Multicommodity Flow

Steffen Kionke

Sommerakademie Görlitz

12. September 2007

Inhalt

Das ganzzahlige Mehrgüterfluss-Problem

Problem (IMF)

Gegeben:

- *ungerichteter Graph* $G = (V, E)$
- *Kantengewichte (genannt Kapazitäten)* $c_e \in \mathbb{N}$
- *Menge von Paaren von Knoten* $L = \{ (s_1, t_1), \ldots, (s_k, t_k) \}$

Ziel:
Für alle $i = 1, \ldots, k$ *soll ein Gut von* s_i *nach* t_i *transportiert werden.*
Der Güterfluss soll ganzzahlig maximiert werden ohne die Kapazitäten zu überschreiten.

Englisch: *Integer Multicommodity Flow*

Wiederholung: MinCut-MaxFlow Theorem

Der Sonderfall $k = 1$ (d.h. eine Quelle und eine Senke) ist in P.
Variationen des Algorithmus von Ford und Fulkerson lösen dieses Problem z.B. in $O(|V||E|^2)$.

Wiederholung: MinCut-MaxFlow Theorem

Der Sonderfall $k = 1$ (d.h. eine Quelle und eine Senke) ist in P.
Variationen des Algorithmus von Ford und Fulkerson lösen dieses Problem z.B. in $O(|V||E|^2)$.

Theorem

In einem gerichteten Graphen $G = (V, E)$ mit positiven Kantenkapazitäten ist der maximale Fluss zwischen zwei Knoten s und t gleich dem Wert des minimalen s-t Schnittes.

Das Problem des minimalen Mehrfachschnittes

Gegeben:

- ungerichteter Graph $G = (V, E)$
- Kantengewichte $c_e \in \mathbb{N}$
- Menge von Knotenpaaren $L = \{ (s_1, t_1), \ldots, (s_k, t_k) \}$

Definition (multicut)

Ein Mehrfachschnitt (multicut) ist eine Teilmenge $D \subseteq E$ der Kanten so, dass das Entfernen der Kanten in D jeweils s_i von t_i trennt.
Weiter nennen wir $c(D) \stackrel{def}{=} \sum_{e \in D} c_e$ die Kosten des Schnittes.

Das Problem des minimalen Mehrfachschnittes

Gegeben:

- ungerichteter Graph $G = (V, E)$
- Kantengewichte $c_e \in \mathbb{N}$
- Menge von Knotenpaaren $L = \{ (s_1, t_1), \ldots, (s_k, t_k) \}$

Definition (multicut)

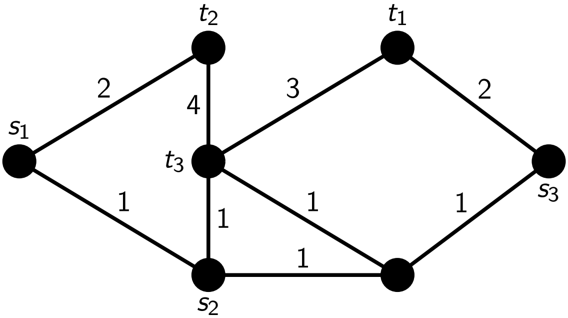
Ein Mehrfachschnitt (multicut) ist eine Teilmenge $D \subseteq E$ der Kanten so, dass das Entfernen der Kanten in D jeweils s_i von t_i trennt.
Weiter nennen wir $c(D) \stackrel{def}{=} \sum_{e \in D} c_e$ die Kosten des Schnittes.

Problem (MM)

Finde einen Mehrfachschnitt D mit minimalen Kosten $c(D)$.

Englisch: *Minimum Multicut*

Ein Beispiel



Bekannte Sonderfälle

Sonderfall $k = 1$

Das Problem des minimalen s-t-Schnittes.

Bekannte Sonderfälle

Sonderfall $k = 1$

Das Problem des minimalen s-t-Schnittes.

Sonderfall *multiway cut*

Gegeben: Eine Menge $T = \{ s_1, \dots, s_k \} \subseteq V$ von Terminalen.
Gesucht: Eine Partition des Graphen in k Teile so, dass keine zwei Terminale in der selben Teilmenge liegen und das Gewicht der Schnittkanten minimal ist.

