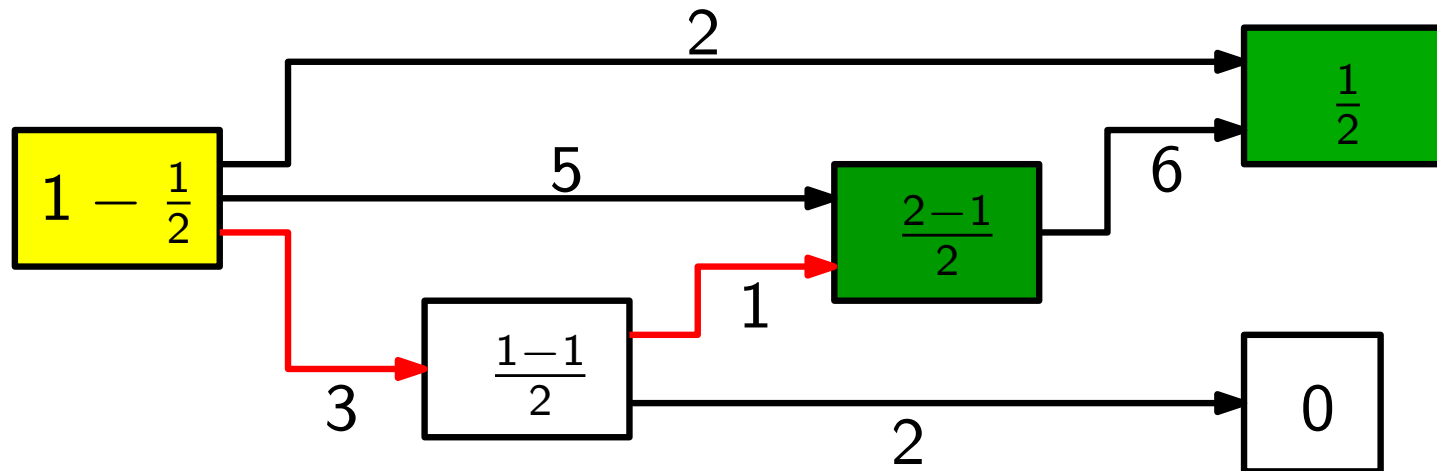
Rechengraph: – gerichteter Graph mit Kantengewichten

Einleitung



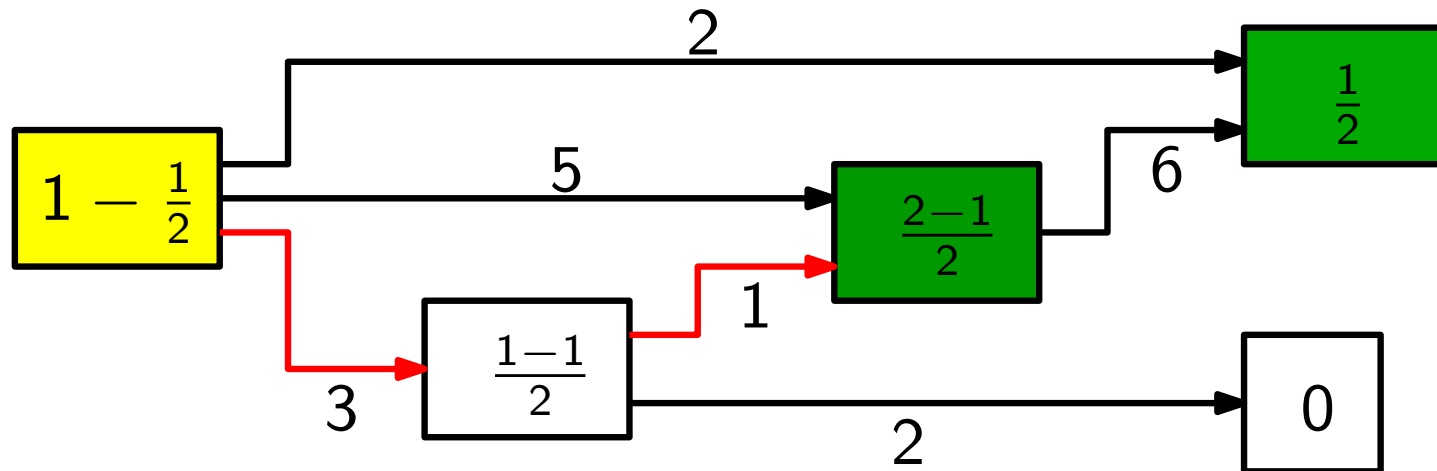
- Rechengraph:** – gerichteter Graph mit Kantengewichten
– erstmals 1999 von Martin Hennecke vorgestellt

Einleitung



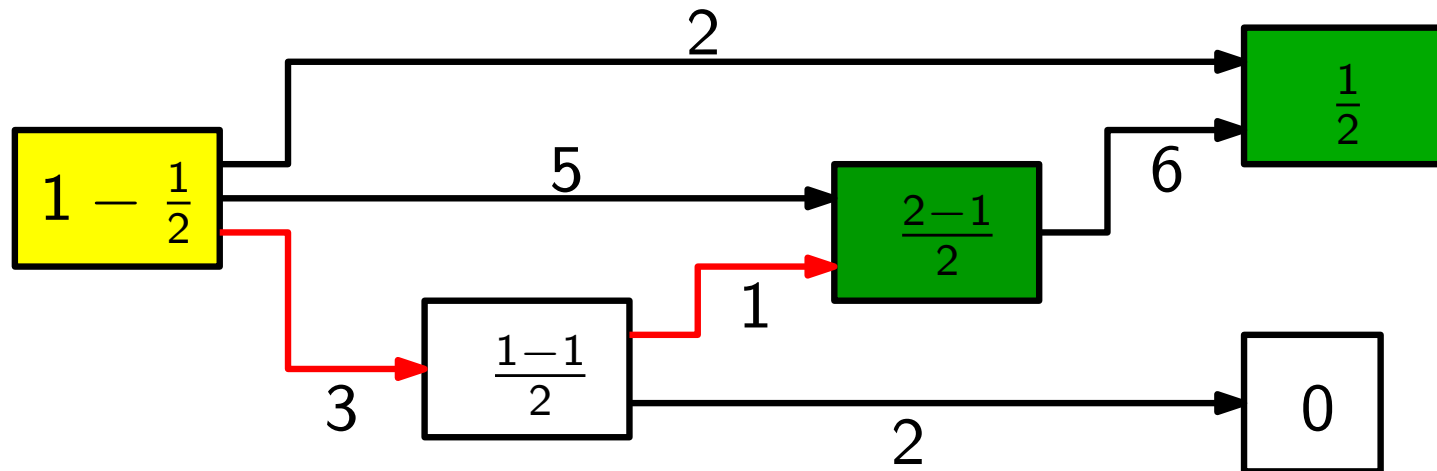
- Rechengraph:**
- gerichteter Graph mit Kantengewichten
 - erstmals 1999 von Martin Hennecke vorgestellt
 - zeigt das Vorgehen von Schülern beim Bearbeiten von Mathematikaufgaben

Einleitung



- Rechengraph:**
- gerichteter Graph mit Kantengewichten
 - erstmals 1999 von Martin Hennecke vorgestellt
 - zeigt das Vorgehen von Schülern beim Bearbeiten von Mathematikaufgaben
 - erleichtert das Finden von häufigen Fehlerquellen
→ Verbesserung der Unterrichtsqualität

Einleitung



- Rechengraph:**
- gerichteter Graph mit Kantengewichten
 - erstmals 1999 von Martin Hennecke vorgestellt
 - zeigt das Vorgehen von Schülern beim Bearbeiten von Mathematikaufgaben
 - erleichtert das Finden von häufigen Fehlerquellen
→ Verbesserung der Unterrichtsqualität

