Aufgabe 4.3

T. Adam, M. ben Ahmed

Universität Osnabrück

Æ

January 11, 2021

Aufgabe 4.3 - KCT Heuristiken

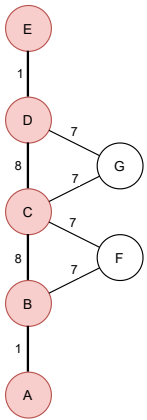
Heuristics for the K-Cardinality Tree and Subgraph Problems (1997)

- M. Ehrgott, J. Freitag, H. W. Hamacher, F. Maffiali †
- 3 Klassen von Heuristiken für EKCT
- 2 Klassen von Heuristiken für NKCT
- Umwandlung von NKCT $\Rightarrow$ EKCT

Greedy Klasse

MST basierte Heuristiken

- Prim mit $|E(mst)| = k$
- Kruskal mit mehreren Zusammenhangskomp.
- Kruskal mit einer Zusammenhangskomp.



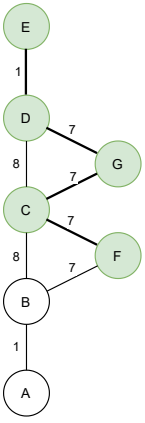
k = 4

$c^* = 18$

Greedy - k-CardPrim

k-CardPrim

- Starte Alg. von Prim für alle Knoten
- Breche ab, sobald MST $k + 1$ Knoten enthält
- Gebe günstigsten MST zurück



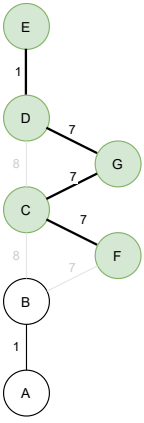
k = 4

$c = 22$

Greedy - DualGreedy1

DualGreedy1

- Starte mit Ausgangsgraph
- Lösche teuerste Kante
- Wiederhole, solange min. eine Zhk. mit min. $k + 1$ Knoten existiert



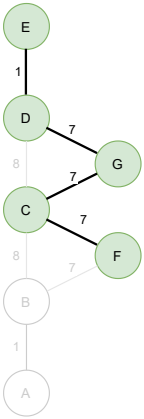
$k = 4$

$c = 22$

Greedy - DualGreedy2

DualGreedy2

- Starte mit Ausgangsgraph
- Lösche teuerste Kante inkl. Knoten, deren Löschung keine weiteren Zhk. erzeugt
- Wiederhole, bis $k + 1$ Knoten im Graph bleiben



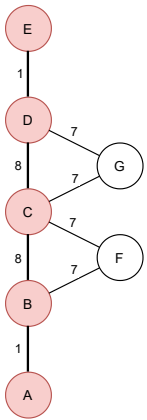
k = 4

$c = 22$

Path Klasse

Shortest-Path basierte Heuristiken

- Dijkstra mit max. Pfadlänge k (Gewichte)
- Dijkstra mit max. Pfadlänge k (Kardinalität)
- Erweitere Pfade aus Dijkstra $< k$ mit Prim
- Führe beide Dijkstra Alg. aus und wähle beste Lösung



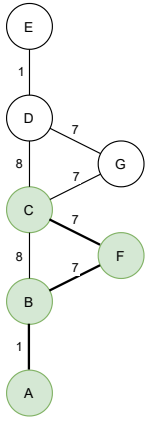
$k = 4$

$c^* = 18$

Path - k-DijkstraA(B)

k-DijkstraA(B)

- Dijkstra mit max. Pfadlänge k von Knoten $s \Rightarrow x$
- Führe zusätzlich `len[]` und `dist[]` ein
- Speichere Länge und Gewicht des bisher besten Pfades
- (A) `dist[x]` wird aktualisiert, wenn Gesamtgewicht echt kleiner wird
- (B) `dist[x]` wird aktualisiert, wenn Gesamtgewicht kleiner oder Länge größer wird
- Füge Knoten mit minimalen `dist[]` am Ende des Pfades hinzu solange $< k$ Kanten

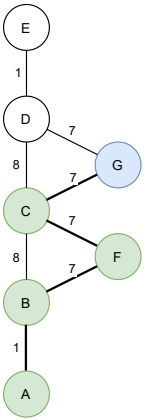


k = 4

Path - DijkstraPrim-A(B)

DijkstraPrim-A(B)

- Wende k-DijkstraA(B) für alle Knoten an
- Erweitere Pfade kürzer als k mit Prim
- Wähle beste Lösung



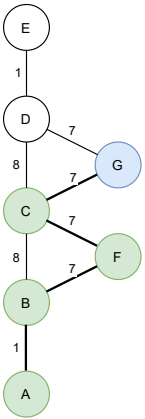
$k = 4$

$c = 22$

Path - DijkstraPrim

DijkstraPrim

- Führe DijkstraPrim-A aus
- Führe DijkstraPrim-B aus
- Wähle beste Lösung



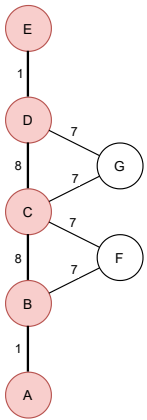
$k = 4$

$c = 22$

DynamicTree Klasse

Dynamic Programming basierte Heuristiken

- DynamicTree findet garantiert einen optimalen KCT in gegebenen MST
- Erzeuge MST mit Prim und wende DT an
- Erzeuge MST durch DijkstraPrim erweitert durch Prim
- Führe beide Dijkstra Alg. aus und wähle beste Lösung



