S1G1 Diskrete Mathematik: Kostenminimale Flüsse

Andreas Gwilt

Recap optimality criterion for Maximum flows??

1 Problemstellung

Draw picture with bakeries, bread lorries and graph, cost:road length ⇒ Graph G mit

- Kapazitäten $u : E(G) \to \mathbb{R}_+$ "upper bound"
- Kosten $c : E(G) \to \mathbb{R}$ "cost"
- Balance $b : V(G) \to \mathbb{R}$: Was wollen von diesem Knoten? Quelle/Senke

Das bringt uns zur ersten Definition:

Def (b-Fluss)**.** Gegeben sei ein Digraph G, Kapazitäten $u : E(G) \to \mathbb{R}_+$ und Zahlen $b : V(G) \to \mathbb{R}$ mit $\sum_{v \in V(G)} b(v) = 0$. Ein **b-Fluss** in (G, u) ist eine Funktion $f : E(G) \to \mathbb{R}_+$ mit $f(e) \leq u(e)$ für alle $e \in E(G)$ und

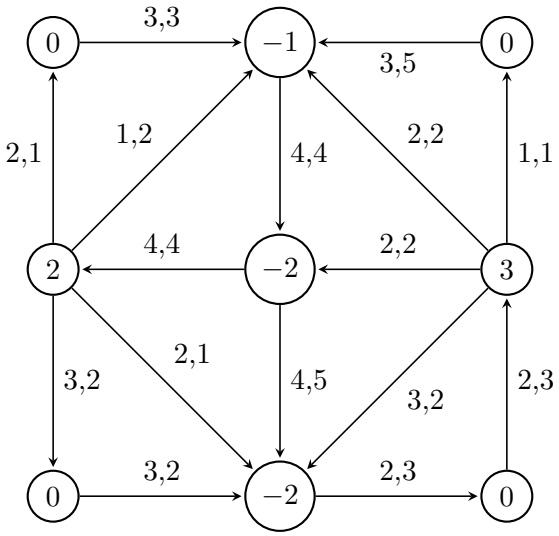
$$\sum_{e \in \delta^+(v)} f(e) - \sum_{e \in \delta_-(v)} f(e) = b(v)$$

für alle $v \in V(G)$.

Bem.

- Ein b-Fluss mit $b \equiv 0$ heißt Zirkulation.
- $b(v)$ heißt **Balance** des Knotens v.
- Falls $b(v) > 0$: v heißt **Quelle**, $b(v)$ heißt **Angebot**.
- Falls $b(v) < 0$: v heißt **Senke**, $|b(v)|$ heißt **Nachfrage**.
- Ein b-Fluss kann leicht mit einem Max-Flow Algorithmus in z.B. $O(m^2 n)$ gefunden werden (oder es kann entschieden werden, dass es keinen Gibt).

Aufgabe 1: Finde einen (Doesn't have to be minimal!) b-Fluss in dem beigelegten Graphen (Legende: $u(e), c(e)$):



Wir wollen aber nicht nur irgendeinen b-Fluss, dass können wir schon. Wir wollen einen minimalen. Das bringt uns zu folgendem Berechnungsproblem:

Minimum-Cost-Flow-Problem

Instanz:	Ein Digraph G, Kapazitäten $u : E(G) \to \mathbb{R}_+$, Zahlen $b : V(G) \to \mathbb{R}$ mit $\sum_{v \in V(G)} b(v) = 0$ und Gewichte $c : E(G) \to \mathbb{R}$.
Aufgabe:	Bestimme eine b-Fluss f mit minimalen Kosten $c(f) := \sum_{e \in E(G)} f(e)c(e)$ (oder entscheide, dass es keinen solchen gibt).

Show example with two flows f_1, f_2 from book.

Aufgabe 2: Was sind die Kosten von f_1 und f_2? Wo unterscheiden sie sich?

$\Rightarrow$ f_2 teurer als f_1. Wie kommen wir von f_1 zu f_2 ? $\rightsquigarrow$ Kreis, an dem wir augmentieren. Ähnlich zu Max-Flow ....

2 Ein Optimalitätskriterium

Zuerst ein Paar Definitionen und Notationen:

Def. Sei G ein Digraph. Dann ist $\overleftrightarrow{G} := (V(G), E(G) \cup \{\overleftarrow{e} | e \in E(G)\})$. (Also G, wo aber jede Kante zusätzlich auch in die andere Richtung geht.)

