UNIVERSITÄT ZU LÜBECK
INSTITUT FÜR
THEORETISCHE INFORMATIK

Experimentelle Analyse von Algorithmen zum Erweitern kausaler Graphen

Experimental Analysis of Algorithms for Extending Causal Graphs

Kolloquium von Malte Luttermann

16. September 2021

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Motivation

- Kausale Zusammenhänge sind in vielen Forschungsbereichen relevant

Motivation

- Kausale Zusammenhänge sind in vielen Forschungsbereichen relevant
- Experimente eignen sich zwar gut, sind aber nicht immer umsetzbar

Motivation

- Kausale Zusammenhänge sind in vielen Forschungsbereichen relevant
- Experimente eignen sich zwar gut, sind aber nicht immer umsetzbar
- Daher: Kausale Zusammenhänge anhand von beobachteten Daten lernen

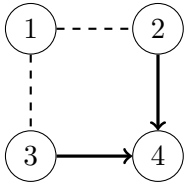
Motivation

- Kausale Zusammenhänge sind in vielen Forschungsbereichen relevant
- Experimente eignen sich zwar gut, sind aber nicht immer umsetzbar
- Daher: Kausale Zusammenhänge anhand von beobachteten Daten lernen
  - Kausaler Graph zur Modellierung

Motivation

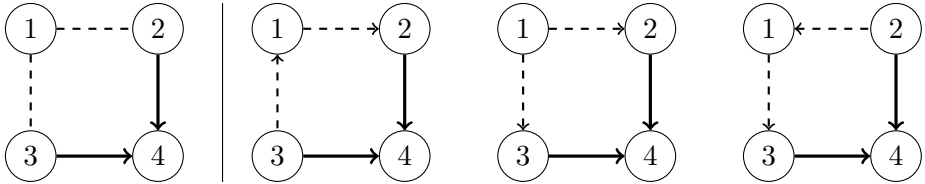
- Kausale Zusammenhänge sind in vielen Forschungsbereichen relevant
- Experimente eignen sich zwar gut, sind aber nicht immer umsetzbar
- Daher: Kausale Zusammenhänge anhand von beobachteten Daten lernen
  - Kausaler Graph zur Modellierung
  - Problem: Beobachtete Daten liefern unvollständige Informationen

Das Erweiterungsproblem



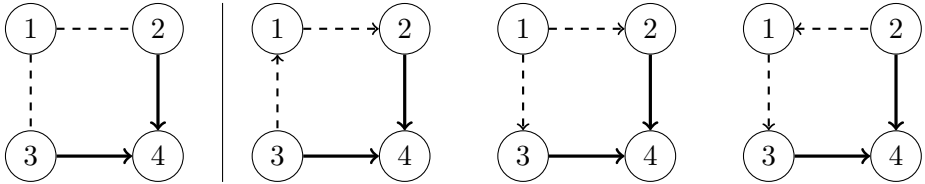
Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Das Erweiterungsproblem



Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Das Erweiterungsproblem



Das Erweiterungsproblem

Eingabe: Ein partiell gerichteter Graph G.

Ausgabe: Eine konsistente DAG-Erweiterung von G falls G erweiterbar ist, ansonsten eine negative Antwort.

Erweitern kausaler Graphen

Malte Luttermann

Motivation

Erweiterungen für Graphen

Das Erweiterungsproblem

Algorithmen zum Lösen des Erweiterungsproblems

Evaluation

Enumeration von Erweiterungen

Zusammenfassung

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G

[1] D. Dor und M. Tarsi. »A simple algorithm to construct a consistent extension of a partially oriented graph«. In: *Technicial Report R-185, Cognitive Systems Laboratory, UCLA* (1992).

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:

[1] Dor und Tarsi, »A simple algorithm to construct a consistent extension of a partially oriented graph«.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:
   - **2.1** Wähle einen Knoten s aus, der folgende Eigenschaften erfüllt:
     - s besitzt keine ausgehenden Kanten
     - Jeder ungerichtete Nachbar von s ist zu allen anderen Nachbarn von s benachbart

[1] Dor und Tarsi, »A simple algorithm to construct a consistent extension of a partially oriented graph«.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:
   - **2.1** Wähle einen Knoten s aus, der folgende Eigenschaften erfüllt:
     - s besitzt keine ausgehenden Kanten
     - Jeder ungerichtete Nachbar von s ist zu allen anderen Nachbarn von s benachbart
   - **2.2** Falls kein Knoten s gefunden wurde, gib eine negative Antwort zurück

[1] Dor und Tarsi, »A simple algorithm to construct a consistent extension of a partially oriented graph«.

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:
   - **2.1** Wähle einen Knoten s aus, der folgende Eigenschaften erfüllt:
     - s besitzt keine ausgehenden Kanten
     - Jeder ungerichtete Nachbar von s ist zu allen anderen Nachbarn von s benachbart
   - **2.2** Falls kein Knoten s gefunden wurde, gib eine negative Antwort zurück
   - **2.3** Richte alle ungerichteten Kanten, die zu s benachbart sind, auf s in G'

[1] Dor und Tarsi, »A simple algorithm to construct a consistent extension of a partially oriented graph«.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:
   - **2.1** Wähle einen Knoten s aus, der folgende Eigenschaften erfüllt:
     - s besitzt keine ausgehenden Kanten
     - Jeder ungerichtete Nachbar von s ist zu allen anderen Nachbarn von s benachbart
   - **2.2** Falls kein Knoten s gefunden wurde, gib eine negative Antwort zurück
   - **2.3** Richte alle ungerichteten Kanten, die zu s benachbart sind, auf s in G'
   - **2.4** Entferne s und alle benachbarten Kanten aus G

[1] Dor und Tarsi, »A simple algorithm to construct a consistent extension of a partially oriented graph«.

