UNIVERSITÄT LEIPZIG

Algorithmen und Datenstrukturen II

Vorlesung шест6

Leipzig, 07.05.2024

Peter F. Stadler & Thomas Gatter & Ronny Lorenz

GOMORY HU

Intro: Gomory Hu Bäume I

Wissensbasis

Wir haben einen ***ungerichteten*** Graph $G = (V, E)$, mit $n = |V|$ Knoten und $m = |E|$ Kanten. Gegeben Quelle $s \in V$ und Senke $t \in V$.

Aus der vorigen Vorlesung wissen wir, dass max-flow und min-cut denselben Wert haben (Max-flow min-cut Theorem) und, dass wir dieses Problem effizient lösen können.

Die Definition der Cut Kapazität $c(A, B)$ in ungerichteten Graphen erfolgt analog zum gerichteten Fall betrachten.

Paarweises min-cut Problem:

Wir wollen für alle Paare $s, t \in V, s \neq t$ das min-cut Problem lösen.

Intro: Gomory Hu Bäume II

Gibt es einen effizienteren Weg als den *brute-force* Ansatz die $\binom{n}{2}$ min-cut Probleme zu lösen?

Antwort: Ja, und wir klären nun wie!

Wikipedia: Gomory-Hu Tree *(Link)*

Gomory Hu Bäume

Definition

Ein *Schnitt-Baum* oder **Gomory-Hu Baum** $T^{\#}$ von (G, c) ist ein Spann-Baum mit Kantengewichten α und der Eigenschaft, dass für jedes Paar von Knoten s, t eine Kante e_{st} entlang des Pfades zwischen s und t existiert, sodass der Schnitt (A, B) der durch Entfernen von e_{st} entsteht, ein minimaler Schnitt zwischen A und B ist.

Wenn $T^{\#}$ ein Gomory-Hu Baum für G ist, dann ist $\alpha_{st} = c(A, B)$ wobei (A, B) ein minimaler Schnitt zwischen A und B.

Hinweis: G und $T^{\#}$ sind ungerichtet.

Gomory Hu Bäume

Theorem

Jeder zusammenhängende Graph mit Kapazität $c(e) > 0$ für alle Kanten $e \in E$ hat einen Gomory-Hu Baum.

Der Beweis benutzt einfache Eigenschaften von Matroiden und deren Rangfunktion.

Mit tiefgehender Beweisführung:
Kombinatorische Optimierung, Korte, Vygen, Kap. 8.6

Berechnung von Gomory Hu Bäumen

Beobachtungen

- Die Kanten inzident mit $x \in V$ bilden einen Schnitt mit Kapazität $c(\{x\}, V \setminus \{x\}) = \sum_{(x,y) \in E} c(x, y)$
- Wird in G eine Kante $e = xy$ kontrahiert um den Knoten p im kontrahierten Graphen G' zu erzeugen, so gilt

$$c_G(\{x, y\}, V \setminus \{x, y\}) = c_{G'}(\{p\}, V \setminus \{x, y\})$$

  wobei $c_{G'}(p, u) = c_G(x, u) + c_G(y, u)$ für alle $u \in V \setminus \{x, y\}$ gesetzt wird.
- Kontrahiert man (schrittweise) alle Kanten innerhalb eines induzierten Teilgraphen $G[W]$ und ersetzt man W dann durch einen einzelnen Knoten p in P, dann gilt

$$c_G(W, V \setminus W) = c_{G'}(\{p\}, V \setminus W)$$

Gomory-Hu Algorithmus

Algorithmus

input : gewichteter ungerichteter Graph G
output : Gomory-Hu Tree

```
GomoryHu(G):
    if (|V| ≥ 2) then
        Wähle s und t (zufällig) in V
        (A, B) ← minimum-cut(s, t)
        konstruiere G_1 und G_2 durch Kontraktion von A und B
        GomoryHu(G_1); GomoryHu(G_2);
```

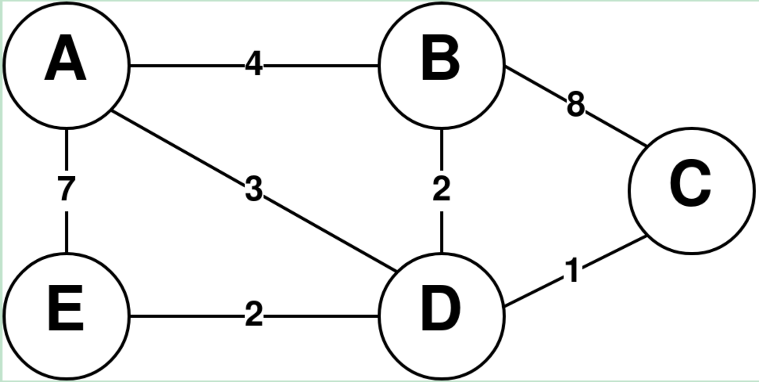
Jeder minimale Schnitt (A, B) enspricht einer Kante e im Gomory-Hu Baum. Das entsprechende Gewicht $\alpha(e)$ der Kante e ist gerade die Kapazität $c(A, B)$ dieses minimalen Schnittes.

