Einführung in die lineare und kombinatorische Optimierung
Serie 3

Maurice Althoff (FU 4745454)
Michael R. Jung (HU 502133)
Felix Völker (TU 331834)

6. November 2014

Aufgabe 9

Berechnen wir zunächst die Kodierungslänge der größten vorkommenden Zahl im Algorithmus. Für $|n| \geq 1$ ist das offensichtlich das Ergebnis. Was berechnet also der Algorithmus? Für ein festes k ist dies n^{3^k}, und da $k = \langle n \rangle$, ist dies $c := n^{3^{\lceil \log_2(|n|+1) \rceil + 1}}$. Schätzen wir zunächst einmal ab: $\lceil \log_2(|n|+1) \rceil + 1 \leq 2 + \log_2 |n|$, außerdem gilt, dass $\log_2 x = \frac{\log_3 x}{\log_3 2}$ und $\log_3 2 > \frac{1}{2}$. Somit ist $3^{\lceil \log_2(|n|+1) \rceil + 1} \leq 3^{2 + \frac{\log_3 |n|}{\log_3 2}} < 9 \cdot 3^{2 \log_3 |n|} = 9|n|^2$.
Folglich ist $\langle c \rangle \leq \lceil \log_2(|n^{(3n^2)}| + 1) \rceil + 1 \leq \log_2 |n^{(3n^2)}| + 2 = 3n^2 \log_2 |n| + 2 \in \mathcal{O}(n^2 \log |n|)$.
Zählen wir nun die elementaren Operationen. In jedem Schleifendurchlauf haben wir 2 Multiplikationen sowie eine Addition (Erhöhung des Zählers), also müssen wir für die Laufzeit des Algorithmus $\langle c \rangle$ mit $\langle n \rangle \cdot 3$ multiplizieren. Es ergibt sich:

$$\begin{aligned}
f_A(n) &\leq (3n^2 \log_2 |n| + 2) \cdot 3(\lceil \log_2(|n|+1) \rceil + 1) \\
&\leq (3n^2 \log_2 |n| + 2) \cdot (3 \log_2(|n|) + 6) \\
&\leq 9n^2 (\log_2 |n|)^2 + 6 \log_2 |n| + 18n^2 \log_2 |n| + 12 \\
&\in \mathcal{O}(n^2 (\log |n|)^2)
\end{aligned}$$

Aufgabe 10

Sei $\mathcal{Z} = \{Z_i | 1 \leq i \leq m\}$ eine Familie von Klauseln über Variablen aus $X = \{x_1, \ldots, x_n\}$.
Folgender Algorithmus `Alg` findet eine Belegung `f` von X, die mindestens die Hälfte der Klauseln erfüllt:

```
Alg:
FOR i=1 TO n:
  BEGIN
    A:={Z ∈ 𝒵 |x_i ∈ Z}
    B:={Z ∈ 𝒵 |¬x_i ∈ Z}
    IF |A|<|B| THEN f(x_i)=falsch ELSE f(x_i)=wahr
    𝒵=𝒵\(A∪B)
  END
```

- Analysieren wir zunächst die Laufzeit:
  Eingabegröße ist ungefähr $\sum_{i=1}^{m} \langle Z_i \rangle =: g$. In jedem Schritt müssen wir zur Berechnung von `A` und `B` maximal alle Klauseln, deren Kodierung insgesamt in der Größe durch g beschränkt ist, durchsuchen. Die entsprechenden Indizes in `A` und `B` zu speichern (Speicherbedarf als Bitmaske: m), reicht völlig aus, des Weiteren können wir bereits beim Scan die entsprechenden Klauseln löschen. Also ist die Laufzeit jedes Schleifendurchlaufs durch $g + 2m$ beschränkt. Da wir insgesamt n Durchläufe haben, ist die ganze Laufzeit durch $n(g + 2m)$ beschränkt.
  Es gilt: $n \leq g,\ m \leq g$.
  Folglich haben wir eine Laufzeitschranke von $g(g + 2g) \in \mathcal{O}(g^2)$.

- Nun zur Korrektheit:
  Am Ende ist $\mathcal{Z} = \varnothing$, denn jede Klausel wird entfernt, wenn die Variable mit dem kleinsten vorkommenden Index betrachtet wird. Jedes mal wenn Klauseln entfernt werden, wird durch die definierte Belegung mindestens die Hälfte dieser erfüllt. Da am Ende aber alle Klauseln entfernt werden, wird also mindestens die Hälfte aller Klauseln erfüllt. ✓

Aufgabe 11

Sei $D = (V, A)$ ein Digraph mit $|V| = n$ und Adjazenzmatrix $A \in \mathbb{R}^{n \times n}$. Im folgenden gehen wir o.B.d.A. davon aus, dass $V = \{1, 2, \ldots, n\}$ gilt. Außerdem sei $A^k =: \left(a_{i,j}^{(k)}\right)_{1 \leq i,j \leq n}$.
Zeigen zunächst:

> In $a_{i,j}^{(k)}$steht die Anzahl aller verschiedenen gerichteten Ketten der Länge genau k zwischen den Knoten i und j.