Reduktion auf das Mehrfachschnittproblem durch Bilden aller Paare (s_i, s_j) für $i \neq j$.

Bekannte Sonderfälle

Sonderfall $k = 1$

Das Problem des minimalen s-t-Schnittes.

Sonderfall *multiway cut*

Gegeben: Eine Menge $T = \{ s_1, \ldots, s_k \} \subseteq V$ von Terminalen.
Gesucht: Eine Partition des Graphen in k Teile so, dass keine zwei Terminale in der selben Teilmenge liegen und das Gewicht der Schnittkanten minimal ist.

Reduktion auf das Mehrfachschnittproblem durch Bilden aller Paare (s_i, s_j) für $i \neq j$.

Folgerung

Für $k \geq 3$ ist das Mehrfachschnittproblem MM NP-schwer.

Einschränkung auf Bäume

Bemerkung

Wir werden die Probleme des Mehrfachschnittes (MM) und des Mehrgüterflusses (IMF) im Folgenden auf Bäume einschränken.

- *MM ist selbst für Bäume der Höhe* 1 *und ungewichteten Kanten NP-schwer.*
- *IMF ist für Bäume der Höhe* 3 *bereits NP-schwer.*

Einschränkung auf Bäume

Bemerkung

Wir werden die Probleme des Mehrfachschnittes (MM) und des Mehrgüterflusses (IMF) im Folgenden auf Bäume einschränken.

- *MM ist selbst für Bäume der Höhe* 1 *und ungewichteten Kanten NP-schwer.*
- *IMF ist für Bäume der Höhe* 3 *bereits NP-schwer.*

Erinnerung

Ein Baum ist ein zusammenhängender, zykelfreier Graph.

Eigenschaft: Zwischen zwei Knoten gibt es immer genau einen einfachen Weg.

Einige Definitionen

Definition

Sei G ein gewurzelter Baum und u, v zwei beliebige Knoten.

Die Tiefe von u ist die Länge des Pfades von u zur Wurzel.
Der höchste Knoten auf dem Weg von u nach v heißt der tiefste gemeinsame Vorfahre $lca(u, v)$.

LP-Form von MM für Bäume

Gegeben: Instanz von MM mit k Paaren (s_i, t_i).

Variablen: $d_e \in \{0,1\}$ für alle $e \in E$, sodass $e \in D \iff d_e = 1$.

Notation

Bezeichne P_i die Menge aller Kanten auf dem Weg von s_i nach t_i.

LP-Form von MM für Bäume

Gegeben: Instanz von MM mit k Paaren (s_i, t_i).
Variablen: $d_e \in \{0,1\}$ für alle $e \in E$, sodass $e \in D \iff d_e = 1$.

Notation

Bezeichne P_i die Menge aller Kanten auf dem Weg von s_i nach t_i.

Ganzzahlige LP-Form

$$
\begin{aligned}
\text{minimize} \quad & \sum_{e \in E} c_e d_e \\
\text{subject to} \quad & \forall\, i \in \{1, \ldots, k\} \sum_{e \in P_i} d_e \geq 1 \\
& \forall\, e \in E \; d_e \in \{0,1\}
\end{aligned}
$$

LP-Relaxierung von MM

LP-Relaxierung primales Problem

$$\begin{array}{rll} \text{minimize} & \displaystyle\sum_{e \in E} c_e d_e & \\ \text{subject to} & \forall\, i \in \{1, \ldots, k\} \displaystyle\sum_{e \in P_i} d_e & \geq 1 \\ & \forall\, e \in E \; d_e & \geq \; 0 \end{array}$$

zum relaxierten Schlupf

Das duale Problem

Duales Problem

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{k} f_i \\
\text{subject to} \quad & \forall\, e \in E \sum_{\substack{i=1 \\ e \in P_i}}^{k} f_i \ \leq c_e \\
& \forall\, i \in \{1, \ldots, k\}\ f_i \ \geq \ 0
\end{aligned}
$$