Ziel des Algorithmus

Wichtig! Es gibt verschiedene Notationen. Wir vermeiden hier bewusst eine exakte Definition der Kontraktion zwischen A und B und geben im folgenden Beispiel zwei gängige Interpretationen.

- Wir beginnen mit allen Knoten in einer Menge $V = \{a, \ldots, n\}$.
- Jeder min-cut zwischen zwei zufälligen Knoten s und t gibt uns zwei Mengen A und B, sodass $s \in A$ und $t \in B$ und $A \cup B = V$:
  - die Kanten zwischen A und B definieren eine Kante im Gomory-Hu Baum
  - der Algorithmus wird auf Teilmengen A und B rekursiv angewendet
  - V wird so in A und B zerlegt
- Die Rekursion wird angewendet bis jeder Knoten als einzelne Menge betrachtet wurde: $\{a\}, \{b\}, \ldots, \{n\}$.
- Am Ende haben wir n Einzelmengen als Knoten und $n - 1$ Kanten zwischen diesen und damit den Gomory-Hu Baum gebaut.

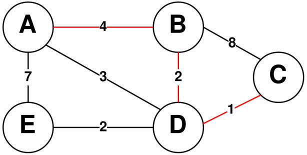
Beispiel

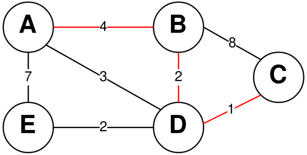


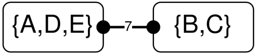
Wir wenden Gomory-Hu parallel in zwei Varianten auf den selben Graphen an.

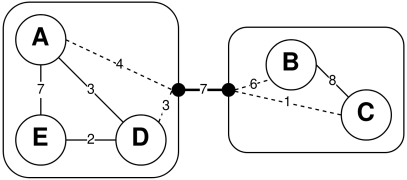
- Links:
  - Cuts werden im Originalgraphen bestimmt
  - Kontraktionen repräsentieren die Mengenzerlegung
  - Gomory-Hu Baum wird über Mengenzerlegung aufgebaut
- Rechts:
  - Cuts werden im aktuellen kontrahierten Graphen bestimmt
  - Kontraktionen werden explizit am Graphen ausgeführt
  - Gomory-Hu Baum entsteht über Kontraktionen im selben Graphen

Notation:
rot: der aktuelle Cut, *fett:* finale Kanten des Gomory-Hu Baums,
gestrichelt: Kanten die an kontrahierten Knoten anliegen;
die Boxen im rechen Teil werden nicht explizit erzeugt, sondern dienen hier nur der Illustration der Mengenzerlegung

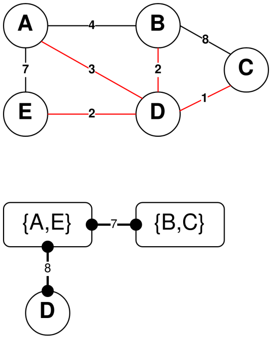


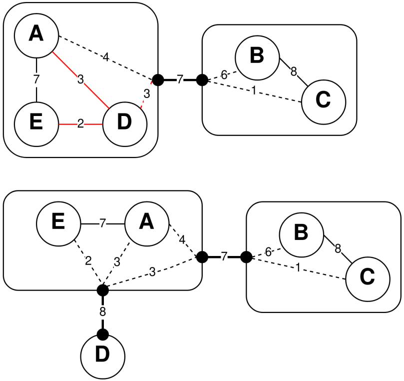




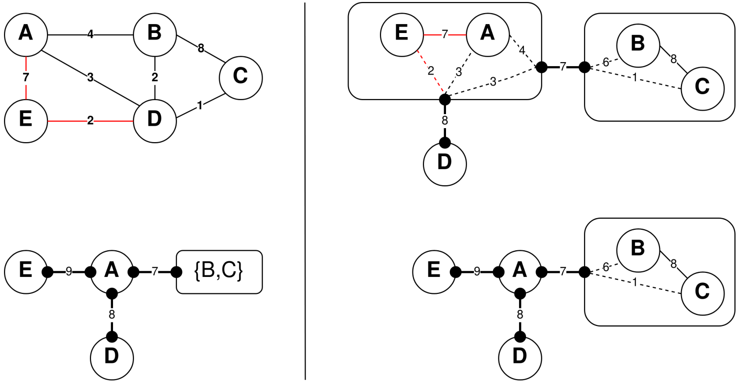


Wir wählen für dieses Beispiel immer die zwei kleinsten Elemente der Menge als s und t.
Der min-cut zwischen A und B ergibt sich als $\{A, D, E\}$ und $\{B, C\}$ mit Cut Wert 7.
Links: Der Gomory-Hu Baum wurde als ein einzelner Knoten mit der Menge $\{A, B, C, D, E\}$ initialisiert und durch den cut in zwei neue Knoten entsprechend der Mengenzerlegung unterteilt.
Rechts: Der jeweilige Restgraph wird durch neue kontrahierte Knoten abstrahiert.
Für beide wird eine Kante mit Wert 7 als finales Element des Gomory-Hu Baums eingefügt.

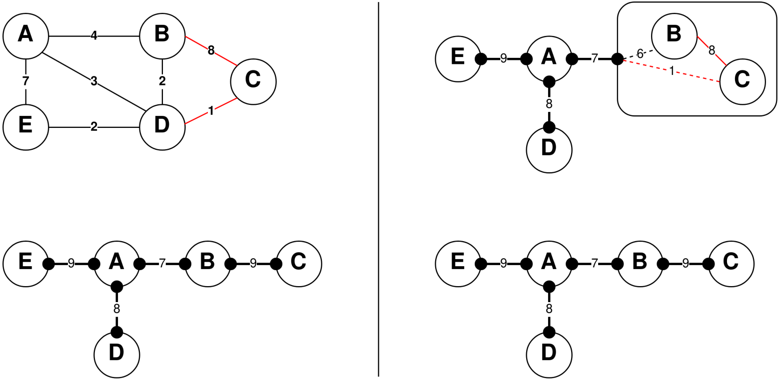




Der min-cut zwischen A und D ergibt sich als $\{A, E\}$ und $\{D\}$ mit Cut Wert 8.
Die Menge $\{D\}$ enthält nur noch ein einzelnes Element. Der Knoten D kann daher final in beide Gomory-Hu Bäume eingefügt werden.
Rechts: Finale Knoten werden in diesem Schritt mit allen anliegenden kontrahierten Knoten vereinigt.



Der min-cut zwischen A und E ergibt sich als $\{A\}$ und $\{E\}$ mit Cut Wert 9. Knoten A und E können finalisiert werden.



Der min-cut zwischen B und C ergibt sich als $\{B\}$ und $\{C\}$ mit Cut Wert 9.
Knoten B und C können finalisiert werden.
Wir erhalten zwei identische Gomory-Hu Bäume.

Ein komplexeres Beispiel

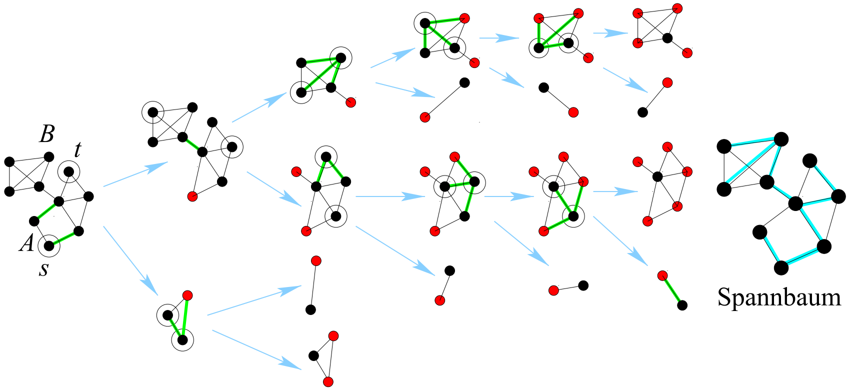
Beispiel 2

- In jedem Schritt markieren die eingekreisten Vertices unser s, t zwischen denen der min-cut bestimmt wird.
- Die grünen Kanten werden geschnitten, die von s aus erreichbaren (nicht über grüne Kanten gehen!) Vertices, landen in der Menge A.

$\rightarrow$ Für t dann äquivalent in B.

- Rote Vertices stellen jeweils eine Kontraktion des verbliebenen Graphen dar.
- Betrachten Sie insbesondere den Schritt nach unten, wo nur die beiden schwarzen Vertices aus A bestehen bleiben, der ganze verbliebene Graph aus B ist kontrahiert.

Graphische Darstellung des Gomory-Hu Algorithmus



Warum?

Reale Netzwerke (z.B. Verkehrs-, Computer-, Auftragsplanungs-, soziale oder biologische Netzwerke, . . .) haben nur selten einen klaren Start- und Endpunkt. Vielmehr interessiert uns die Verbindung zwischen allen (Paaren von) Knoten.

Außerdem erhalten wir so Anhaltspunkte zur “störanfälligkeit” von Netzwerken. Z.B. können wir nun beantworten wie viele Verbindungen in einem Netzwerk ausfallen müssen bevor ein a) beliebiger oder b) spezifischer Knoten abgeschnitten ist.