

# Constraint Processing mit Hilfe von Baumzerlegung

Sebastian Wenzel

# Gliederung

- Einführung
  - Begriffe
- Acyclic Networks
  - Lösen
  - Erkennen
- Bilden von Acyclic Networks
  - Clustering

# Einführung

- Hypergraphen

Graph bestehend aus Knoten und Hyperkanten

Hyperkanten können mehr als 2 Knoten verbinden

Hyperkanten entsprechen den Constraints

- Primal Graph

Wird aus dem Hypergraphen gebildet

Knoten Bleiben

Hyperkanten werden durch mehrere Kanten ersetzt

# Einführung

- Dual Graph

Wird aus Hypergraph gebildet

Hyperkanten werden zu Knoten

Kanten entstehen wenn im Hypergraphen zu einem Knoten mehrere Hyperkanten gehörten

# Einführung

- Join-Graph

Entspricht dem Dual Graph

Muss die Eigenschaft der Connectedness erfüllen

- Connectedness

Jedes Knotenpaar mit einer gemeinsamen Variable

Muss mindest einen Weg haben der die Variable durchgehend enthält

# Einführung

- Join-tree

Join-Graph der die eigenschaften eines Baums hat

- Hypertree

Ein Hypergraph aus dessen Dual Graphen ein Join-tree gebildet werden konnte

- acyclic network

Ein Constraint Problem dessen Hypergraph ein Hypertree entspricht

# Acyclic Networks lösen

Lösen der Probleme durch verwenden des join-trees

Mit hilfe des Acyclic-Solving Algorithmus

- Abarbeitung der Knoten von den Blättern bis zur Wurzel
- Verringern der gültigen Lösungen des Vaterknotens
- Bei gefundener Lösung auffüllen der in der Wurzel unbekanntenen Variablen

# Acyclic Networks lösen

Acyclic-Solving Algorithmus :

1.

Sortiere die Constraints nach den join-tree

$\{R_1, \dots, R_r\}$

$R_1$  = Wurzel

Folgend die Kinderknoten des join-trees

# Acyclic Networks lösen

Acyclic-Solving Algorithmus :

2.

Beginne bei den Kindknoten und verringere die Menge der Gültigen Lösungen der Elternknoten

Falls leere Menge entsteht gibt es keine Lösung

# Acyclic Networks lösen

Acyclic-Solving Algorithmus :

3.

Traversiere den Baum von der Wurzel

Füge der gültigen Teillösung die Werte der noch nicht vorhandenen Parameter hinzu

# Acyclic Networks erkennen

Wie bilde ich aus einem Hypergraphen einen join-tree

2 Alternativen :

- Ausgehend vom Dual Graph
- Ausgehend vom Primal Graph

# Acyclic Networks erkennen

Alternative 1:

- Den Dualgraph gewichten mit der Anzahl der gemeinsamen Variablen
- Bilden des maximal spannenden Baums
- Connectedness überprüfen

Komplexität ist  $O(n^3)$  mit  $n = \text{Anzahl Constraints}$

# Begriffe

- Chordal Graphs

Ein Graph ist Chordal wenn jeder Kreis mit der Länge 4 oder größer eine weitere Kante enthält die 2 Kanten des Kreises verbindet

- Ordered Graph

In einem Ordered Graph liegen die Knoten geordnet vor  
Alle vorangehenden Knoten die mit dem Knoten  $k$  verbunden sind heißen Elternknoten von  $k$

# Begriffe

- Cliques

Ein vollständiger Untergraph

- Conformality

Diese Eigenschaft ist erfüllt, wenn die maximalen cliques im Primary Graph und die Constraints identisch sind

# Acyclic Networks erkennen

Alternative 2:

Ein Hypergraph besitzt einen Join-Tree wenn  
der Primal Graph chordal und conformal ist

# Acyclic Networks erkennen

Alternative 2:

Ein Graph ist chordal wenn in seiner „maximal cardinality ordering“ alle Elternknoten untereinander Verbunden sind (cliquen bilden)

Im chordal Graph können wir die cliques in linearer Zeit finden durch auflisten der Knoten mit ihren Elternknoten

# Acyclic Networks erkennen

Alternative 2:

- Test chordality: Mit „max-cardinality ordering“
- Finden der maximalen cliquen
- Test conformality: Sind die cliquen gleich zu den constraints
- Bauen des Join-trees

Komplexität ist  $O(n^2)$

# Acyclic Networks bilden

- Umwandlung von beliebigen Graphen
- Bilden von Clustern aus den vorhanden Constraints
- Ergebnisse sind neue Constraints eines Acyclischen Netzwerkes

# Acyclic Networks bilden

Join Tree Clustering:

Erweitern des 2ten Ansatzes zum Finden

Anstelle abzurechnen

Graphen so anpassen das join-tree entsteht

Entstandene Subprobleme Lösen

# Acyclic Networks bilden

Join Tree Clustering Anpassungen:

Max cardinality ordering aufbauen.

Fehlende Kanten hinzufügen bis chordality erfüllt ist

Lösung: cliques herstellen mit den Elternknoten

Problem: conformality ist nicht mehr gegeben

# Acyclic Networks bilden

Join Tree Clustering Anpassungen:

Problem: conformality ist nicht mehr gegeben

Bilden neuer Constraints aus den maximalen cliquen

Ergebnisse der neuen Constraints bilden Grundlage für  
den join-tree

# Acyclic Networks bilden

Komplexität  $O(r \cdot k^{w^*(d)+1})$

Mit :

$r$  = Anzahl der Constraints

$k$  = Maximale Domain Größe

$w^*(d)$  = induced width des ordered Graph

# Zusammenfassung

- Acyclic Networks
- Lösen
- Erkennen
  - Dual
  - Primal
- Bilden