Das duale Problem

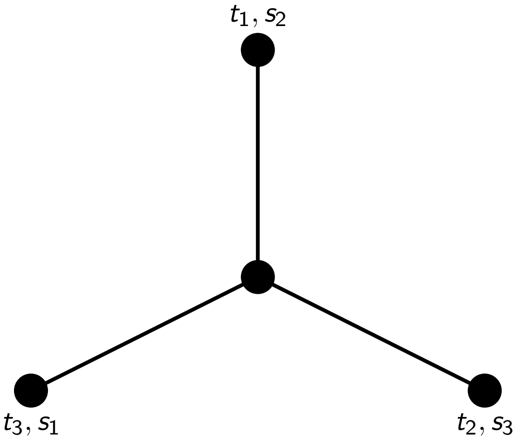
Duales Problem

$$
\begin{aligned}
\text{maximize} \quad & \sum_{i=1}^{k} f_i \\
\text{subject to} \quad & \forall\, e \in E \sum_{\substack{i=1\\ e \in P_i}}^{k} f_i \ \le c_e \\
& \forall\, i \in \{1, \ldots, k\}\ f_i \ \ge\ 0
\end{aligned}
$$

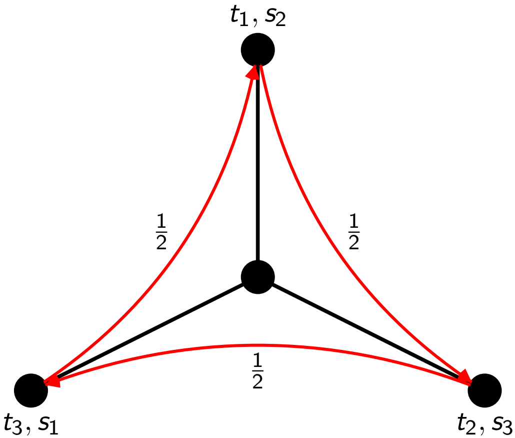
Bemerkung

Ganzzahlige Flusswerte sind nicht verlangt.
Fractional Multicommodity Flow (FMF)

Ein vielseitiges Beispiel



Ein vielseitiges Beispiel



Konsequenzen für die Lösungen

Lösungen im Beispiel

Minimaler Mehrfachschnitt (MM):	2
Maximaler ganzzahliger Fluss (IMF) :	1
Maximaler Fluss (FMF):	$\frac{3}{2}$

Ein bisschen Theorie

Betrachten wir ein lineares Programm:

primales Programm

$$\begin{array}{rl} \text{minimize} & c^T x \\ \text{subject to} & Ax \;\geq\; b \\ & \forall\, j \quad x_j \geq 0 \end{array}$$

Ein bisschen Theorie

Betrachten wir ein lineares Programm:

primales Programm

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \geq b \\ & \forall j \quad x_j \geq 0 \end{aligned}$$

duales Programm

$$\begin{aligned} \text{maximize} \quad & b^T y \\ \text{subject to} \quad & A^T y \leq c \\ & \forall i \quad y_i \geq 0 \end{aligned}$$

Der komplementäre Schlupf

Wir wissen, dass gilt...

Theorem

Seien $x = (x_1, \ldots, x_n)$ und $y = (y_1, \ldots, y_m)$ zulässige Lösungen für ein primales und sein duales lineares Programm. Beide Lösungen sind optimal genau dann, wenn gilt:

- *primal: Für alle $j \in \{1, \ldots, n\}$ gilt $x_j = 0$ oder $\sum_{i=1}^{m} a_{ij} y_i = c_j$*
- *dual: Für alle $i \in \{1, \ldots, m\}$ gilt $y_i = 0$ oder $\sum_{j=1}^{n} a_{ij} x_j = b_i$*

Englisch: Complementary slackness conditions

Das Primal-Dual-Schema

Seien zwei reelle Zahlen $\alpha, \beta \geq 1$ gegeben.

α-β relaxierte Schlupfbedingungen

- primal: Für alle $j \in \{1, \ldots, n\}$ gilt $x_j = 0$ oder $\frac{c_j}{\alpha} \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$
- dual: Für alle $i \in \{1, \ldots, m\}$ gilt $y_i = 0$ oder $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta b_i$

Das Primal-Dual-Schema

Seien zwei reelle Zahlen $\alpha, \beta \geq 1$ gegeben.