Def (Residualgraph)**.** Sei G ein Digraph mit Kapazitäten $u : E(G) \to \mathbb{R}_+$, Kosten $c : E(G) \to \mathbb{R}$ und ein b-Fluss f. Dann definieren wir die **Residualkapazitäten** $u_f : E(\overleftrightarrow{G}) \to \mathbb{R}_+$ durch $u_f(e) := u(e) - f(e)$ und $u_f(\overleftarrow{e}) = f(e)$ für alle $e \in E(G)$ und die **Residualkosten** $\overleftrightarrow{c} : E(\overleftrightarrow{G}) \to \mathbb{R}$ durch $\overleftrightarrow{c}(e) := c(e)$ und $\overleftrightarrow{c}(\overleftarrow{e}) := -c(e)$ für alle $e \in E(G)$. Ab jetzt: Einfach $c(e)$ statt $\overleftrightarrow{c}(e)$.

Der **Residualgraph** ist der Graph G_f definiert durch $V(G_f) := V(G)$ und $E(G_f) := \{e \in E(\overleftrightarrow{G}) | u_f(e) > 0\} = \{e \in E(G) | f(e) < u(e)\} \dot\cup \{\overleftarrow{e} | e \in E(G), f(e) > 0\}$.

Ein **f-augmentierender Kreis** ist ein Kreis in G_f.

Aufgabe 3: Macht euch klar, dass Augmentierung entlang eines f-augmentierenden Kreises C um $\gamma \leq \min\{u_f(e) | e \in C\}$ wieder einen b-Fluss f' ergibt mit

$$f'(e) = \begin{cases} f(e) + \gamma & : e \in C \\ f(e) - \gamma & : \overleftarrow{e} \in C \\ f(e) & : \text{sonst} \end{cases}$$

Es geht aber auch (fast) anders herum:

Lemma 2.1. *Sei G ein Digraph mit Kapazitäten $u : E(G) \to \mathbb{R}_+$. Seien f und f' b-Flüsse in (G, u). Dann ist die Funktion $g : E(\overleftrightarrow{G}) \to \mathbb{R}_+$, gegeben durch $g(e) := \max\{0, f'(e) - f(e)\}$ und $g(\overleftarrow{e}) := \max\{0, f(e) - f'(e)\}$ für $e \in E(G)$, eine Zirkulation in $\overleftrightarrow{G}$. Ferner ist $g(e) = 0$ für alle $e \notin E(G_f)$ und $c(g) = c(f') - c(f)$.*

Beweis. 1. In jedem Knoten $v \in V(\overleftrightarrow{G})$ haben wir

$$\sum_{e \in \delta^+_{\overleftrightarrow{G}}(v)} g(e) - \sum_{e \in \delta^-_{\overleftrightarrow{G}}(v)} g(e) = \sum_{e \in \delta^+_G(v)} (f'(e) - f(e)) - \sum_{e \in \delta^-_G(v)} (f'(e) - f(e))$$
$$= b(v) - b(v) = 0$$

also ist g eine Zirkulation in $\overleftrightarrow{G}$.

2. Nun betrachten wir für jedes $e \in E(\overleftrightarrow{G}) \setminus E(G_f)$ zwei Fälle:
   - $e \in E(G) \Rightarrow f(e) = u(e) \Rightarrow f'(e) \leq f(e) \Rightarrow g(e) = 0$
   - $e = \overleftarrow{e_0}$ für ein $e_0 \in E(G) \Rightarrow f(e_0) = 0 \Rightarrow g(\overleftarrow{e_0}) = 0$.
3. Die letzte Aussage lässt sich leicht beweisen:

$$c(g) = \sum_{e \in E(\overleftrightarrow{G})} c(e)g(e) = \sum_{e \in E(G)} c(e)f'(e) - \sum_{e \in E(G)} c(e)f(e) = c(f') - c(f)$$

□

Diese Zirkulationen lassen sich aber nun in Kreise aufteilen:

Lemma 2.2 (Ford und Fulkerson, 1962)**.** *Für jede Zirkulation f in einem Digraphen G gibt es eine Familie $\mathcal{C}$ von höchstens $|E(G)|$ Kreisen in G und positive Zahlen $h(C)$ $(C \in \mathcal{C})$ mit $f(e) = \sum_{C \in \mathcal{C}} \sum_{e \in E(C)} h(e)$ für alle $e \in E(G)$.*

Beweis. Folgt direkt aus AlMa I, Lemma 6.10. □

Damit können wir ein Optimalitätskriterium ähnlich zu dem von s-t-Flüssen beweisen:

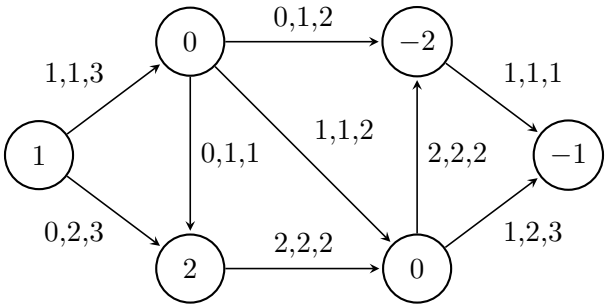
Satz 2.3 (Klein, 1967)**.** *Sei (G, u, b, c) eine Instanz der* Minimum-Cost-Flow-Problems*. Ein b-Fluss f hat genau dann minimale Kosten, wenn es keinen f-augmentierenden Kreis mit negativem Gesamtgewicht gibt (i.e. G_f ist konservativ).*

Beweis. "⇒" Gibt es einen f-augmentierenden Kreis C mit negativem Gewicht, so können wir f entlang C um ein $\varepsilon > 0$ augmentieren und einen b-Fluss f' mit gesenkten Kosten erhalten. Somit ist f kein Fluss mit minimalen Kosten.

"⇐" Ist f kein b-Fluss mit minimalen Kosten, dann gibt es einen anderen b-Fluss f' mit geringeren Kosten. Betrachte das in Lemma 2.1 definierte g. Dann ist g eine Zirkulation mit $c(g) < 0$. Nach Lemma 2.2 kann man g dann in Flüsse entlang einzelnen Kreisen zerlegen. Da $g(e) = 0$ für alle $e \notin E(G_f)$, sind all diese Kreise f-augmentierend. Mindestens einer von ihnen muss jedoch negatives Gesamtgewicht haben, womit der Satz bewiesen ist.

□

Aufgabe 4: Ist der folgende b-Fluss minimal? Wenn nicht, welcher Kreis in G_f hat negative Kosten? (Legende: $f(e), u(e), c(e)$)



3 Zwei Algorithmen

3.1 Der Minimum-Mean-Cycle-Cancelling-Algorithmus

Ähnlich wie Ford-Fulkerson für maximale Flüsse, legt Satz 2.3 einen Algorithmus nahe: Finde zuerst einen beliebigen b-Fluss (mit einem Max-Flow-Algorithmus), und augmentiere wiederholt entlang augmentierenden Kreisen negativen Gewichts.

Wie bei Ford-Fulkerson gibt es aber Probleme mit der Laufzeit/Terminierung, wenn wir nicht die richtigen Kreise auswählen. Dazu wählen wir Kreise mit möglichst niedrigen gewichten, genauer gesagt: mit minimalem durchschnittlichem Kantengewicht. Wenn ein solcher Kreis nicht mehr negatives Gesamtgewicht hat, dann sind wir fertig.

Minimum-Mean-Cycle-Cancelling-Algorithmus (Klein, 1967)

(1) Bestimme einen b-Fluss f (z.B. mit Edmonds-Karp).

(2) Bestimme einen Kreis C in G_f mit minimalem durchschnittlichem Kantengewicht.
If C hat nichtnegatives Gesamtgewicht (oder G_f ist azyklisch) **then stop**.

(3) Berechne $\gamma := \min_{e \in E(C)} u_f(e)$. Augmentiere f entlang C um γ. **Go to** (2).

Die Laufzeit braucht leider etwas lange zu zeigen, aber der Algorithmus geht (für $c_{\min}$ das Minimum aller Kantenkosten) in $O(m^2 n^2 \log(n|c_{\min}|))$ oder in $O(m^3 n^2 \log n)$ Zeit. (Steht in [1], Lemma 9.8, Korollar 9.9, Satz 9.10.)

Wir gehen jetzt nicht näher darauf ein (Korrektheit ist klar), sondern gehen sofort zum nächsten Algorithmus. Hier ist die Idee, schon von Anfang an einen minimalen Fluss zu haben — allerdings kein b-Fluss. Dann bauen wir f nach und nach zu einem b-Fluss auf, wobei wir Optimalität beibehalten. Intuitiv würden wir hier kürzeste Wege von Quellen zu Senken finden, und genau das werden wir machen. Zuerst brauch wir aber den folgenden Satz:

Satz 3.1. *Sei (G, u, b, c) eine Instanz des* Minimum-Cost-Flow-Problems *und f ein b-Fluss mit minimalen Kosten. Sei P ein kürzester (bzgl. $\overleftrightarrow{c}$) s-t-Weg in G_f (für irgendwelche $s, t \in V(G)$). Sei f' der durch Augmentierung von f entlang P um den Wert $\gamma \leq \min_{e \in P} u_f(e)$ entstehender Fluss. Dann ist f' ein kostenminimaler Fluss zu den Balancen b' mit*

$$b'(v) = \begin{cases} b(v) + \gamma & : v = s \\ b(v) - \gamma & : v = t \\ b(v) & : \text{sonst.} \end{cases}$$

Beweis. Nach Konstruktion ist f' ein b'-Fluss. Angenommen, f' sei nicht kostenminimal. Nach Satz 2.3 existiert dann ein Kreis C in G'_f mit negativem Gesamtgewicht.