Der Algorithmus von Dor und Tarsi[1]

(Algorithmus DT)

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:
   - **2.1** Wähle einen Knoten s aus, der folgende Eigenschaften erfüllt:
     - s besitzt keine ausgehenden Kanten
     - Jeder ungerichtete Nachbar von s ist zu allen anderen Nachbarn von s benachbart
   - **2.2** Falls kein Knoten s gefunden wurde, gib eine negative Antwort zurück
   - **2.3** Richte alle ungerichteten Kanten, die zu s benachbart sind, auf s in G'
   - **2.4** Entferne s und alle benachbarten Kanten aus G
3. Gib G' zurück

[1] Dor und Tarsi, »A simple algorithm to construct a consistent extension of a partially oriented graph«.

Der Algorithmus von Wienöbst, Bannach, und Liśkiewicz[2]

(Algorithmus WBL bzw. WBL+DO)

- Ziel: Wiederholtes Iterieren über die Nachbarschaften vermeiden

[2] Marcel Wienöbst, Max Bannach und Maciej Liśkiewicz. »Extendability of Causal Graphical Models: Algorithms and Computational Complexity«. In: *Conference on Uncertainty in Artificial Intelligence (UAI)* (2021).

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Der Algorithmus von Wienöbst, Bannach, und Liśkiewicz[2]

(Algorithmus WBL bzw. WBL+DO)

- Ziel: Wiederholtes Iterieren über die Nachbarschaften vermeiden
- Dazu: Neue Datenstruktur
  - Bei Initialisierung Informationen über Nachbarschaften abspeichern
  - Bei Entfernen eines Knotens diese Informationen aktualisieren

[2] Wienöbst, Bannach und Liśkiewicz, »Extendability of Causal Graphical Models: Algorithms and Computational Complexity«.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Der Algorithmus von Wienöbst, Bannach, und Liśkiewicz[2]

(Algorithmus WBL bzw. WBL+DO)

- Ziel: Wiederholtes Iterieren über die Nachbarschaften vermeiden
- Dazu: Neue Datenstruktur
  - Bei Initialisierung Informationen über Nachbarschaften abspeichern
  - Bei Entfernen eines Knotens diese Informationen aktualisieren
- Initialisierung kann optimiert werden (Algorithmus WBL+DO)
  - Günstigere Reihenfolge der Kanteneinfügungen in die Datenstruktur

[2] Wienöbst, Bannach und Liśkiewicz, »Extendability of Causal Graphical Models: Algorithms and Computational Complexity«.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Eine Heuristik für den Algorithmus von Dor und Tarsi

(Algorithmus DT+H)

- Idee: Knoten mit einem kleineren Grad verursachen weniger Kosten

Eine Heuristik für den Algorithmus von Dor und Tarsi

(Algorithmus DT+H)

- Idee: Knoten mit einem kleineren Grad verursachen weniger Kosten

1. Erstelle eine Kopie G' des Eingabegraphen G
2. Wiederhole, solange noch Knoten in G vorhanden sind:
   - 2.1 Wähle einen Knoten s aus, der folgende Eigenschaften erfüllt:
     - s besitzt keine ausgehenden Kanten
     - Jeder ungerichtete Nachbar von s ist zu allen anderen Nachbarn von s benachbart
   - 2.2 Falls kein Knoten s gefunden wurde, gib eine negative Antwort zurück
   - 2.3 Richte alle ungerichteten Kanten, die zu s benachbart sind, auf s in G'
   - 2.4 Entferne s und alle benachbarten Kanten aus G
3. Gib G' zurück

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Das Erweiterungsproblem
Algorithmen zum Lösen des Erweiterungsproblems
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Datensatzgenerierung für die Evaluation

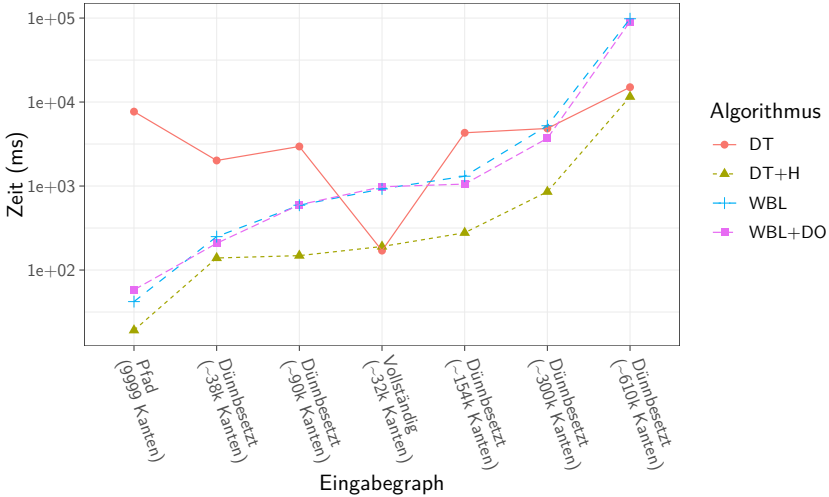
- Zwei Arten von Graphinstanzen im Datensatz:
  1. Zufällig generierte Graphen
  2. Händisch ausgewählte »Worst-Case« Graphinstanzen

Datensatzgenerierung für die Evaluation

- Zwei Arten von Graphinstanzen im Datensatz:
  1. Zufällig generierte Graphen
  2. Händisch ausgewählte »Worst-Case« Graphinstanzen
- Datensatz enthält insgesamt 1 127 Graphinstanzen:
  - 139 gerichtete Graphen
  - 193 ungerichtete Graphen
  - 795 partiell gerichtete Graphen