Im Folgenden meinen wir mit Ketten/Wegen/Kreisen immer gerichtete Ketten/Wege/Kreise.

Beweis. Beweis per Induktion über k.
Induktionsanfang
Für $k = 1$ entspricht dies genau der Definition der Adjazenzmatrix.
Induktionsvoraussetzung
$a_{i,j}^{(k)}$=Anzahl der verschiedenen Ketten der Länge genau k zwischen den Knoten i und j.
Induktionsbehauptung
n $a_{i,j}^{(k+1)}$ =Anzahl der verschiedenen Ketten der Länge $k + 1$ zwischen den Knoten i und j.
Induktionsschritt
Betrachte $a_{i,j}^{(k+1)}$.
Nach Definition der Matrixmultiplikation ist $A^{k+1} = A^k A$ und somit $a_{i,j}^{(k+1)} = \sum_{p=1}^{n} a_{i,p}^{(k)} \cdot a_1^{(k)} p, j$.

Nach Induktionsvoraussetzung ist $a_{i,p}^{(k)}$ die Anzahl der verschiedenen Ketten der Länge k von i nach p. Dies wird nun mit der Anzahl der Kanten zwischen p und j, nämlich $a_{p,j}^{(1)}$, multipliziert, und wir erhalten die Anzahl der verschiedenen Ketten zwischen i und j der Länge $k + 1$ mit vorletztem Knoten p. Wenn wir dies nun über alle möglichen vorletzten Knoten summieren, erhalten wir genau das gewünschte. □

Definiere $R := \sum_{p=1}^{n} A^p$. In $R_{i,j}$ steht also folglich die Anzahl der verschiedenen Ketten von i nach j der Länge $\leq n$. Ist j von i aus erreichbar, so muss dieser Eintrag verschieden 0 sein, da

- ein Weg auch eine Kette ist,
- wenn eine i-j-Kette existiert, auch ein i-j-Weg existiert und
- ein i-j-Weg höchstens Länge n (für $i \neq j$ sogar nur Länge $n - 1$) haben kann.

Zeige nun: D stark zusammenhängend $\Leftrightarrow R_{i,j} \neq 0 \forall 1 \leq i, j \leq n$

Beweis.

"⇒" Seien $i, j \in V, i \neq j$. Dann existiert ein i-j-Weg in in D. Also ist nach obigem Beweis $R_{i,j} \neq 0$. Für $i = j$ gilt, das jeder Knoten auf einem Kreis liegen muss, da per definitionem auch i von i aus erreichbar sein muss. Also sind auch die Diagonaleinträge nicht 0. ✓

"⇐" Betrachte zwei Knoten i, j. Für $i = j$ gilt, dass i auf einem Kreis liegt, da $R_{i,i} \neq 0$. Also ist i von i aus erreichbar. Für $i \neq j$ gilt, dass ein i-j-Weg und ein j-i-Weg existieren, weil $R_{i,j}$ respektive $R_{j,i}$ verschieden 0 sind. Also ist D stark zusammenhängend.

□

Aufgabe 12

Dass die genannten Probleme in $\mathcal{NP}$ sind, sieht man leicht:

a) Das Zertifikat ist eine Permutation π der Knoten*. Hier muss man nun prüfen, ob $(\pi(1), \ldots, \pi(|V|), \pi(1))$ ein Hamiltonkreis ist und, falls ja, ob die Summe der Kantengewichte auf diesem Kreis K nicht übersteigt.

b) Das Zertifikat ist eine Permutation π der Knoten*. Hier muss man nun prüfen, ob $[\pi(1), \ldots, \pi(|V|)]$ ein Hamiltonweg und $\pi(1) = u \wedge \pi(|V|) = v$ ist.

c) Das Zertifikat ist eine Permutation π der Knoten*. Hier muss man nun prüfen, ob $[\pi(1), \ldots, \pi(|V|)]$ ein Hamiltonweg ist.

d) Im gerichteten Fall funktionieren die gleichen Zertifikate, hier muss man stattdessen überprüfen ob es die gerichteten Äquivalente sind.

Reduktionen:

a) Gegeben sei ein Graph $G = (V, E)$. Wir definieren uns nun noch eine Kantenbewertung $c : E \to \mathbb{N}$ mit $c(e) = 1 \; \forall \, e \in E$.
Nun gibt es genau dann einen Hamiltonkreis in G, wenn es eine Tour in G der Länge $\leq |V|$ gibt.

