Karlsruher Institut für Technologie

A Linear-Time Algorithm For Finding Tree-Decompositions Of Small Treewidth

Maximilian F. Göckel – *uzkns@student.kit.edu*

Institut für Theoretische Informatik - Proseminar Algorithmen für NP-schwere Probleme

Motivation

Viele (NP-schwere) Graphenprobleme sind auf Graphen, bei denen eine Baumzerteilung mit Baumweite max. k (k Konstante) gegeben ist, in Linearzeit lösbar.

Tree-decomposition: Definition

Eine Baumzerteilung eines Graphen $G = (V, E)$ ist ein Tupel (X, T) wo $T = (I, F)$ ein Baum ist und $X = \{X_i | i \in I\}$ eine Familie von Teilmengen von V wobei jedes X_i einen Knoten in T darstellt.

1. $\bigcup_{i \in I} X_i = V$
2. $\forall (v, w) \in E : \exists i \in I : v, w \in X_i$
3. $\forall w \in X_i, X_j$: Jedes X_k im Pfad zwischen X_i, X_j enthält w

Tree-decomposition: Beispiel

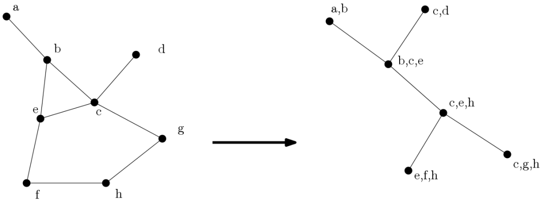

Figure: 1

Treewidth

Jede Baumzerteilung hat eine "Baumweite" (treewidth).

- Baumweite einer Zerteilung: $\max(|X_i|_{i \in I} - 1)$ ("Zerteilungsweite")
- Baumweite eines Graphen: Minimale Zerteilungsweite aller Zerteilungen

Hauptidee

Viele NP-schwere Probleme sind in Linearzeit lösbar, wenn die Baumweite des Graphen konstant ist. $\rightarrow$ Kann man die Baumweite (für bel., festes $k \in \mathbb{N}$) in Linearzeit errechnen?

ZZ: $\forall k \in \mathbb{N} : \exists$ Linearzeitalgorithmus welcher für $G = (V, E)$ prüft ob die Baumweite max. k ist und eine Zerteilung ausgibt.

Für $k = 1, 2, 3, 4$ existieren schon Linearzeitalgorithmen

Algorithmus

2 Schritte:

1. Für gegebenen Graph $G = (V, E)$ und geg. $k \in \mathbb{N}$ eine Zerteilung mit max. Baumweite linear in k finden
2. Graph-Zugehörigkeit zur Klasse "Graphen mit Baumweite k" prüfen

Problem *"Für einen Graph $G = (V, E)$ und ein $k \in \mathbb{N}$: Ist die Baumweite von G maximal k?"* ist NP-Vollständig für bel. k