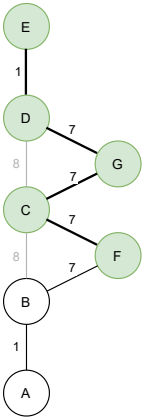
k = 4

$c^* = 18$

DynamicTree - DynamicTreePrim

DynamicTreePrim

- Finde MST mit Prim
- Führe Dynmic Tree auf MST aus



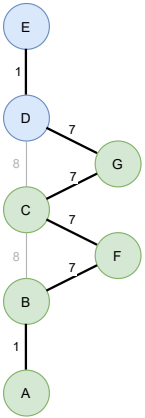
$k = 4$

$c = 22$

DynamicTree - DynamicDijkstraPath

DynamicDijkstraPath

- Führe DijkstraPrim aus
- Erweitere Ergebnis mit Prim zu MST
- Führe Dynamic Tree auf MST aus



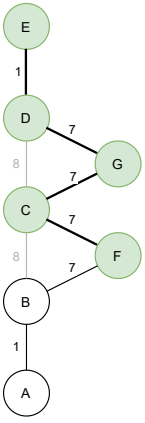
$k = 4$

$c = 22$

DynamicTree - DynamicDijkstraTree

DynamicDijkstraTree

- Führe k-Dijkstra-A(B) aus
- Erweitere Pfade mit Prim zu MSTs
- Führe Dynamic Tree auf MSTs aus



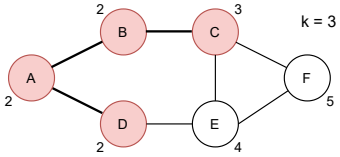
$k = 4$

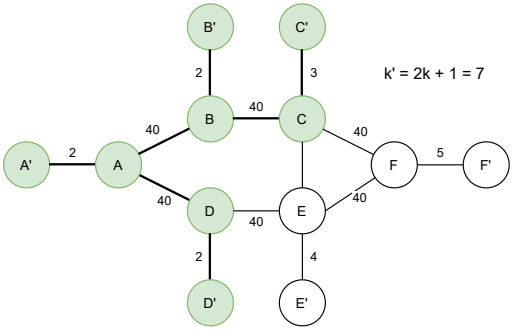
$c = 22$

NKCT $\rightarrow$ EKCT

NKCT(G) $\rightarrow$ EKCT(G')

- Verdoppele alle Knoten in G
- Kantengewicht zwischen verdoppelten Knoten entspricht Knotengewicht
- Sonstige Kantengewichte: $(|E| + 1) \cdot w_{max}$
- EKCT mit $k' = 2k + 1$ auf G' anwenden



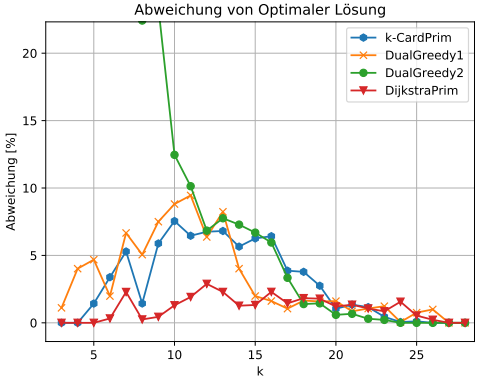


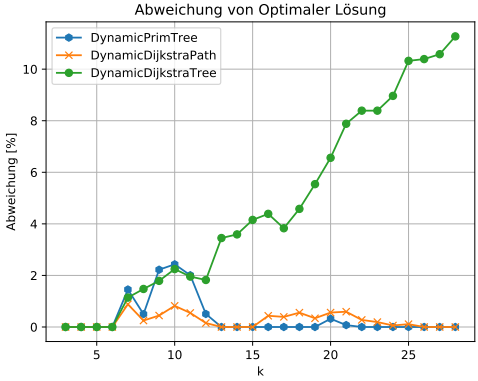
Tests

Laufzeiten und Instanzen

- Komplexität der Algorithmen in $\mathcal{O}(n^3)$
- Außer DynamicTree und DualGreedy in $\mathcal{O}(n^4)$
- Alle Instanzen lösbar innerhalb von Sekunden
- 20 Random Graphen mit jeweils $[10, 20, 30]$ Knoten
- Zufällige generierte Kantengewichte (exponenziell-, gleich- und normalverteilt)
- Getestete k: $3 \leq k \leq n - 2$
- Relativ große k realistischer als kleine k: Ölfeld Bsp. $k = \frac{n}{2}$

Ergebnisse





Genereller Graph mit 30 Knoten