Idea: Hilfsgraph aus C und P, der Subgraph von G_f ist, aber einen günstigeren s-t-Weg als P hat, im Widerspruch zur Wahl von P.

Betrachte den Graphen H, der aus $(V(G), E(C) \dot\cup E(P))$ durch das Entfernen von Paaren entgegengesetzt orientierter Kanten hervorgeht. (Hier werden Kanten, die sowohl in C als auch in P vorkommen, zweimal gezählt.)

Es gilt $E(H) \subseteq E(G_f)$, da jede Kante aus P schon in G_f ist und jedes $e \in G_{f'} \setminus G_f$ durch Augmentierung entlang P in $G_{f'}$ gelandet ist, also $\overleftarrow{e} \in P \Rightarrow e$ wurde zusammen mit $\overleftarrow{e}$ herausgelöscht.

Es gilt $c(E(H)) = \overbrace{c(E(C))}^{<0} + c(E(P)) < c(E(P))$ (Zur Erinnerung: $c(\overleftarrow{e}) = -c(e)$). Ferner ist H die Vereinigung eines s-t-Weges und einigen Kreisen. (Übungsaufgabe!) Da aber $E(H) \subseteq E(G_f)$, kann keiner der Kreise negativ sein (sonst wäre f nicht minimal).

Damit enthält H einen Weg P' mit $c(E(P)) \overset{s.o.}{>} c(E(H)) = c(E(P')) + \overbrace{c(E(H \setminus P))}^{>0} \Rightarrow P'$ hat kleineres Gewicht als P ↯ zur Wahl von P. □

Sind die Gewichte konservativ (keine Kreise in G_f mit negativen Kosten), können wir mit $f \equiv 0$ als optimale Zirkulation beginnen. Sonst saturieren wir alle Kanten mit negativen Kosten und setzen b entsprechend.

Sukzessive-Kürzeste-Wege-Algorithmus

(1) Setze $b' \leftarrow b$ und $f(e) \leftarrow 0$ für alle $e \in E(G)$.

(2) **If** $b' = 0$ **then stop, else:**

Wähle einen Knoten s mit $b'(s) > 0$.

Wähle einen Knoten t mit $b'(t) < 0$, so dass t von s aus in G_f erreichbar ist.

If es gibt kein solches t **then stop**. (Es gibt keinen b-Fluss)

(3) Bestimme einen s-t-Weg P (z.B. mit Moore-Bellman-Ford in $O(nm)$) in G_f mit minimalen Kosten bzgl. c_f.

(4) Berechne $\gamma \leftarrow \min\{\min_{e \in E(P)} u_f(e), b'(s), -b'(t)\}$.
Setze $b'(s) \leftarrow b'(s) - \gamma$ und $b'(t) \leftarrow b'(t) + \gamma$. Augmentiere f entlang P um γ.
Go to (2)

Aufgabe 5: Findet mithilfe des Sukzessive-Kürzeste-Wege-Algorithmus einen minimalen Fluss in dem Graphen an der Tafel.

Wir gehen jetzt von ganzzahligen u und b aus, da sonst ähnliche Probleme wie bei Ford-Fulkerson entstehen. Dann terminiert der Algorithmus nach höchstens $B := \frac{1}{2} \sum_{v \in V(G)} |b(v)|$ Iterationen, da in jeder Iteration b' mindestens "um 1 kleiner wird". Nach Satz 3.1 ist der resultierende Fluss optimal, wenn der Anfgangsfluss optimal war.

Mit Moore-Bellman-Ford für die kürzesten Wege kommen wir zu einer Laufzeit von $O(Bnm)$. Note: Graph always stays conservative, since f is always minimal $\rightsquigarrow$ no negative cycles in G_f.

Bem. Mit etwas tricksen können wir unsere Wege in einem Graphen ohne negativen Gewichten berechnen, und bekommen die Laufzeit $O(nm + B(m + n \log n))$.

Damit ist dieser Algorithmus für kleine b und große Graphen deutchlich besser, aber dafür nicht streng polynomiell. Für $B = O(n)$ ist die "trickreiche" Version der schnellste bekannte Algorithmus.

Literatur

[1] Bernhard Korte, Jens Vygen: *Kombinatorische Optimierung (2. Auflage)* – Springer, 2012

[2] Christina Büsing: *Grapen- und Netzwerkoptimierung* – Spektrum, 2010