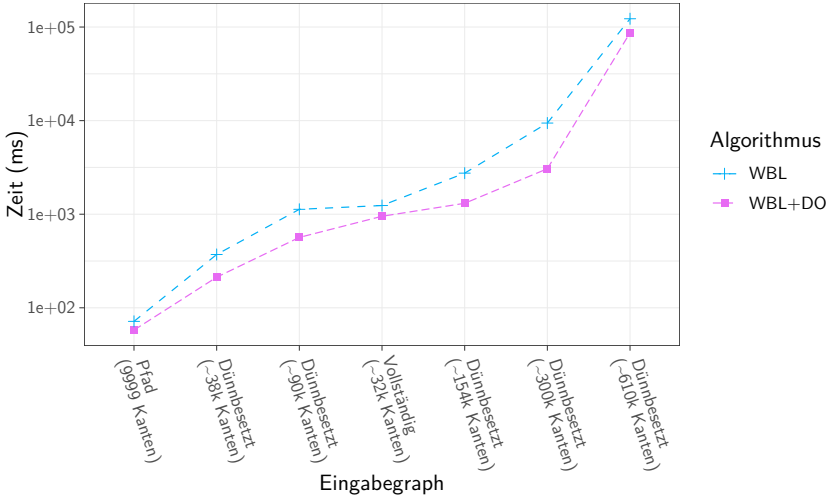
Ergebnisse

Partiell gerichtete Graphen mit ca. 50% gerichteten Kanten



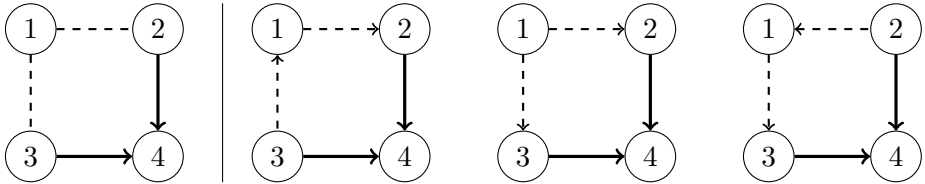
Ergebnisse

Auswirkung der Reihenfolge von Kanteneinfügungen

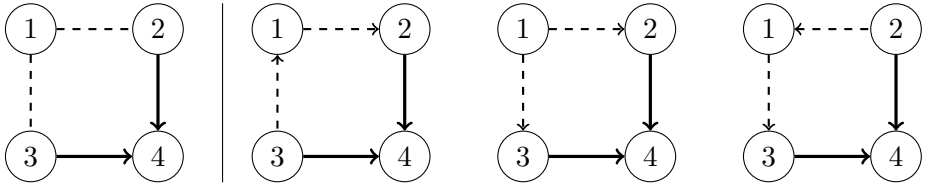


Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Datensatzgenerierung
Ergebnisse
Enumeration von Erweiterungen
Zusammenfassung

Enumeration von Erweiterungen



Enumeration von Erweiterungen



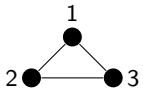
Das Enumerationsproblem

Eingabe: Ein partiell gerichteter Graph G.
Ausgabe: Alle konsistenten DAG-Erweiterungen von G.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Enumeration von Erweiterungen
Das Enumerationsproblem
Algorithmen zum Lösen des Enumerationsproblems
Zusammenfassung

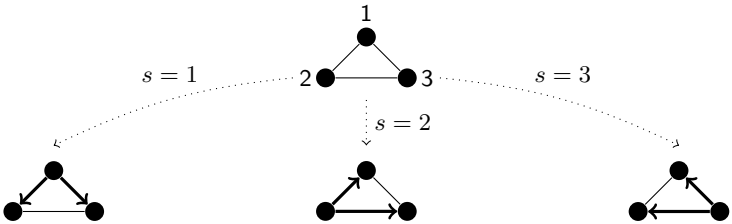
Enumeration für CPDAGs

- Idee: Rekursives Berechnen sogenannter s-Orientierungen
- Beispiel:



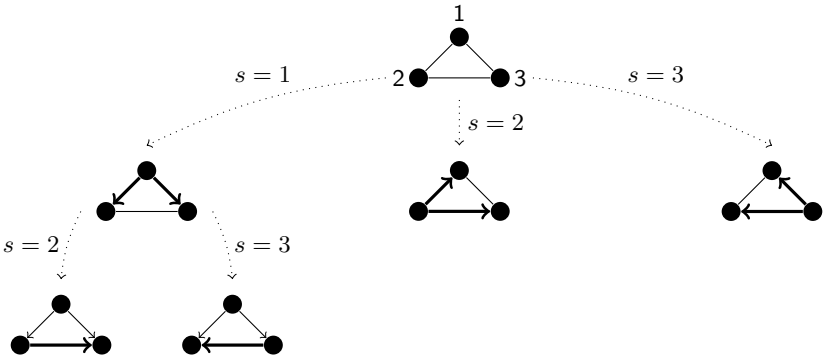
Enumeration für CPDAGs

- Idee: Rekursives Berechnen sogenannter s-Orientierungen
- Beispiel:



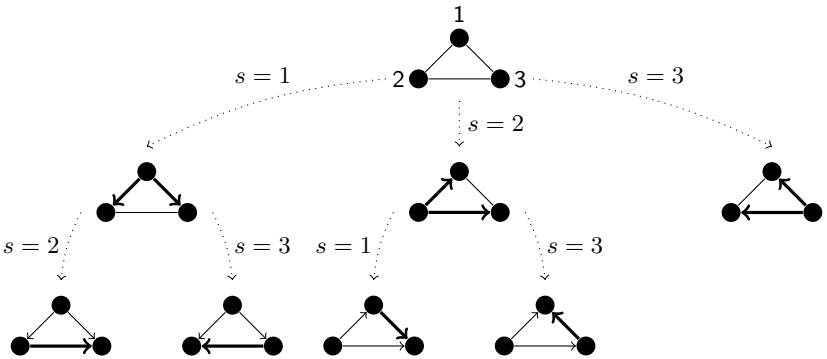
Enumeration für CPDAGs

- Idee: Rekursives Berechnen sogenannter s-Orientierungen
- Beispiel:



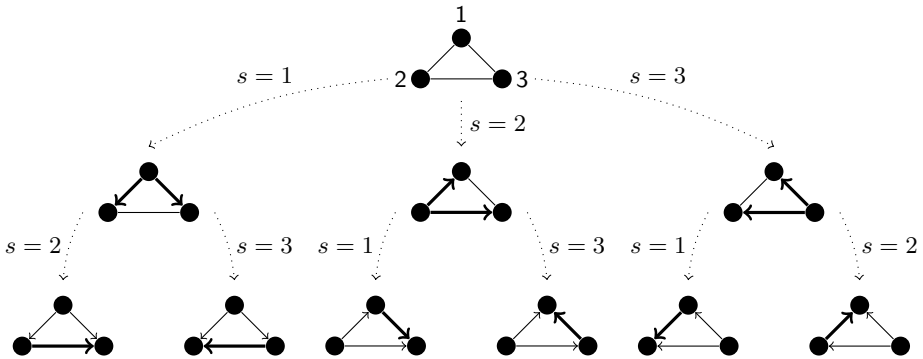
Enumeration für CPDAGs

- Idee: Rekursives Berechnen sogenannter s-Orientierungen
- Beispiel:



Enumeration für CPDAGs

- Idee: Rekursives Berechnen sogenannter s-Orientierungen
- Beispiel:



Erweitern kausaler Graphen

Malte Luttermann

Motivation

Erweiterungen für Graphen

Evaluation

Enumeration von Erweiterungen

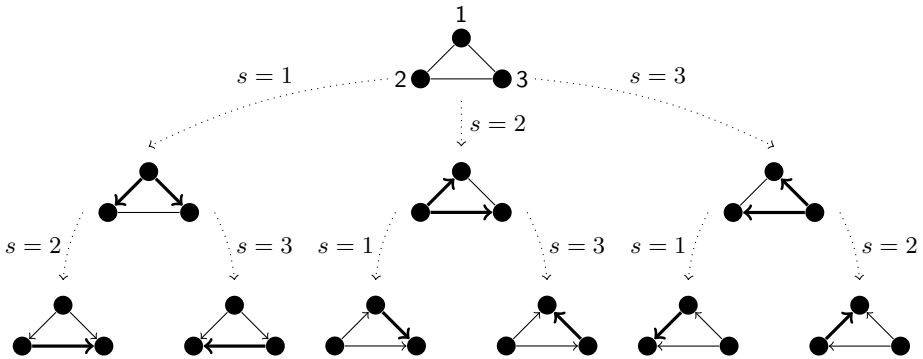
Das Enumerationsproblem

Algorithmen zum Lösen des Enumerationsproblems

Zusammenfassung

Enumeration für CPDAGs

- Idee: Rekursives Berechnen sogenannter s-Orientierungen
- Beispiel:



- Verzögerung zwischen zwei Ausgaben ist durch $O(nm)$ begrenzt

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Enumeration von Erweiterungen
Das Enumerationsproblem
Algorithmen zum Lösen des Enumerationsproblems
Zusammenfassung

Zusammenfassung

- DT ist nicht zuverlässig, da Performance sehr stark schwanken kann

Zusammenfassung

- DT ist nicht zuverlässig, da Performance sehr stark schwanken kann
- WBL+DO ist stabil, d.h. Performance ist unabhängig von Reihenfolgen

Zusammenfassung

- DT ist nicht zuverlässig, da Performance sehr stark schwanken kann
- WBL+DO ist stabil, d.h. Performance ist unabhängig von Reihenfolgen
- Heuristik (DT+H) liefert sehr gute Performance in der Praxis

Zusammenfassung

- DT ist nicht zuverlässig, da Performance sehr stark schwanken kann
- WBL+DO ist stabil, d.h. Performance ist unabhängig von Reihenfolgen
- Heuristik (DT+H) liefert sehr gute Performance in der Praxis
- Enumeration mit Verzögerung in $O(nm)$ für CPDAGs

Anhang

Der Algorithmus DT

Input: *A partially directed graph* $G = (V, E, A)$
Output: *A consistent DAG extension of* G *if* G *admits an extension, else a negative answer*

result $\leftarrow$ *copy of* G
temp $\leftarrow$ *copy of* G
while there are vertices left in temp do
  $s \leftarrow$ *any vertex in temp that satisfies the following properties*:
    1. *s has no arc directed outward from s*
    2. *all undirected neighbors of s are adjacent to all other neighbors of s*

  if no vertex s satisfying the properties was found then
    return a negative answer
  end

  orient all undirected edges incident to s towards s in result
  remove s and all incident edges from temp
end
return result

Der Algorithmus WBL

Input: *A partially directed graph* $G = (V, E, A)$
Output: *A consistent DAG extension of* G *if* G *admits an extension, else a negative answer*

result $\leftarrow$ *copy of* G
temp $\leftarrow$ *empty initialization of the data structure*
foreach edge e *in* G *do*
  insert e *into temp and update the data structure accordingly*
end
$L \leftarrow$ *list of potential sinks in temp*
while L *is not empty do*
  $s \leftarrow$ *pop any element from* L
  orient all undirected edges incident to s *towards* s *in result*
  remove s *and all incident edges from temp and update the data structure accordingly*
  add all previous neighbors of s *to* L *that became a potential sink in temp*
end
if there are edges left in temp then
  return a negative answer
end
return result