Beweis. In G existiert ein Hamiltonkreis.
⇔ Es existiert eine Permutation π der Knoten mit $e_i := \{\pi(i), \pi(i+1)\} \in E \; \forall 1 \leq i < |V|$ und $e_{|V|} := \{\pi(|V|), \pi(1)\} \in E$.
⇔ Es existiert eine Permutation π der Knoten mit $e_i := \{\pi(i), \pi(i+1)\} \in E \; \forall 1 \leq i < |V|$ und $e_{|V|} := \{\pi(|V|), \pi(1)\} \in E$ und es gilt $\sum\limits_{i=1}^{|V|} c(e_i) = n$.
⇔ Es existiert eine Tour in G mit Gewicht $= |V|$.
⇔ Es existiert eine Tour in G mit Gewicht $\leq |V|$.
Die letzte Äquivalenz folgt, da per Konstruktion jede Tour (die ja alle genau $|V|$ Kanten haben) genau Gewicht $|V|$ hat. □

b) Gegeben sei ein Graph $G = (V, E)$. Sei $u \in V$ beliebig.
Konstruiere $G' = (V' := V \cup \{u'\}, E' := E \cup \{\{v, u'\} | \{v, u\} \in E\})$.
Nun gibt es genau dann einen Hamiltonkreis in G, wenn es einen $[u, u']$-Hamiltonweg in G' gibt.

*Für einen Graphen $G = (V, E)$ oder einen Digraphen $D = (V, A)$ ist eine Permutation der Knoten eine bijektive Abbildung $\pi : \{1, 2, \ldots, |V|\} \to V$.

Beweis. In G existiert ein Hamiltonkreis.
$\Leftrightarrow$ Es ex. eine Permutation π von V' mit $\pi(1) = u,\ \pi(|V'|) = u'$,
$\{\pi(i), \pi(i+1)\} \in E\ \forall 1 \leq i < |V|$ und $\{\pi(|V|), \pi(1) = u\} \in E$.
(Dies gilt, da der "Startknoten" beliebig gewählt werden kann.)
$\Leftrightarrow$ Es ex. eine Permutation π von V' mit $\pi(1) = u,\ \pi(|V'|) = u'$ und
$\{\pi(i), \pi(i+1)\} \in E'\ \forall 1 \leq i < |V'|$. (Für $i = |V'| - 1 = |V| : \{\pi(|V|), u'\} \in E'$.)
(Dies gilt, da u und u' die gleiche Nachbarschaft besitzen.)
$\Leftrightarrow$ Es existiert ein $[u, u']$-Hamiltonweg in G'. □

c) Da wir nun bereits wissen, dass das $[u, v]$-Hamiltonweg-Problem $\mathcal{NP}$ -vollständig ist werden wir dieses auf das allgemeine Hamiltonweg-Problem reduzieren.
Sei $G = (V, E)$ ein Graph und seien $u, v \in V$ zwei ausgezeichnete Knoten. G enthält genau dann einen $[u, v]$-Hamiltonweg, wenn $G' = (V' := V \cup \{u', v'\}, E' := E \cup \{\{u', u\}, \{v, v'\}\})$ einen Hamiltonweg besitzt.

Beweis.

"$\Rightarrow$" Sei $|V| = n$ und sei $[u = v_1, v_2, \ldots, v_n = v]$ ein $[u, v]$-Hamiltonweg in G, dann ist $[u', v_1, v_2, \ldots, v_n, v']$ ein Hamiltonweg in G'.

"$\Leftarrow$" Sei $[x_0, x_1, \ldots, x_{n+1}]$ ein Hamiltonweg in G'. Da $\deg(u') = 1$ muss gelten, dass $u' \in \{x_0, x_{n+1}\}$, denn alle anderen Knoten müssen ja mindestens Grad 2 haben. Da auch $\deg(v') = 1$, gilt: $v' \in \{x_0, x_{n+1}\} \setminus \{u'\}$. Also existiert ein $[u', v']$-Hamiltonweg oder ein $[v', u']$-Hamiltonweg, im ungerichteten Fall also beide. Sei nun o.B.d.A. $x_0 = u'$ und $x_{n+1} = v'$. Da u' nur einen Nachbarn, nämlich u, und v' nur einen Nachbarn, nämlich v hat, muss gelten: $x_1 = u \wedge x_n = v$. Also ist $[x_1, x_2, \ldots, x_n]$ ein $[u, v]$-Hamiltonweg in G.

□

d) Die Konstruktioen und Beweise funktionieren genauso, dabei haben wir darauf geachtet, dass man bei hinzugefügten Bögen die geschweiften Klammern nur durch runde zu ersetzten braucht, d.h. die richtige Orientierung ist bereits beachtet. In a) verändert sich also tatsächlich nichts, außer dass c nun eine Bogenbewertung ist. In b) bekommt u' nur die eingehenden Kanten von u und in c) ist durch die Orientierung bereits sichergestellt, dass ein gerichteter Hamiltonweg bereits ein $[u', v']$-Hamiltonweg ist.