α-β relaxierte Schlupfbedingungen

- primal: Für alle $j \in \{1, \ldots, n\}$ gilt $x_j = 0$ oder $\frac{c_j}{\alpha} \leq \sum_{i=1}^{m} a_{ij} y_i \leq c_j$
- dual: Für alle $i \in \{1, \ldots, m\}$ gilt $y_i = 0$ oder $b_i \leq \sum_{j=1}^{n} a_{ij} x_j \leq \beta b_i$

Theorem

Seien x und y zulässige Lösungen des primalen bzw. dualen Programms welche α-β relaxierten Schlupfbedigungen genügen, so gilt

$$\sum_{j=1}^{n} c_j x_j \;\leq\; \alpha\beta \sum_{i=1}^{m} b_i y_i$$

Entwurf eines Algorithmus

Idee

Anwendung des Primal-Dual-Schemas mit $\alpha = 1$ und $\beta = 2$

zum LP

Entwurf eines Algorithmus

Idee

Anwendung des Primal-Dual-Schemas mit $\alpha = 1$ und $\beta = 2$

zum LP

Der relaxierte Schlupf

Primale Bedingung:

Für alle $e \in E$ gilt $d_e = 0$ oder $\sum_{\substack{i=1 \\ e \in P_i}}^{k} f_i \;=\; c_e$.

Duale Bedingung:

Für alle $i \in \{1, \ldots, k\}$ gilt $f_i = 0$ oder $1 \;\leq\; \sum_{e \in P_i} d_e \;\leq\; 2$.

zum Lemma zum Satz

Grundstruktur des Algorithmus

Ziel

Konstruktion eines Algorithmus der die relaxierten Schlupfbedingungen erfüllt und gleichzeitig Lösungen für MM und IMF findet.

Grundstruktur des Algorithmus

Ziel

Konstruktion eines Algorithmus der die relaxierten Schlupfbedingungen erfüllt und gleichzeitig Lösungen für MM und IMF findet.

Struktur des Algorithmus

- Initialisation
- Flow routing (primale Bedingung)
- Reverse delete (duale Bedingung)

Initialisation und Flow routing

Initialisation:

wähle eine beliebige Wurzel r;
for i = 1 to k do $f_i := 0$;
$D := \emptyset$;

Initialisation und Flow routing

Initialisation:

```
wähle eine beliebige Wurzel r;
for i = 1 to k  do    f_i := 0;
D := ∅;
```

Flow routing:

```
Ordne die Knoten in abnehmender Tiefe:  v_1, ..., v_n;
for i = 1 to n do
    for j = 1 to k do
        if  lca(s_j, t_j) = v_i  then
           -sende soviel Fluss f_j wie möglich
            von s_j nach t_j;
           -füge jede neu gesättigte Kante zu  D hinzu;
```

Reverse delete Phase

Reverse delete:

Seien $e_1, \ldots, e_p$ die Kanten in D
in der Reihenfolge in der sie zugefügt wurden;

for i = p downto 1 do
    wenn $D \setminus \{ e_i \}$ noch multicut ist dann
        lösche e_i aus D ;

Erfüllung des relaxierten dualen Schlupfes

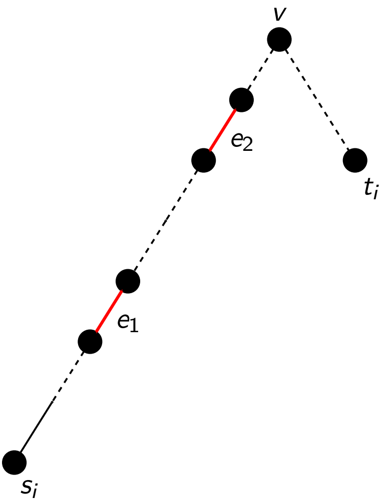
Lemma

Sei (s_i, t_i) ein source-sink Paar mit positivem Fluss ($f_i \neq 0$) und sei $lca(s_i, t_i) = v$. Dann enthält D höchstens je eine Kante in beiden Pfaden s_i nach v und v nach t_i.