Der Algorithmus WBL+DO (auszugsweise)

Input: *A partially directed graph* $G = (V, E, A)$
Output: *A consistent DAG extension of* G *if* G *admits such an extension, else a negative answer*

result $\leftarrow$ *copy of* G
temp $\leftarrow$ *empty initialization of the data structure*
vertex_order $\leftarrow$ *degeneracy ordering of* G
foreach vertex v *in vertex_order do*
  foreach neighbor u *of* v *in* G *do*
    if u *precedes* v *in vertex_order then*
      insert the edge between u *and* v *into temp and update the data structure accordingly*
    end
  end
end
$L \leftarrow$ *list of potential sinks in temp*
...

Der Algorithmus DT+P

Input: *A partially directed graph* $G = (V, E, A)$
Output: *A consistent DAG extension of* G *if* G *admits an extension, else a negative answer*

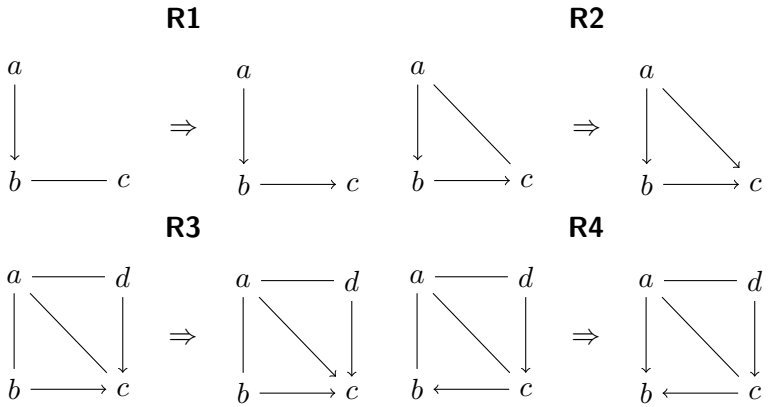
result $\leftarrow$ *copy of* G
temp $\leftarrow$ *copy of* G
$L \leftarrow$ *list of potential sinks in* G
while there are vertices left in temp do
  if L *is empty then*
    return a negative answer
  end
  $s \leftarrow$ *pop any element from* L
  orient all undirected edges incident to s *towards* s *in result*
  remove s *and all incident edges from temp*
  add all previous neighbors of s *in temp that became a potential sink to* L
end
return result

Der Algorithmus DT+H

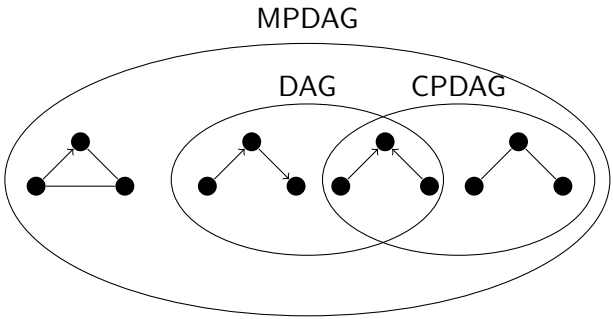
Input: *A partially directed graph* $G = (V, E, A)$
Output: *A consistent DAG extension of* G *if* G *admits an extension, else a negative answer*

result $\leftarrow$ *copy of* G
temp $\leftarrow$ *copy of* G
$L \leftarrow$ *list of hash sets with* $L[i] = \{v \in V \mid \Delta(v) = i\}$
while there are vertices left in temp do
  foreach set S *in* L *do*
    $s \leftarrow$ *any vertex from* S *that satisfies the following properties*:
      1. *s has no arc directed outward from s*
      2. *all undirected neighbors of s are adjacent to all other neighbors of s*
    if a vertex s satisfying the properties was found then
      break
    end
  end
  if no vertex s satisfying the properties was found then
    return a negative answer
  end
  orient all undirected edges incident to s towards s in result
  remove s and all incident edges from temp
  update the position in L for all previous neighbors of s in temp
end
return result

Regeln von Meek



Beziehungen zwischen MPDAGs, DAGs und CPDAGs



Das Erdős–Rényi Modell

Input: *An integer* $n \geq 0$ *and an integer* m *with* $0 \leq m \leq \frac{n \cdot (n-1)}{2}$
Output: *A randomly generated graph with* n *vertices and* m *edges*

$G \leftarrow$ *empty graph with* n *vertices*
while G *contains less than* m *edges do*
  $u \leftarrow$ *uniformly chosen random vertex in* G
  $v \leftarrow$ *uniformly chosen random vertex in* G
  if u *is different from* v *and the edge* $\{u, v\}$ *is not already in* G *then*
    insert the edge $\{u, v\}$ *into* G
  end
end
return G

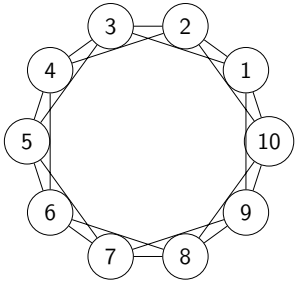
Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