Problem: Rechengraphen sind oft sehr groß (500 - 1000 Knoten)

Problemstellung

Gegeben: – Rechengraph G mit gewichteten Kanten (Knoten)

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche
- Gesucht:** Zeichnung eines Teilgraphen von G mit folgenden Eigenschaften:

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche
- Gesucht:** Zeichnung eines Teilgraphen von G mit folgenden Eigenschaften:
- möglichst hohes Gewicht \rightarrow viele Informationen

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche
- Gesucht:** Zeichnung eines Teilgraphen von G mit folgenden Eigenschaften:
- möglichst hohes Gewicht \rightarrow viele Informationen
 - Kanten verlaufen in die gleiche Richtung (links nach rechts)

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche
- Gesucht:** Zeichnung eines Teilgraphen von G mit folgenden Eigenschaften:
- möglichst hohes Gewicht \rightarrow viele Informationen
 - Kanten verlaufen in die gleiche Richtung (links nach rechts)
 - möglichst wenige Kantenkreuzungen und Kantenknicke

Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche
- Gesucht:** Zeichnung eines Teilgraphen von G mit folgenden Eigenschaften:
- möglichst hohes Gewicht \rightarrow viele Informationen
 - Kanten verlaufen in die gleiche Richtung (links nach rechts)
 - möglichst wenige Kantenkreuzungen und Kantenknicke
- Aber:** Problem ist NP-schwer

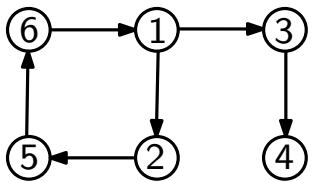
Problemstellung

- Gegeben:**
- Rechengraph G mit gewichteten Kanten (Knoten)
 - Höhe und Breite für jeden Knoten
 - Größe der Zeichenfläche
- Gesucht:** Zeichnung eines Teilgraphen von G mit folgenden Eigenschaften:
- möglichst hohes Gewicht \rightarrow viele Informationen
 - Kanten verlaufen in die gleiche Richtung (links nach rechts)
 - möglichst wenige Kantenkreuzungen und Kantenknicke
- Aber:** Problem ist NP-schwer
 \Rightarrow Einsatz von Heuristiken ist sinnvoll

Hierarchisches Zeichnen von Graphen

Sugiyama et al. [1981]

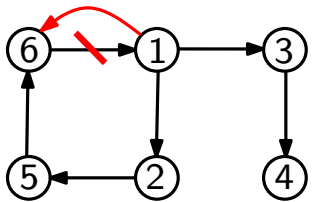
1. Entfernen von Kreisen



Hierarchisches Zeichnen von Graphen

Sugiyama et al. [1981]

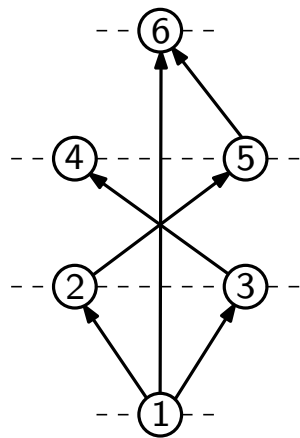
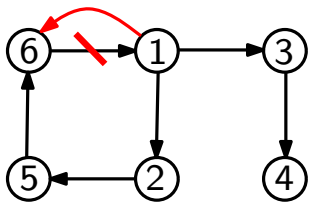
1. Entfernen von Kreisen



Hierarchisches Zeichnen von Graphen

Sugiyama et al. [1981]

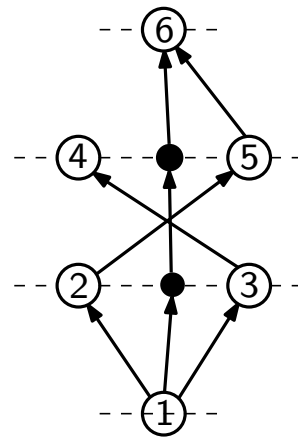
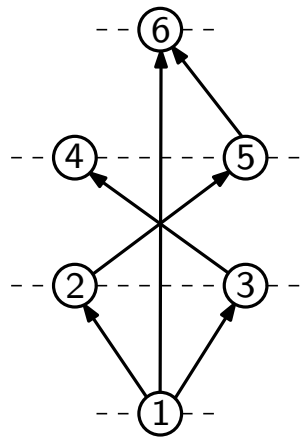
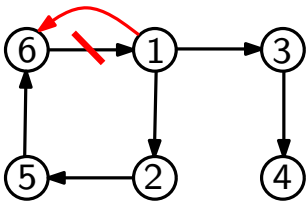
1. Entfernen von Kreisen
2. Lagenzuordnung



Hierarchisches Zeichnen von Graphen

Sugiyama et al. [1981]

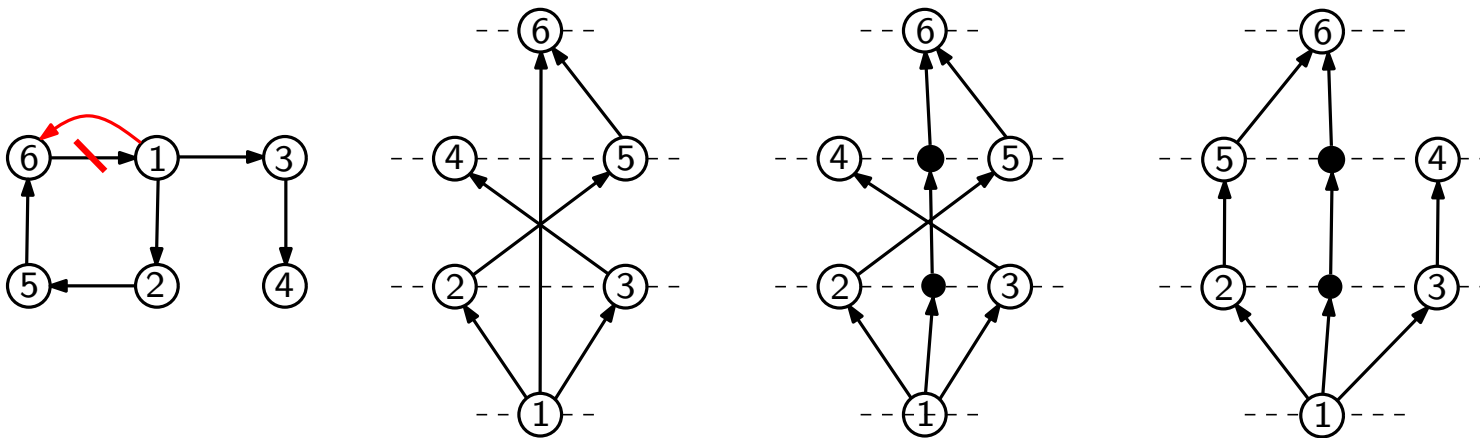
1. Entfernen von Kreisen
2. Lagenzuordnung
3. Hinzufügen von Dummy-Knoten



Hierarchisches Zeichnen von Graphen

Sugiyama et al. [1981]