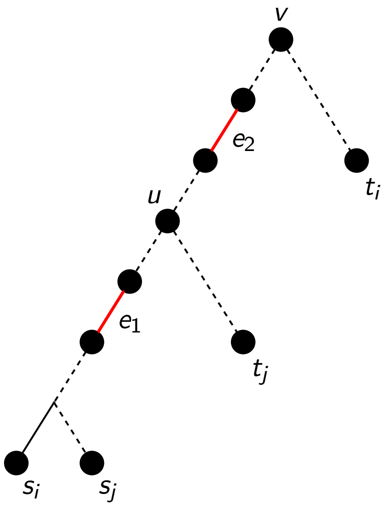
Beweis siehe Tafel und Bilder...

zum relaxierten Schlupf

Beweis des Lemmas



Beweis des Lemmas



Warum *Reverse Delete* benötigt wird

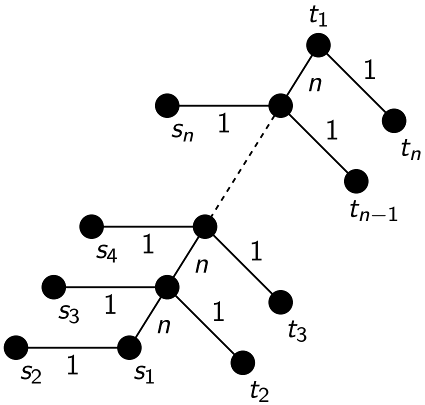
Frage:

Warum müssen wir rückwärts löschen?

Warum *Reverse Delete* benötigt wird

Frage:

Warum müssen wir rückwärts löschen?



Approximationsgüte

Theorem

Der angegebene Algorithmus erreicht für beide Probleme (MM und IMF auf Bäumen) die Approximationsgüte 2.

zum relaxierten Schlupf

Approximationsgüte

Theorem

Der angegebene Algorithmus erreicht für beide Probleme (MM und IMF auf Bäumen) die Approximationsgüte 2.

zum relaxierten Schlupf

Folgerung

Für jeden Baum mit ganzzahligen Kantengewichten und gegebener Menge von Knotenpaaren L gilt immer

$$OPT_{IMF} \;\leq\; \min_{multicut\ D} c(D) \;\leq\; 2\,OPT_{IMF}$$

Ein negatives Resultat

Frage

Können wir diese Technik für allgemeinere Klassen von Graphen anwenden?

Ein negatives Resultat

Frage

Können wir diese Technik für allgemeinere Klassen von Graphen anwenden?

Lemma

Die Mehrgüterfluss-Ganzzahligkeitslücke in Abhängigkeit von der Anzahl der Paare k auf allgemeinen Graphen ist mindestens $\frac{k}{2}$.

Beispiel an der Tafel...

Ein negatives Resultat

Frage

Können wir diese Technik für allgemeinere Klassen von Graphen anwenden?

Lemma

Die Mehrgüterfluss-Ganzzahligkeitslücke in Abhängigkeit von der Anzahl der Paare k auf allgemeinen Graphen ist mindestens $\frac{k}{2}$.

Beispiel an der Tafel...

Bemerkung

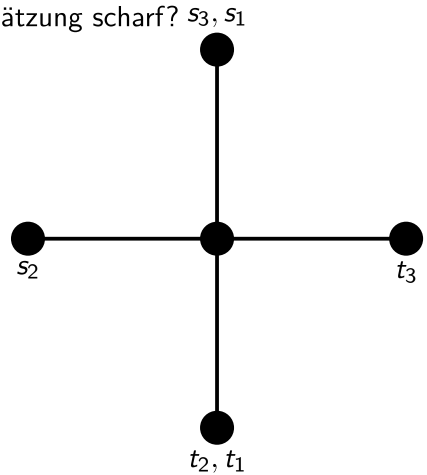
- *Für MM in allgemeinen Graphen gibt es Algorithmen der relativen Güte $O(\log k)$.*
- *Für IMF sind nur sehr komplexe Algorithmen bekannt.*

Zeugen für die Approximationsgüte

Ist unsere Abschätzung scharf?

Zeugen für die Approximationsgüte

Ist unsere Abschätzung scharf?



Zusammenfassung

- Probleme IMF und MM auf Bäumen
- Dualität der Probleme
- Primal-Dual-Schema
- 2-Approximationsalgorithmus und Analyse
- Schwierigkeit für allgemeine Graphen