Das Watts–Strogatz Modell

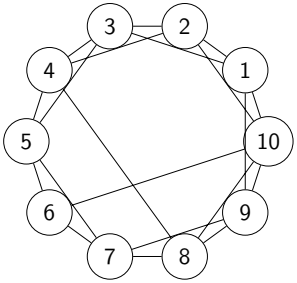
Input: *An integer* $n \geq 0$, *an integer* k *with* $0 \leq k < n$, *and a real number* $0 \leq \beta \leq 1$
Output: *A randomly generated graph with* n *vertices and* $n \cdot \lfloor k/2 \rfloor$ *edges*

if k *is odd then*
  $k \leftarrow k-1$
end
$G \leftarrow$ *regular ring lattice with* n *vertices where each vertex has* $k/2$ *neighbors on each side*
foreach vertex u *in* G *do*
  foreach neighbor v *of* u *in* G *do*
    rewire the edge $\{u, v\}$ *with probability* β:
      1. *delete the edge* $\{u, v\}$ *from* G
      2. *insert the edge* $\{u, w\}$ *for some uniformly random chosen vertex* w *that is not already connected to* u *into* G
  end
end
return G

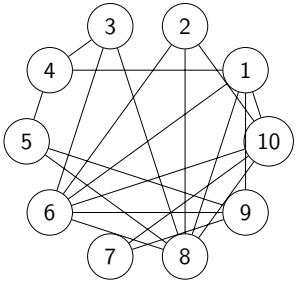
Beispiel für ein »Ring Lattice«



(1)



(2)



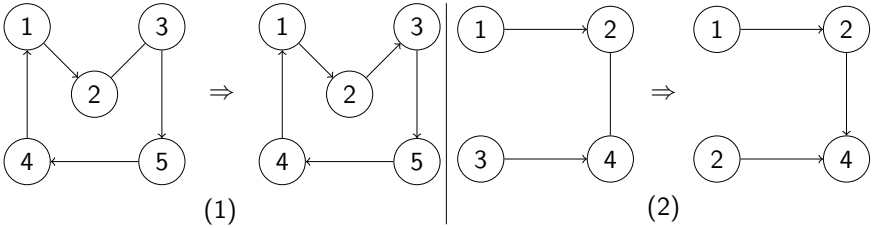
(3)

Das Barabási–Albert Modell

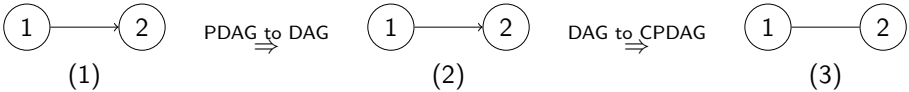
Input: *An integer* $n \geq 0$ *and an integer* k *with* $1 \leq k < n$
Output: *A randomly generated graph with* n *vertices and* $(n - k) \cdot k$ *edges*

$G \leftarrow$ *empty graph with* k *vertices*
insert a new vertex s *into* G *and connect* s *to every existing vertex in* G *via an edge*
for $u \leftarrow k{+}2$ *to* n *do*
  select k *existing vertices from* G*; each vertex* v *is chosen with probability* $\Pi(v) = \frac{\Delta(v)}{\sum_{w \in V} \Delta(w)}$
  insert the new vertex u *into* G
  insert edges between u *and each of the chosen vertices into* G
end
return G

Anwendung von Meek's Regeln auf nicht erweiterbare Graphen



PDAG $\rightarrow$ DAG $\rightarrow$ CPDAG



(1) (2) (3)

Enumeration für PDAGs

Input: *A partially directed graph* $G = (V, E, A)$
Output: $CE(G)$

$\mu \leftarrow$ *number of* $v-$*structures in* G
call enumerate(G)

procedure enumerate(G)
  if G *contains only directed edges then*
    if G *is acyclic and number of* $v-$*structures in* $G = \mu$ *then*
      output G
    end
    exit
  end
  G' $\leftarrow$ *graph obtained by applying Meek's rules exhaustively on* G
  if G' *is acyclic and number of* $v-$*structures in* $G' = \mu$ *then*
    $\{u, v\} \leftarrow$ *any undirected edge in* G'
    G' $\leftarrow$ *graph obtained by orienting* $\{u, v\}$ *as* $u \rightarrow v$
    G'' $\leftarrow$ *graph obtained by orienting* $\{u, v\}$ *as* $v \rightarrow u$
    call enumerate(G')
    call enumerate(G'')
  end
end

Enumeration für CPDAGs

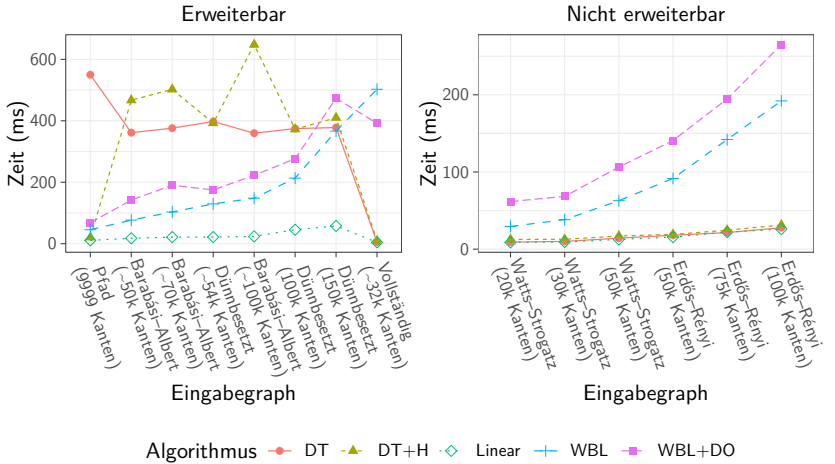
Input: *A CPDAG* $G = (V, E, A)$
Output: $CE(G)$

call enumerate(G)