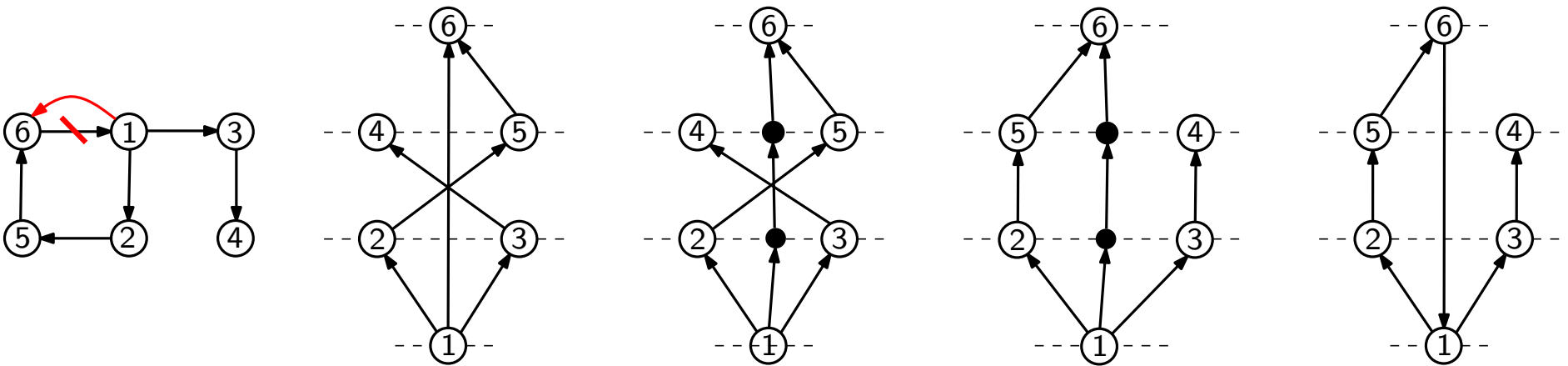
1. Entfernen von Kreisen
2. Lagenzuordnung
3. Hinzufügen von Dummy-Knoten
4. Reduzierung von Kantenkreuzungen



Hierarchisches Zeichnen von Graphen

Sugiyama et al. [1981]

1. Entfernen von Kreisen
2. Lagenzuordnung
3. Hinzufügen von Dummy-Knoten
4. Reduzierung von Kantenkreuzungen
5. Festlegen der Knotenpositionen
6. Zeichnen der Kanten



Hierarchisches Zeichnen von Rechengraphen

1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit

Hierarchisches Zeichnen von Rechengraphen

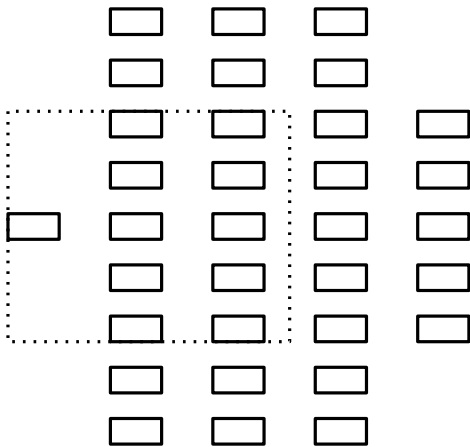
1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit
2. Entfernen von Kreisen (ILP, möglichst geringes Gewicht)

Hierarchisches Zeichnen von Rechengraphen

1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit
2. Entfernen von Kreisen (ILP, möglichst geringes Gewicht)
3. Lagenzuordnung (minimale Höhe, beschränkte Breite)

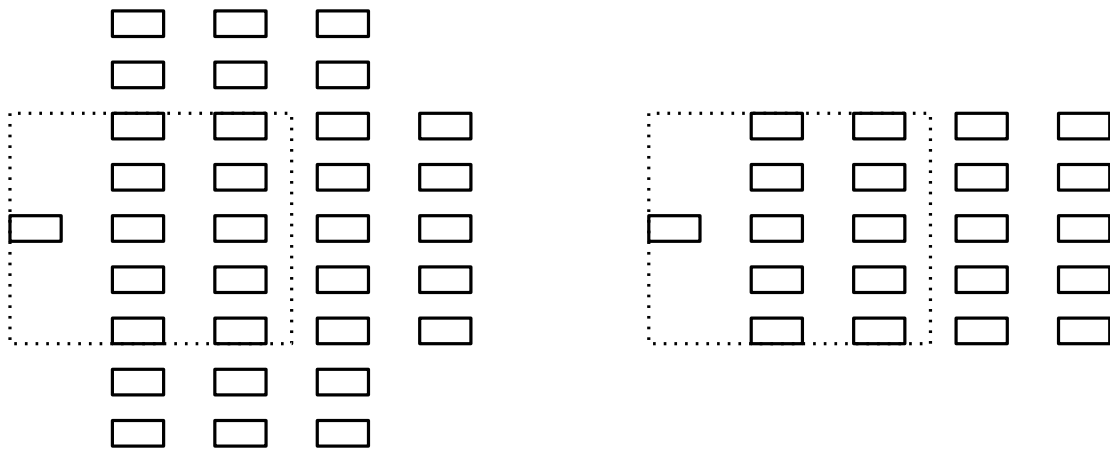
Hierarchisches Zeichnen von Rechengraphen

1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit
2. Entfernen von Kreisen (ILP, möglichst geringes Gewicht)
3. Lagenzuordnung (minimale Höhe, beschränkte Breite)
4. Für alle Lagen: entferne Knoten, bis alle auf die Zeichenfläche passen
⇒ Wichtigkeit der Knoten abhängig von Gewicht und Höhe/Breite



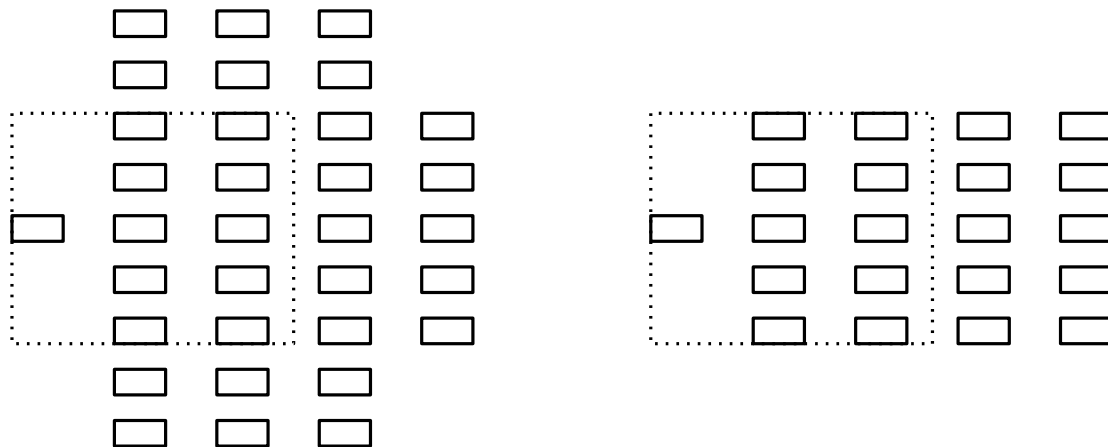
Hierarchisches Zeichnen von Rechengraphen

1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit
2. Entfernen von Kreisen (ILP, möglichst geringes Gewicht)
3. Lagenzuordnung (minimale Höhe, beschränkte Breite)
4. Für alle Lagen: entferne Knoten, bis alle auf die Zeichenfläche passen
⇒ Wichtigkeit der Knoten abhängig von Gewicht und Höhe/Breite



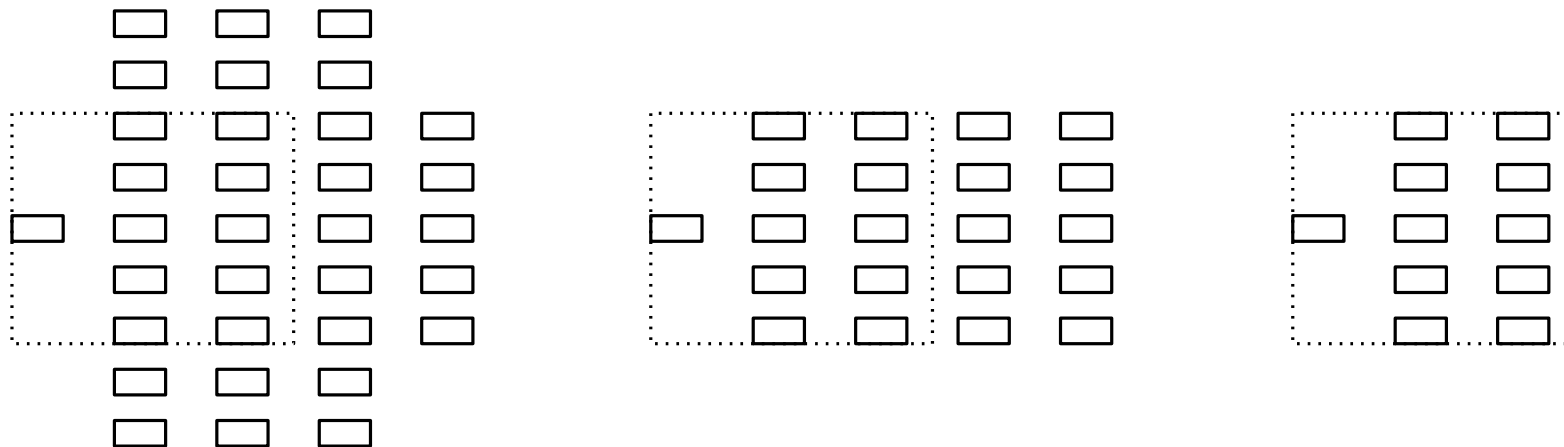
Hierarchisches Zeichnen von Rechengraphen

1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit
2. Entfernen von Kreisen (ILP, möglichst geringes Gewicht)
3. Lagenzuordnung (minimale Höhe, beschränkte Breite)
4. Für alle Lagen: entferne Knoten, bis alle auf die Zeichenfläche passen
⇒ Wichtigkeit der Knoten abhängig von Gewicht und Höhe/Breite
5. Entferne Lagen, bis alle auf die Zeichenfläche passen
⇒ Wichtigkeit der Lagen abhängig von Gewicht und Breite



Hierarchisches Zeichnen von Rechengraphen

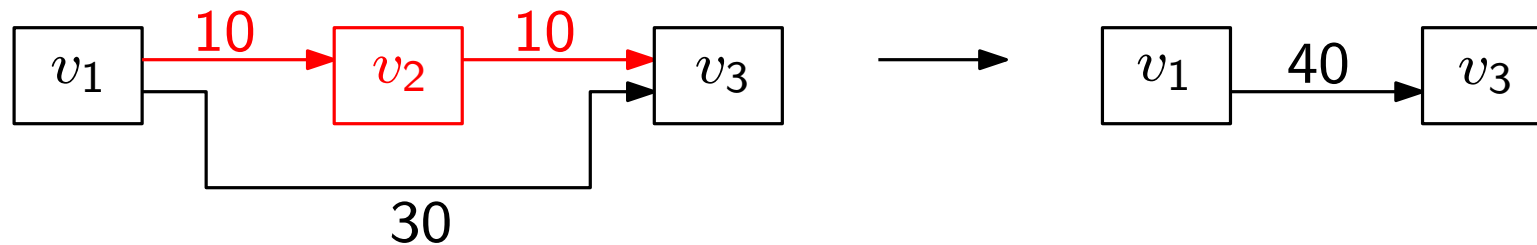
1. Entfernen von leichten Knoten und Kanten
⇒ bessere Ergebnisse, geringere Laufzeit
2. Entfernen von Kreisen (ILP, möglichst geringes Gewicht)
3. Lagenzuordnung (minimale Höhe, beschränkte Breite)
4. Für alle Lagen: entferne Knoten, bis alle auf die Zeichenfläche passen
⇒ Wichtigkeit der Knoten abhängig von Gewicht und Höhe/Breite
5. Entferne Lagen, bis alle auf die Zeichenfläche passen
⇒ Wichtigkeit der Lagen abhängig von Gewicht und Breite



Hierarchisches Zeichnen von Rechengraphen

Umlegen von Knotengewichten

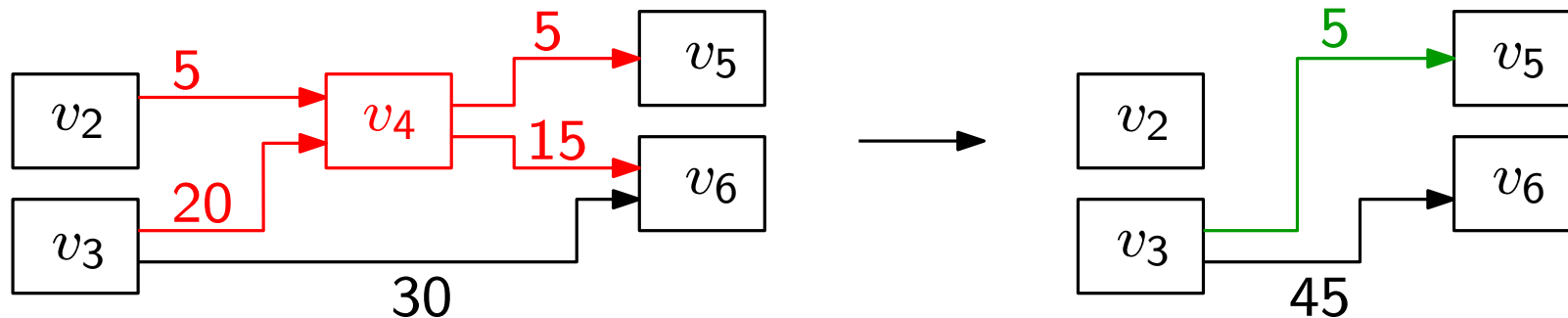
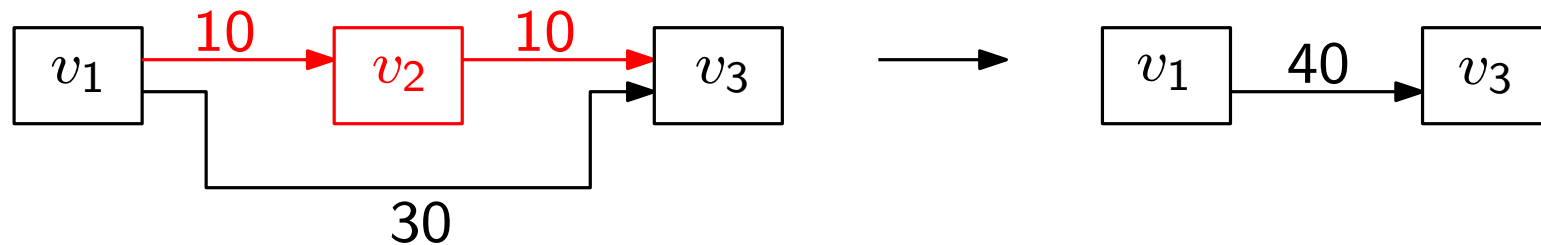
⇒ wenig Informationsverlust beim Entfernen von Knoten



Hierarchisches Zeichnen von Rechengraphen

Umlegen von Knotengewichten

⇒ wenig Informationsverlust beim Entfernen von Knoten



Hierarchisches Zeichnen von Rechengraphen

6. Hinzufügen von Dummy-Knoten

Hierarchisches Zeichnen von Rechengraphen

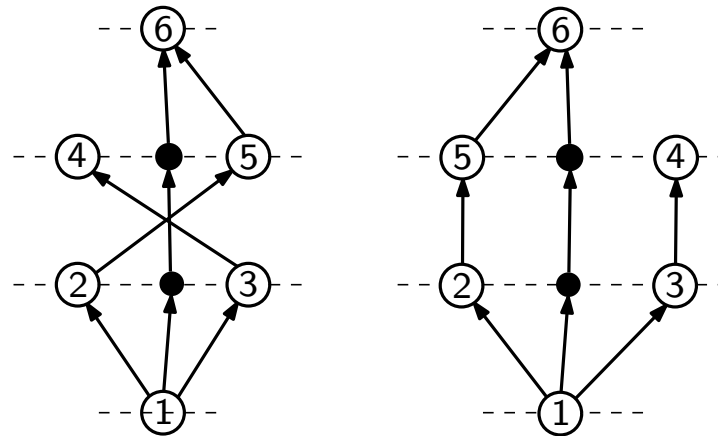
6. Hinzufügen von Dummy-Knoten

7. Reduzierung von Kantenkreuzungen: NP-schwer (Garey, Johnson [1983])

Hierarchisches Zeichnen von Rechengraphen

6. Hinzufügen von Dummy-Knoten

7. Reduzierung von Kantenkreuzungen: NP-schwer (Garey, Johnson [1983])
– Layer-by-Layer-Sweep mit



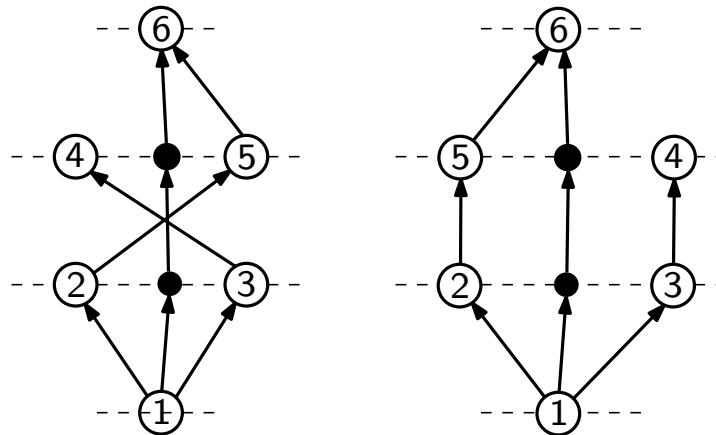
Hierarchisches Zeichnen von Rechengraphen

6. Hinzufügen von Dummy-Knoten

7. Reduzierung von Kantenkreuzungen: NP-schwer (Garey, Johnson [1983])

– Layer-by-Layer-Sweep mit

- Mittelwert-Heuristik
- Adjacent-Exchange-Heuristik (mit oder ohne Gewichte)



Hierarchisches Zeichnen von Rechengraphen

6. Hinzufügen von Dummy-Knoten

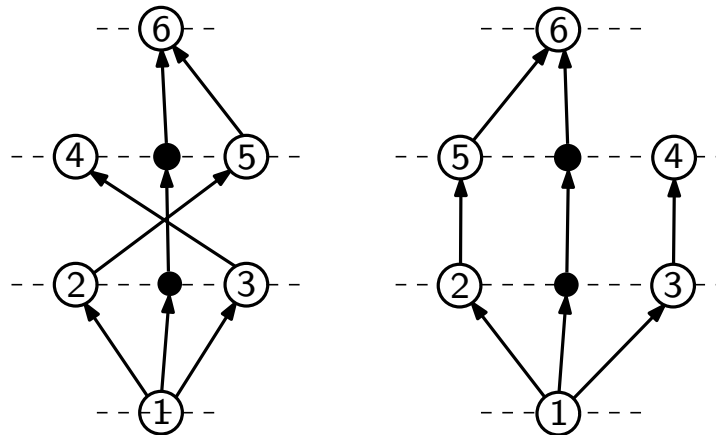
7. Reduzierung von Kantenkreuzungen: NP-schwer (Garey, Johnson [1983])

– Layer-by-Layer-Sweep mit

• Mittelwert-Heuristik

• Adjacent-Exchange-Heuristik (mit oder ohne Gewichte)

– ILP (zu langsam)



Hierarchisches Zeichnen von Rechengraphen

8. Entfernen von Kanten, wenn zu viele Kreuzungen vorhanden sind

Hierarchisches Zeichnen von Rechengraphen

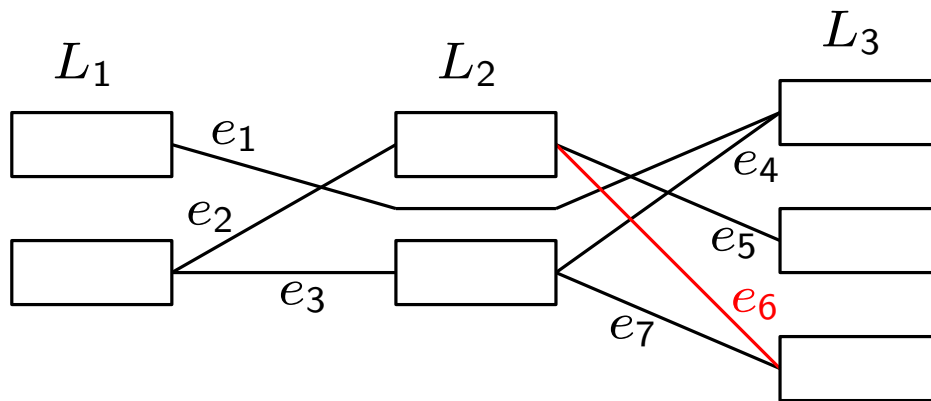
8. Entfernen von Kanten, wenn zu viele Kreuzungen vorhanden sind

$$\text{Wichtigkeit}(e) = \frac{\text{Gewicht}(e)}{\text{Gewicht der Kanten, die } e \text{ schneiden}}$$

Hierarchisches Zeichnen von Rechengraphen

8. Entfernen von Kanten, wenn zu viele Kreuzungen vorhanden sind

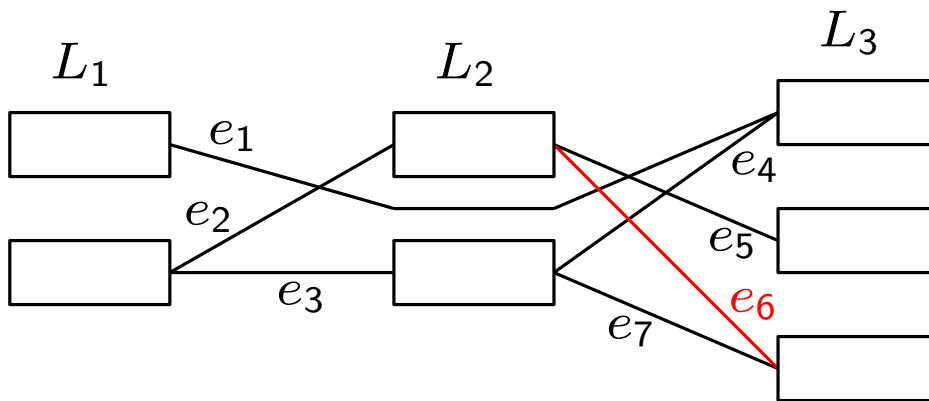
$$\text{Wichtigkeit}(e) = \frac{\text{Gewicht}(e)}{\text{Gewicht der Kanten, die } e \text{ schneiden}}$$



Hierarchisches Zeichnen von Rechengraphen

8. Entfernen von Kanten, wenn zu viele Kreuzungen vorhanden sind

$$\text{Wichtigkeit}(e) = \frac{\text{Gewicht}(e)}{\text{Gewicht der Kanten, die } e \text{ schneiden}}$$

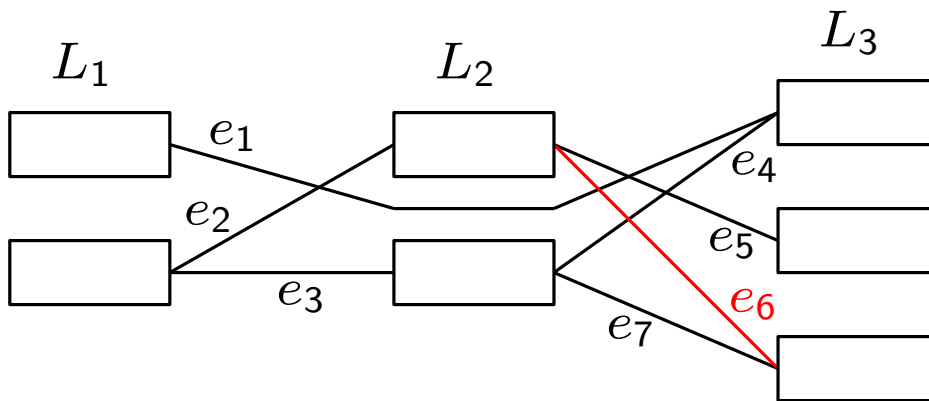


e	$w(e)$	$I(e)$
e_1	100	$\frac{100}{100+40+60} = 0.5$
e_2	100	—
e_3	60	—
e_4	35	$\frac{35}{40+60} = 0.35$
e_5	60	$\frac{60}{100+35} \approx 0.44$
e_6	40	$\frac{40}{100+35} \approx 0.30$
e_7	20	∞

Hierarchisches Zeichnen von Rechengraphen

8. Entfernen von Kanten, wenn zu viele Kreuzungen vorhanden sind

$$\text{Wichtigkeit}(e) = \frac{\text{Gewicht}(e)}{\text{Gewicht der Kanten, die } e \text{ schneiden}}$$



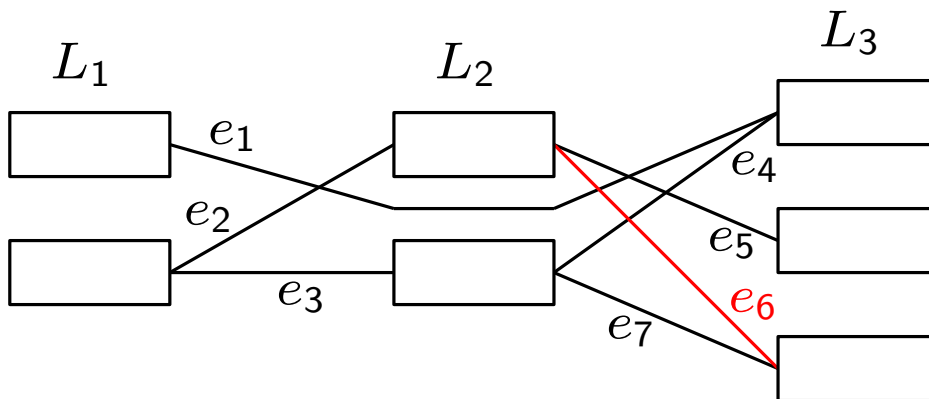
e	$w(e)$	$I(e)$
e_1	100	$\frac{100}{100+40+60} = 0.5$
e_2	100	—
e_3	60	—
e_4	35	$\frac{35}{40+60} = 0.35$
e_5	60	$\frac{60}{100+35} \approx 0.44$
e_6	40	$\frac{40}{100+35} \approx 0.30$
e_7	20	∞

9. Einfügen von entfernten Knoten und Kanten, wenn genug Platz ist
Wiederholung der Schritte 6-8

Hierarchisches Zeichnen von Rechengraphen

8. Entfernen von Kanten, wenn zu viele Kreuzungen vorhanden sind

$$\text{Wichtigkeit}(e) = \frac{\text{Gewicht}(e)}{\text{Gewicht der Kanten, die } e \text{ schneiden}}$$



e	$w(e)$	$I(e)$
e_1	100	$\frac{100}{100+40+60} = 0.5$
e_2	100	—
e_3	60	—
e_4	35	$\frac{35}{40+60} = 0.35$
e_5	60	$\frac{60}{100+35} \approx 0.44$
e_6	40	$\frac{40}{100+35} \approx 0.30$
e_7	20	∞

9. Einfügen von entfernten Knoten und Kanten, wenn genug Platz ist
Wiederholung der Schritte 6-8

10. Festlegen der Knotenpositionen

Hierarchisches Zeichnen von Rechengraphen

11. Reduzierung der Anzahl von Knicken (Sander [1996])
⇒ nicht sinnvoll, zu hoher Platzbedarf

Hierarchisches Zeichnen von Rechengraphen

11. Reduzierung der Anzahl von Knicken (Sander [1996])

⇒ nicht sinnvoll, zu hoher Platzbedarf

12. Zeichnen der Kanten

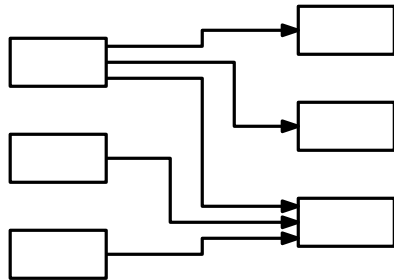
Hierarchisches Zeichnen von Rechengraphen

11. Reduzierung der Anzahl von Knicken (Sander [1996])

⇒ nicht sinnvoll, zu hoher Platzbedarf

12. Zeichnen der Kanten

– orthogonale Kanten mit mehreren Ausgangsports pro Knoten

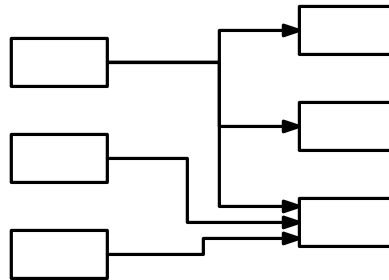
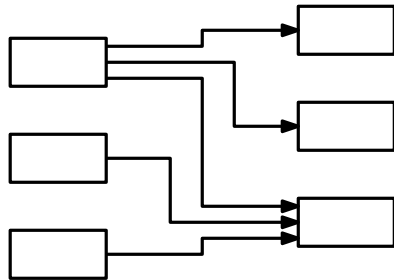


Hierarchisches Zeichnen von Rechengraphen

11. Reduzierung der Anzahl von Knicken (Sander [1996])
⇒ nicht sinnvoll, zu hoher Platzbedarf

12. Zeichnen der Kanten

- orthogonale Kanten mit mehreren Ausgangsports pro Knoten
- orthogonale Kanten mit einem Ausgangsport pro Knoten

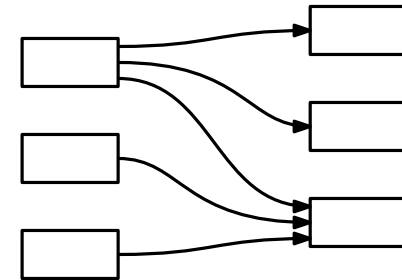
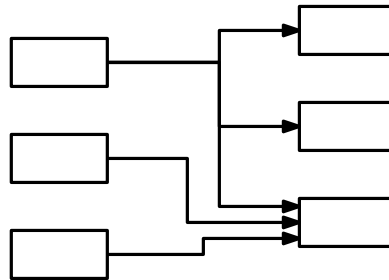
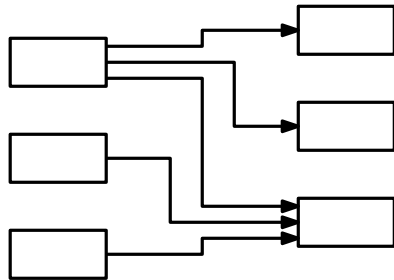


Hierarchisches Zeichnen von Rechengraphen

11. Reduzierung der Anzahl von Knicken (Sander [1996])
⇒ nicht sinnvoll, zu hoher Platzbedarf

12. Zeichnen der Kanten

- orthogonale Kanten mit mehreren Ausgangsports pro Knoten
- orthogonale Kanten mit einem Ausgangsport pro Knoten
- Kanten als Bézierkurven

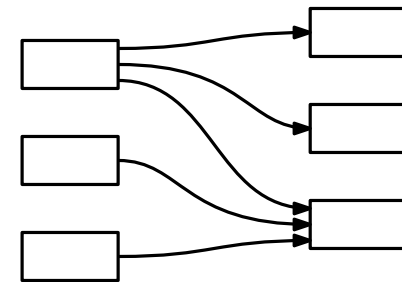
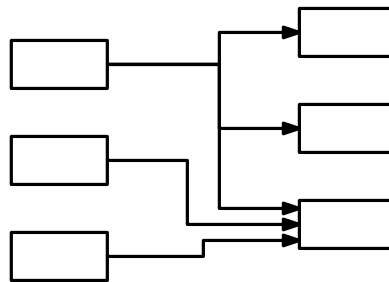
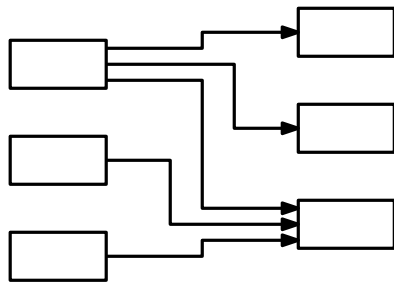


Hierarchisches Zeichnen von Rechengraphen

11. Reduzierung der Anzahl von Knicken (Sander [1996])
⇒ nicht sinnvoll, zu hoher Platzbedarf

12. Zeichnen der Kanten

- orthogonale Kanten mit mehreren Ausgangsports pro Knoten
- orthogonale Kanten mit einem Ausgangsport pro Knoten
- Kanten als Bézierkurven



Dicke der Kanten abhängig von ihrem Gewicht

Orthogonale Kanten

Zuerst: Festlegen der Ports an den Knoten

Orthogonale Kanten

Zuerst: Festlegen der Ports an den Knoten

Kanten bestehen aus drei Teilen:

- zwei horizontale Segmente (durch Ports festgelegt)
- ein vertikales Segment

Orthogonale Kanten

Zuerst: Festlegen der Ports an den Knoten

Kanten bestehen aus drei Teilen:

- zwei horizontale Segmente (durch Ports festgelegt)
- ein vertikales Segment

⇒ Finden einer guten Zeichnung = Finden von guten Positionen für die vertikalen Segmente

Orthogonale Kanten

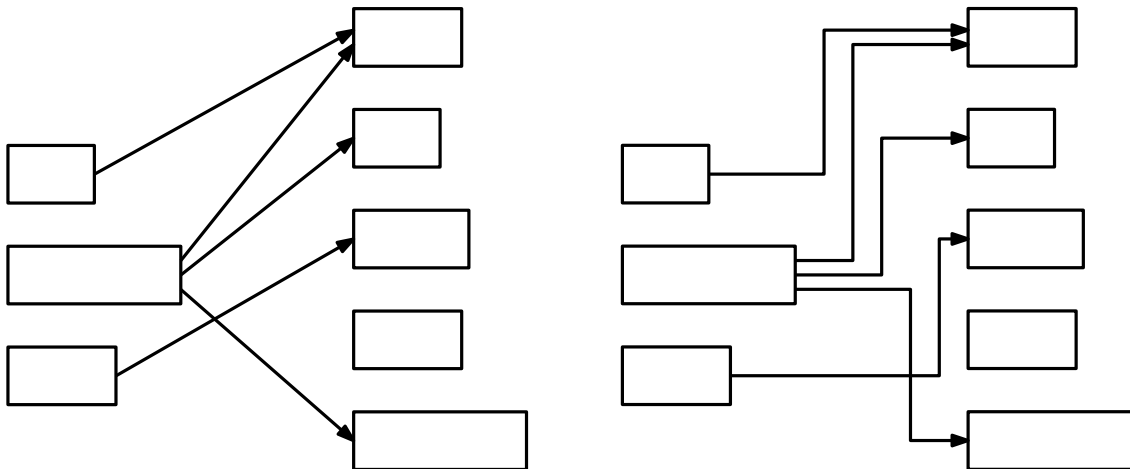
Zuerst: Festlegen der Ports an den Knoten

Kanten bestehen aus drei Teilen:

- zwei horizontale Segmente (durch Ports festgelegt)
- ein vertikales Segment

⇒ Finden einer guten Zeichnung = Finden von guten Positionen für die vertikalen Segmente

1. Setzen der vertikalen Segmente, so dass keine zusätzlichen Kreuzungen entstehen



Orthogonale Kanten

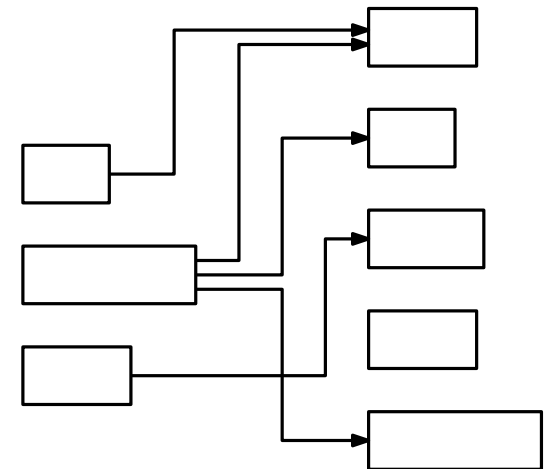
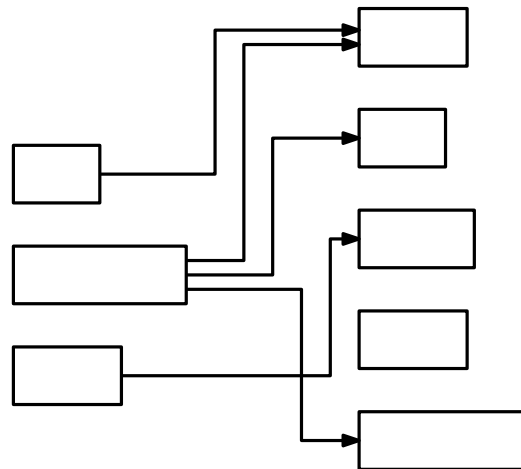
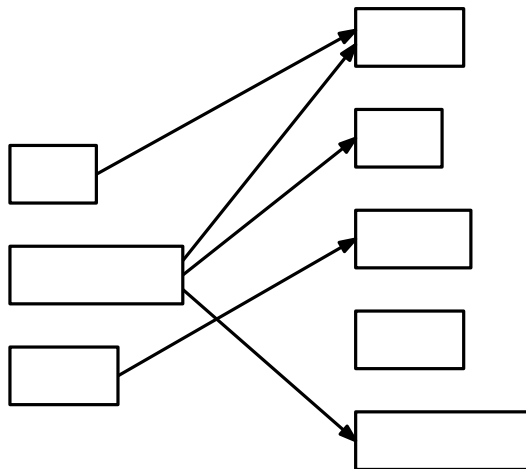
Zuerst: Festlegen der Ports an den Knoten

Kanten bestehen aus drei Teilen:

- zwei horizontale Segmente (durch Ports festgelegt)
- ein vertikales Segment

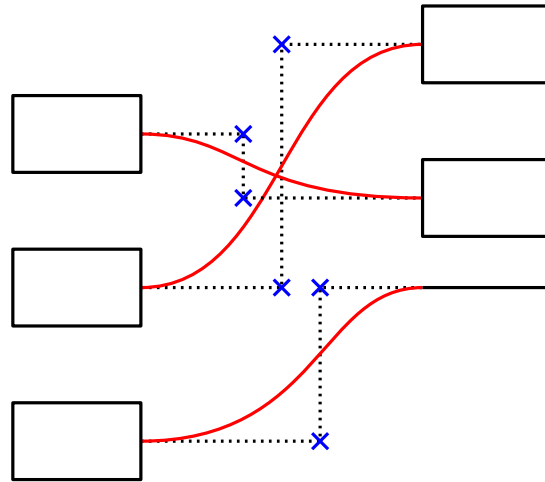
⇒ Finden einer guten Zeichnung = Finden von guten Positionen für die vertikalen Segmente

1. Setzen der vertikalen Segmente, so dass keine zusätzlichen Kreuzungen entstehen
2. Verbesserung der Zeichnung durch kräftebasierten Algorithmus



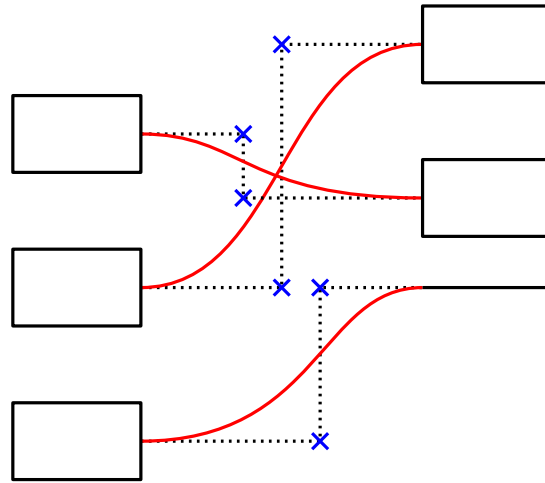
Kanten als Bézierkurven

- Grundlegendes Vorgehen wie bei orthogonalen Kanten

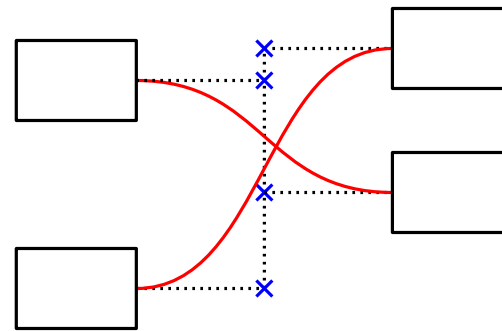
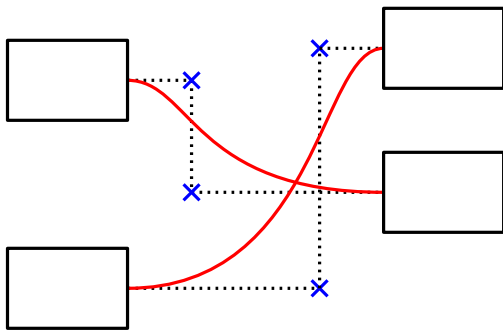


Kanten als Bézierkurven

- Grundlegendes Vorgehen wie bei orthogonalen Kanten



- Leichte Anpassungen, um schönere Kurven zu erhalten



Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%

Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%
- Umlegen von Gewichten: Gewichtserhöhung um 2%

Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%
- Umlegen von Gewichten: Gewichtserhöhung um 2%
- Algorithmen zur Lagenzuordnung: alle ähnlich gut

Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%
- Umlegen von Gewichten: Gewichtserhöhung um 2%
- Algorithmen zur Lagenzuordnung: alle ähnlich gut
- Reduzierung von Kantenkreuzungen:
 - Mittelwert-Heuristik für minimale Anzahl
 - Adjacent-Exchange-Heuristik für minimales Gewicht (-60%)

Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%
- Umlegen von Gewichten: Gewichtserhöhung um 2%
- Algorithmen zur Lagenzuordnung: alle ähnlich gut
- Reduzierung von Kantenkreuzungen:
 - Mittelwert-Heuristik für minimale Anzahl
 - Adjacent-Exchange-Heuristik für minimales Gewicht (-60%)
- Einfügen von entfernten Knoten/Kanten: Gewichtserhöhung um 3,5%

Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%
- Umlegen von Gewichten: Gewichtserhöhung um 2%
- Algorithmen zur Lagenzuordnung: alle ähnlich gut
- Reduzierung von Kantenkreuzungen:
 - Mittelwert-Heuristik für minimale Anzahl
 - Adjacent-Exchange-Heuristik für minimales Gewicht (-60%)
- Einfügen von entfernten Knoten/Kanten: Gewichtserhöhung um 3,5%
- benötigte Laufzeit: 1-4 Sekunden

Testergebnisse

- optimale Schranke für das Aussortieren von leichten Knoten und Kanten wächst mit der Größe des Graphen
 - ⇒ Gewichtserhöhung um bis zu 17%
- Umlegen von Gewichten: Gewichtserhöhung um 2%
- Algorithmen zur Lagenzuordnung: alle ähnlich gut
- Reduzierung von Kantenkreuzungen:
 - Mittelwert-Heuristik für minimale Anzahl
 - Adjacent-Exchange-Heuristik für minimales Gewicht (-60%)
- Einfügen von entfernten Knoten/Kanten: Gewichtserhöhung um 3,5%
- benötigte Laufzeit: 1-4 Sekunden
- Beispielausgaben

Fazit und Ausblick

- Algorithmus liefert gute Zeichnungen

Fazit und Ausblick

- Algorithmus liefert gute Zeichnungen
- Zeichnungen enthalten wenige Knoten, aber einen Großteil des Gewichts

Fazit und Ausblick

- Algorithmus liefert gute Zeichnungen
- Zeichnungen enthalten wenige Knoten, aber einen Großteil des Gewichts
- Erstellen von guten Zeichnungen:
 - Test mit unterschiedlichen Einstellungen
 - mehrere Ausgaben mit guten Einstellungen generieren

Fazit und Ausblick

- Algorithmus liefert gute Zeichnungen
- Zeichnungen enthalten wenige Knoten, aber einen Großteil des Gewichts
- Erstellen von guten Zeichnungen:
 - Test mit unterschiedlichen Einstellungen
 - mehrere Ausgaben mit guten Einstellungen generieren
- mögliche Erweiterungen:
 - Rückwärtskanten
 - Beschriften der Kanten mit ihren Gewichten

Fazit und Ausblick

- Algorithmus liefert gute Zeichnungen
- Zeichnungen enthalten wenige Knoten, aber einen Großteil des Gewichts
- Erstellen von guten Zeichnungen:
 - Test mit unterschiedlichen Einstellungen
 - mehrere Ausgaben mit guten Einstellungen generieren
- mögliche Erweiterungen:
 - Rückwärtskanten
 - Beschriften der Kanten mit ihren Gewichten

Fragen?