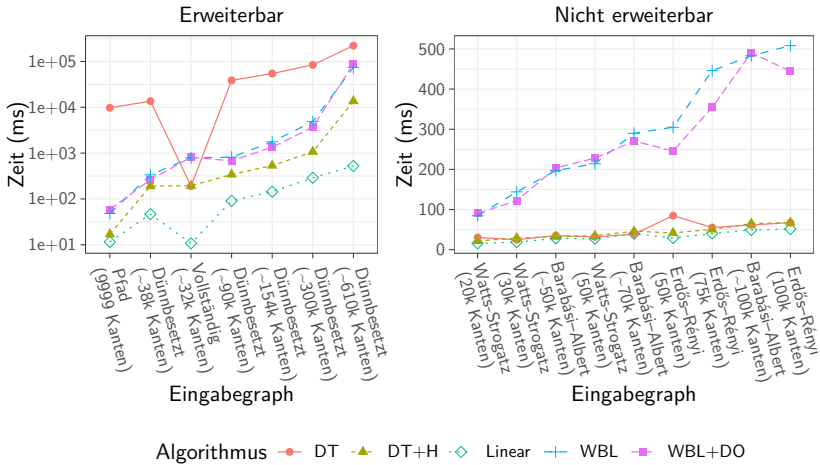
procedure enumerate(G)
  if G *contains only directed edges then*
    output G
    exit
  end
  foreach vertex s *in* G *do*
    if s *is incident to at least one undirected edge in* G *then*
      G' ← *graph obtained by computing the* $s-$*orientation in* G
      call enumerate(G')
    end
  end
end

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

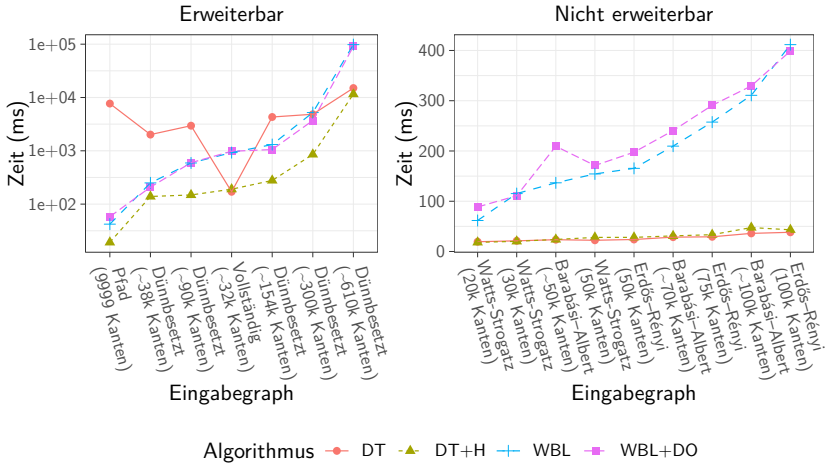
Ergebnisse für gerichtete Graphen



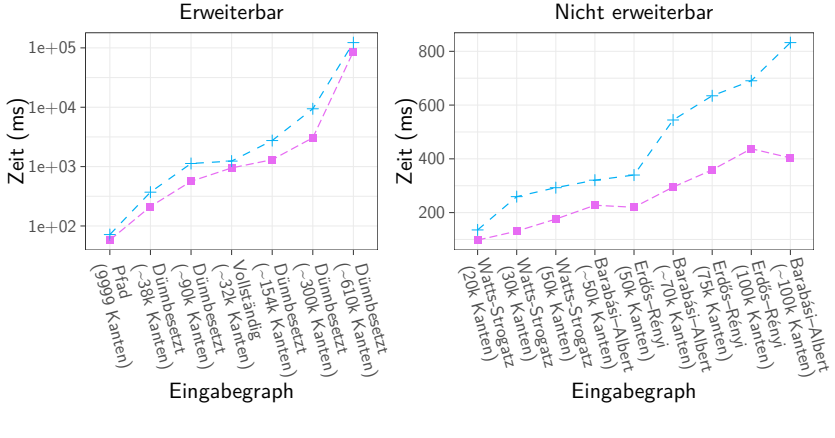
Ergebnisse für ungerichtete Graphen



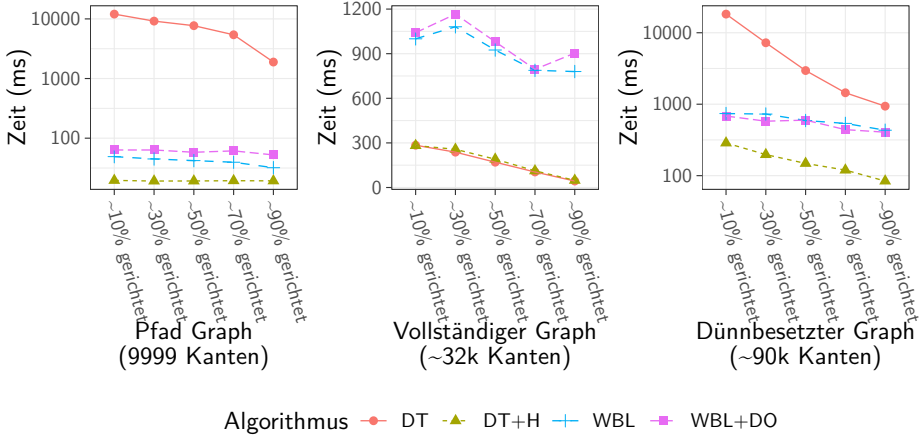
Ergebnisse für partiell gerichtete Graphen



Auswirkung der Reihenfolge von Kanteneinfügungen



Auswirkung verschiedener Anteile an gerichteten Kanten



Enumeration für beliebige partiell gerichtete Graphen

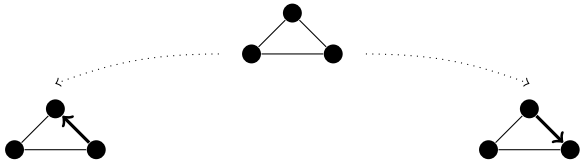
- Idee: Richte eine ungerichtete Kante in beide Richtungen, wende jeweils die Regeln von Meek an und wiederhole rekursiv für beide Richtungen.[3]

[3] Christopher Meek. »Causal Inference and Causal Explanation with Background Knowledge«. In: *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. UAI'95. Montréal, Qué, Canada: Morgan Kaufmann Publishers Inc., 1995, S. 403–410.

Erweitern kausaler Graphen
Malte Luttermann
Motivation
Erweiterungen für Graphen
Evaluation
Enumeration von Erweiterungen
Zusammenfassung

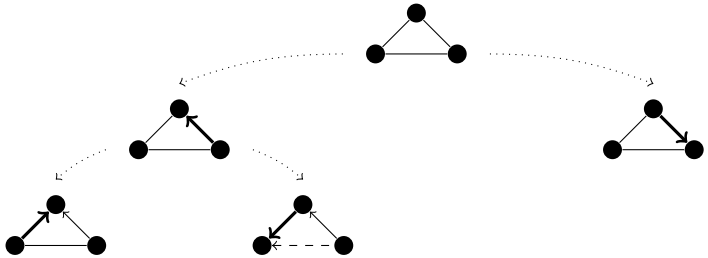
Enumeration für beliebige partiell gerichtete Graphen

- Idee: Richte eine ungerichtete Kante in beide Richtungen, wende jeweils die Regeln von Meek an und wiederhole rekursiv für beide Richtungen.
- Beispiel:



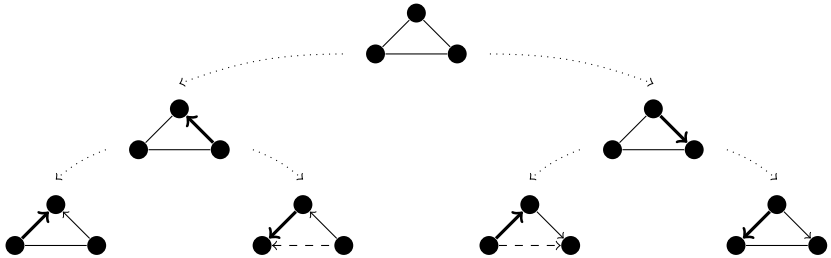
Enumeration für beliebige partiell gerichtete Graphen

- Idee: Richte eine ungerichtete Kante in beide Richtungen, wende jeweils die Regeln von Meek an und wiederhole rekursiv für beide Richtungen.
- Beispiel:



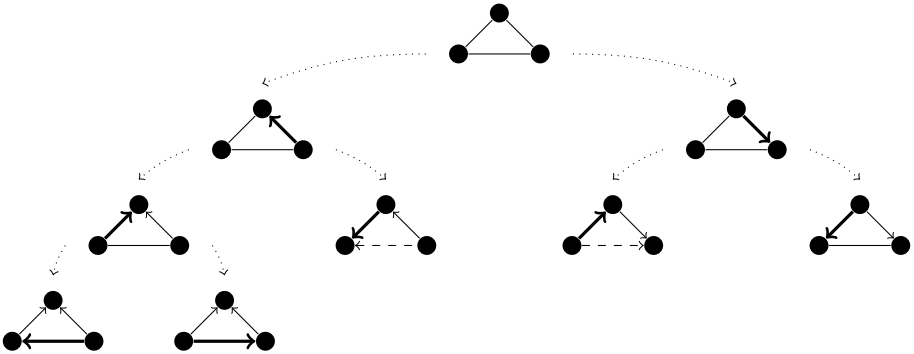
Enumeration für beliebige partiell gerichtete Graphen

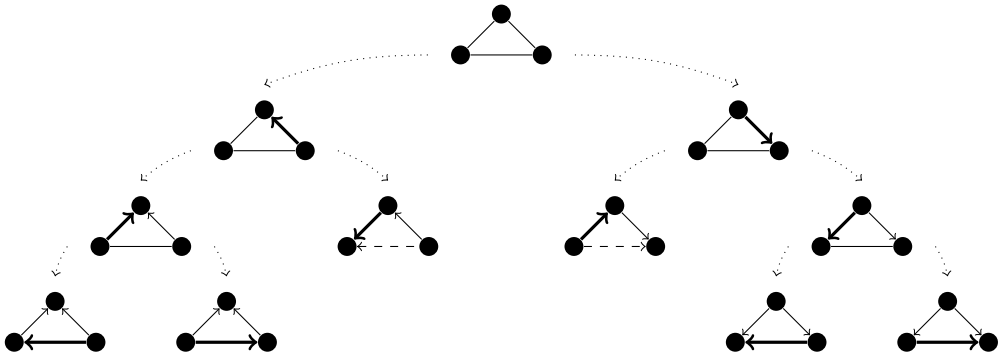
- Idee: Richte eine ungerichtete Kante in beide Richtungen, wende jeweils die Regeln von Meek an und wiederhole rekursiv für beide Richtungen.
- Beispiel:



Enumeration für beliebige partiell gerichtete Graphen

- Idee: Richte eine ungerichtete Kante in beide Richtungen, wende jeweils die Regeln von Meek an und wiederhole rekursiv für beide Richtungen.
- Beispiel:



Enumeration für beliebige partiell gerichtete Graphen

- Idee: Richte eine ungerichtete Kante in beide Richtungen, wende jeweils die Regeln von Meek an und wiederhole rekursiv für beide Richtungen.
- Beispiel: