Vorlesung
Kombinatorische Optimierung
(Wintersemester 2016/17)
Kapitel 1: Kürzeste Wege und Kreise

Volker Kaibel

Otto-von-Guericke Universität Magdeburg

(Version vom 20. Oktober 2016)

Graphen und Digraphen

Definition 1.1

Ein **Graph** ist ein Paar $G = (V, E)$ bestehend aus einer endlichen **Knotenmenge** V und einer Teilmenge $E \subseteq \binom{V}{2}$ der zweielementigen Teilmengen von V, der **Kantenmenge** von G.

Definition 1.2

Ein **Digraph** (**gerichteter Graph**) ist ein Paar $D = (V, A)$ bestehend aus einer endlichen **Knotenmenge** V und einer Teilmenge

$$A \subseteq V \times V \setminus \{(v, v) \mid v \in V\},$$

der **Bogenmenge** von D. Zwei Bögen $(v, w), (w, v) \in A$ heißen **antiparallel**.

Multigraphen und -digraphen

Bemerkung

- *Verallgemeinerung von Digraphen: Multi-Digraphen* $D = (V, A, \Psi)$ *mit* $\Psi : A \longrightarrow V \times V$ *und beliebiger endlicher Indexmenge* A *der Bögen. (Ähnlich: Multi-Graphen)*
- *Man kann Konzepte, Resultate, Algorithmen in der Regel in offensichtlicher Weise von (Di-)Graphen auf Multi-(Di-)Graphen übertragen.*

Wege und Kreise in Graphen

Definition 1.3

Sei $G = (V, E)$ ein Graph mit $V = \{v_0, v_1, \ldots, v_l\}$ und

$$E = \{\{v_0, v_1\}, \{v_1, v_2\}, \ldots, \{v_{l-1}, v_l\}\}$$

1. Sind $v_0, v_1, \ldots, v_l$ paarweise verschieden, so heißt G ein $v_0 - v_l$-**Weg** der **(kombinatorischen) Länge** l.
2. Ist $v_0 = v_l$ und sind $v_0 = v_l, v_1, \ldots, v_{l-1}$ paarweise verschieden, so ist G ein **Kreis** der **(kombinatorischen) Länge** l.

Wege und Kreise in Digraphen

Definition 1.4

Sei $D = (V, A)$ ein Digraph mit $V = \{v_0, v_1, \dots, v_l\}$ und

$$A = \{(v_0, v_1), (v_1, v_2), \dots, (v_{l-1}, v_l)\}$$

1. Sind $v_0, v_1, \dots, v_l$ paarweise verschieden, so heißt D ein **(gerichteter)** $v_0 - v_l$-**Weg** der **(kombinatorischen) Länge** l.
2. Ist $v_0 = v_l$ und sind $v_0 = v_l, v_1, \dots, v_{l-1}$ mit $l \geq 2$ paarweise verschieden, so ist G ein **(gerichteter) Kreis** der **(kombinatorischen) Länge** l.

Teilgraphen

Definition 1.5

Sind $G = (V, E)$ und $G' = (V', E')$ (bzw. $D = (V, A)$ und $D' = (V', A')$) zwei Graphen (bzw. Digraphen) mit $V' \subseteq V$ und $E' \subseteq E$ (bzw. $A' \subseteq A$), so ist G' ein **Teilgraph** von G (bzw. D' ein **Teildigraph** von D). Falls sogar

$$E' = E \cap \binom{V'}{2} \quad \text{bzw.} \quad A' = A \cap (V' \times V')$$

gilt, so heißt der Teilgraph **induzierter Untergraph** (bzw. **induzierter Unterdigraph**). Gilt $V' = V$, so heißt der Teilgraph **spannend**.

Induzierte Knoten-, Kanten-, Bögenmengen

Definition 1.6

Sei $G = (V, E)$ ein Graph. Für Teilmengen $V' \subseteq V$ und $E' \subseteq E$ definieren wir $E(V') := E \cap \binom{V'}{2}$ und

$$V(E') := \{v \in V \mid \{v, w\} \in E' \text{ für ein } w \in V\}$$

Definition 1.7

Sei $D = (V, A)$ ein Graph. Für Teilmengen $V' \subseteq V$ und $A' \subseteq A$ definieren wir $A(V') := A \cap (V' \times V')$ und

$$V(A') := \{v \in V \mid (v, w) \in A' \text{ oder } (w, v) \in A' \text{ für ein } w \in V\}$$

Wege und Kreise in Graphen

Definition 1.8

Für eine endliche Menge $M, c \in \mathbb{R}^M$ und $N \subseteq M$ definieren wir

$$c(N) := \sum_{x \in N} c_x.$$

Definition 1.9

Ist $G = (V, E)$ ein Graph, $c \in \mathbb{R}^E$, und $G' = (V', E')$ ein Untergraph von G, der ein s-t-Weg (bzw. ein Kreis) ist, so heißt $E' \subseteq E$ ein s-t-Weg (bzw. Kreis) in G. Seine c-**Länge** ist $c(E')$. Die Menge der **von E' besuchten Knoten** ist $V' = V(E')$.

Wege und Kreise in Digraphen

Definition 1.10

Ist $D = (V, A)$ ein Digraph, $c \in \mathbb{R}^A$, und $D' = (V', A')$ ein Unterdigraph von D, der ein (gerichteter) s-t-Weg (bzw. ein (gerichteter) Kreis) ist, so heißt $A' \subseteq A$ ein (gerichteter) s-t-Weg (bzw. (gerichteter) Kreis) in D. Seine c-**Länge** ist $c(A')$. Die Menge der **von A' besuchten Knoten** ist $V' = V(A')$.

Wege- und Kreisprobleme

Problem 1.11 (Kürzeste-Wege Problem)

Instanz: *Digraph $D = (V, A)$, $c \in \mathbb{Q}^A$, $s, t \in V$*

Aufgabe: *Finde einen s-t-Weg kürzester c-Länge oder stelle fest, dass es keinen s-t-Weg in D gibt.*

Problem 1.12 (Kürzester-Kreis-Problem)

Instanz: *Digraph $D = (V, A)$, $c \in \mathbb{Q}^A$*

Aufgabe: *Finde einen Kreis kürzester c-Länge oder stelle fest, dass es keinen Kreis in D gibt.*

Ungerichtete Graphen

Bemerkung 1.13

Das Problem, in einem ungerichteten Graphen $G = (V, E)$ kürzeste Wege oder Kreise zu finden, kann man durch Konstruktion des Digraphen $D = (V, A)$ mit

$$A = \{(v, w) \in V \times V \mid \{v, w\} \in E\}$$

auf die Probleme 1.11 bzw. 1.12 zurück führen.

Weitere Bemerkungen

Bemerkung 1.14

- *Analog zum Kürzesten-Wege bzw. Kürzesten-Kreis Problem ist das* **Längste-Wege Problem** *bzw. das* **Längster-Kreis Problem** *definiert.*
- *Durch Multiplikation des Längenvektors c mit (-1) kann man Kürzeste- und Längste-Wege/Kreise-Probleme ineinander transformieren (solange es nicht auf Vorzeichen ankommt).*
- *Kürzester/Längster-Kreis Probleme kann man mit Hilfe von $|A|$ Kürzeste/Längste-Wege Problemen lösen.*
- *Das Kürzeste/Längste-Wege/Kreise Probleme für beliebige D und c sind NP-schwer (z.B. Reduktion von Hamilton-Kreis oder Hamilton-Pfad, s. Übungen).*

Konservative Längen

Definition 1.15

Sei $D = (V, A)$ ein Digraph. Der Längenvektor $c \in \mathbb{R}^A$ heißt **konservativ**, wenn es keinen Kreis negativer c-Länge in D gibt. (d. h. $c(C) \geq 0$ für alle Kreise $C \subseteq A$)

Satz 1.16

Seien $D = (V, A)$ ein Digraph mit konservativen Kantenlängen $c \in \mathbb{R}^A$ und $s, w \in V$. Hat $W \subseteq A$ unter den s-w-Wegen mit kombinatorischer Länge $\leq k$ kürzeste c-Länge und ist $(v, w) \in W$ der letzte Bogen auf W, so hat $W \setminus \{(v, w)\}$ kürzeste c-Länge unter allen s-v-Wegen der kombinatorischen Länge $\leq k - 1$.

Korollar 1.17

Ist $D = (V, A)$ ein Digraph mit konservativen Bogenlängen $c \in \mathbb{R}^A$, und ist $W \subseteq A$ ein c-kürzester s-t-Weg in D, so ist für jeden Knoten $w \in V(W)$ der s-w-Weg $W' \subseteq W$ ein c-kürzester s-w-Weg in D.

Nachbarn, Sterne, Knotengrade

Definition 1.18

Für einen Digraphen $D = (V, A)$ und $v \in V$ definieren wir:

- $\mathrm{N}^{\mathrm{aus}}(v) := \{w \in V \mid (v, w) \in A\}$
- $\mathrm{N}^{\mathrm{ein}}(v) := \{w \in V \mid (w, v) \in A\}$
- $\delta^{\mathrm{aus}}(v) := \{(v, w) \mid w \in \mathrm{N}^{\mathrm{aus}}(v)\}$
- $\delta^{\mathrm{ein}}(v) := \{(w, v) \mid w \in \mathrm{N}^{\mathrm{ein}}(v)\}$

$|\delta^{\mathrm{aus}}(v)|$ und $|\delta^{\mathrm{ein}}(v)|$ sind der **Ausgrad** bzw. der **Eingrad** von v.
Für einen Graphen $G = (V, E)$ und $v \in V$ definieren wir:

- $\mathrm{N}(v) := \{w \in V \mid \{v, w\} \in E\}$
- $\delta(v) := \{\{v, w\} \mid w \in \mathrm{N}(v)\}$

$|\delta(v)|$ ist der **Grad** von v.

Distanzen

Definition 1.19

Für einen Digraphen $D = (V, A)$, $c \in \mathbb{R}^A$ und $s, v \in V$ ist die c-**Distanz** zwischen s und v

$$\mathrm{dist}_c(s, v) \in \mathbb{R} \cup \{\infty\}$$

die Länge eines s-v-Weges minimaler c-Länge bzw. ∞, wenn kein s-v-Weg in D existiert.

Korollar 1.20

Für einen Digraphen $D = (V, A)$ mit konservativen Bogenlängen $c \in \mathbb{R}^A$ gilt für alle $s, v \in V$ mit $s \neq v$:

- $\mathrm{dist}_c(s, s) = 0$
- $\mathrm{dist}_c(s, v) = \min \left\{ \mathrm{dist}_c(s, u) + c_{(u,v)} : u \in \mathrm{N}^{ein}(v) \right\}$

Zusammenhang

Definition 1.21

Ein Graph $G = (V, E)$ heißt **zusammenhängend**, wenn es für jedes Paar $s, t \in V$ einen s-t-Weg in G gibt. Die (inklusions-)maximalen zusammenhängenden Teilgraphen eines Graphen sind seine **Zusammenhangskomponenten**.

Definition 1.22

Für einen Digraphen $D = (V, A)$ heißt $G = (V, E)$ mit

$$E = \{\{v, w\} \mid (v, w) \in A \text{ oder } (w, v) \in A\}$$

der D **zugrunde liegende ungerichtete Graph**.

Definition 1.23

Ein Digraph $D = (V, A)$ heißt **stark zusammenhängend**, wenn es für jedes Paar $s, t \in V$ einen (gerichteten) s-t-Weg in D gibt. Ist der D zugrunde liegende ungerichtete Graph zusammenhängend, so heißt D **schwach zusammenhängend**.

Bäume und Wälder

Definition 1.24

Ein **Wald** ist ein (ungerichteter) Graph, der keine Kreise enthält.
Ein Wald ist ein **Baum**, wenn er zusammenhängend ist.

Bemerkung 1.25

Ein Graph $G = (V, E)$ *mit* $|V| \geq 1$ *ist genau dann ein Baum, wenn er zusammenhängend ist und* $|E| = |V| - 1$ *gilt.*

Branchings und Arboreszenen

Definition 1.26

Ein Digraph $D = (V, A)$ heißt ein **Branching**, wenn er keine antiparallelen Bögen hat, der D zugrunde liegende ungerichtete Graph ein Wald ist und $|\delta^{\text{ein}}(v)| \leq 1$ für alle $v \in V$ gilt. Ein schwach zusammenhängendes Branching ist eine **Arboreszenz**.

Bemerkung 1.27

In einer Arboreszenz $D = (V, A)$ gibt es genau einen Knoten $r \in V$ mit $\mathrm{N}^{ein}(r) = \emptyset$. Er heißt die **Wurzel** *von D. Für jedes $v \in V$ gibt es einen eindeutigen r-v-Weg in D.*

Bemerkung 1.28

Eine Arboreszenz $D = (V, A)$ mit Wurzel $r \in V$ ist eindeutig bestimmt durch die Abbildung $p : V \setminus \{r\} \longrightarrow V$ mit $(p(v), v) \in A$.

Kürzeste-Wege Bäume

Definition 1.29

Für einen Digraphen $D = (V, A)$ und $s \in V$ bezeichnen wir mit $\mathrm{R}_D(s) \subseteq V$ die Menge alle Knoten $v \in V$, für die ein s-v-Weg in D existiert.

Korollar 1.30 (aus Satz 1.16)

Für einen Digraphen $D = (V, A)$ mit konservativen Bogenlängen $c \in \mathbb{R}^A$ und $s \in V$ gibt es eine Arboreszenz $T = (\mathrm{R}_D(s), A')$, so dass für jedes $v \in \mathrm{R}_D(s)$ der s-v-Weg in T ein c-kürzester s-v-Weg in D ist.

Definition 1.31

Eine Arboreszenz wie in Kor. 1.30 heißt ein **Kürzeste-Wege-Baum** (für D, c, s).

Azyklische Digraphen

Definition 1.32

Ein Digraph heißt **azyklisch**, wenn er keinen (gerichteten) Kreis hat.

Definition 1.33

Für $n \in \mathbb{N}$ definieren wir $[n] := \{1, 2, \ldots, n\}$.

Topologische Sortierung

Definition 1.34

Eine **topologische Sortierung** eines Digraphen $D = (V, A)$ ist eine Nummerierung $V = \{v_1, v_2, \ldots, v_n\}$ der Knoten mit

$$(v_i, v_j) \in A \Rightarrow i < j \text{ für alle } i, j \in [n].$$

Satz 1.35

Sei $D = (V, A)$ ein Digraph.

1. *D hat genau dann eine topologische Sortierung, wenn D azyklisch ist.*
2. *Es gibt einen Algorithmus, der für mittels Adjazenzliste gegebenes D in $O(|V| + |A|)$ Zeit eine topologische Sortierung von D berechnet oder feststellt, dass D einen Kreis hat.*

Kürzeste Wege in azyklischen Digraphen

Algorithmus 1.36 (Kürzeste Wege in azyklischen Digraphen)

Eingabe: *Azyklischer Digraph* $D = (V, A)$ *mit topologischer Sortierung* $V = \{v_1, \dots, v_n\}$, $c \in \mathbb{Q}^A$, $\sigma \in [n]$

Ausgabe: *Für jeden Knoten* $v \in V \setminus \{v_\sigma\}$ *den Wert* $d(v) = \mathrm{dist}_c(v_\sigma, v)$ *und, falls* $d(v) \neq \infty$, *den letzten Bogen* $(p(v), v)$ *auf einem* v_σ*-*v*-Weg kürzester* c*-Länge (kürzeste-Wege Baum bzgl.* v_σ*)*

for $i = 1, \dots, n$ **do**
    $d(v_i) \leftarrow \infty$
$d(v_\sigma) \leftarrow 0$
for $i = \sigma + 1, \dots, n$ **do**
    $d(v_i) \leftarrow \min\{d(w) + c_{(w,v_i)} \mid w \in \mathrm{N}^{ein}(v_i)\}$
    $p(v_i) \leftarrow$ *ein* $w \in \mathrm{N}^{ein}(v_i)$ *mit* $d(v_i) = d(w) + c_{(w,v_i)}$
    ($p(v_i)$ *bleibt undefiniert, falls* $d(v_i) = \infty$ *oder* $\mathrm{N}^{ein}(v_i) = \varnothing$*)*

Analyse von Algorithmus 1.36

Satz 1.37

Mit Algorithmus 1.36 kann man für mittels Adjazenzlisten gegebene azyklische gerichtete Graphen $D = (V, A)$ das Kürzeste-Wege Problem (und genauso das Längste-Wege Problem) in $\mathrm{O}(|V| + |A|)$ Zeit lösen (für beliebige $c \in \mathbb{Q}^A$), sogar simultan für alle $t \in V$ (bei festem $s \in V$).

Dijkstras Algorithmus

Algorithmus 1.38 (Dijkstra-Algorithmus)

Eingabe: *Digraph* $D = (V, A)$, $c \in \mathbb{Q}_+^A$, $s \in V$

Ausgabe: *Für alle* $v \in V \setminus \{s\}$ *den Wert* $d(v) = \mathrm{dist}_c(s, v)$ *und, falls* $d(v) \neq \infty$, *den letzten Bogen* $(p(v), v)$ *auf einem s-v-Weg kürzester c-Länge*

```
1:  for all {v ∈ V \ {s}} do
2:      d(v) ← ∞
3:  d(s) ← 0; F ← ∅
4:  Bestimme v ∈ V \ F mit d(v) = min{d(w) | w ∈ V \ F}
5:  F ← F ∪ {v}
6:  for all {w ∈ N^aus(v) \ F} do
7:      if {d(w) > d(v) + c_vw} then
8:          d(w) ← d(v) + c_vw
9:          p(w) ← v
10: if {F ≠ V} then
11:     Gehe zu Schritt 4.
```

Analyse von Algorithmus ??

Satz 1.39

Dijkstras Algorithmus für nicht-negative Bogenlängen arbeitet korrekt und kann so implementiert werden, dass die Laufzeit $O(|A| + |V| \log |V|)$ *ist.*

Bemerkung 1.40

Das Kürzeste-Wege Problem in einem ungerichteten Graphen $G = (V, E)$ *mit nicht-negativen Kantenlängen kann man mit Hilfe des Dijkstra-Algorithmus im Digraphen* $D = (V, A)$ *mit*

$$A = \{(v, w) \in V \times V \mid \{v, w\} \in E\}$$

lösen (jede Kante wird zu zwei Bögen).

Potenziale

Definition 1.41

Seien $D = (V, A)$ ein Digraph und $c \in \mathbb{R}^A$. Ein Vektor $\pi \in \mathbb{R}^V$ heißt ein c-**Potenzial** für D, wenn für alle $(v, w) \in A$

$$\pi_w \leq \pi_v + c_{vw} \quad \text{(d.h. } c_{vw} \geq \pi_w - \pi_v\text{)}$$

gilt. (Vgl. Def. 9.34 Lineare Optimierung)

Bemerkung 1.42

Ist $D = (V, A)$ ein Digraph, $c \in \mathbb{R}^A$, und $\pi \in \mathbb{R}^V$ ein c-Potenzial, so gilt für alle $s, v \in V$

$$\mathrm{dist}_c(s, v) \geq \pi_v - \pi_s.$$

Potenziale und konservative Längen

Bemerkung 1.43

Ist $D = (V, A)$ *ein Digraph,* $c \in \mathbb{R}^A$ *konservativ und gilt* $\mathrm{R}_D(s) = V$ *für ein* $s \in V$*, so ist wegen Korollar 1.20 durch*

$$\pi_v := \mathrm{dist}_c(s, v) \quad \textit{für alle} \quad v \in V$$

ein c-Potenzial auf D *definiert. Für alle* $v \in V$ *ist dann insbesondere (wegen Bem. 1.42)* $\mathrm{dist}_c(s, v)$ *die maximale Potenzialdifferenz* $\sigma_v - \sigma_s$ *über alle c-Potenziale* $\sigma \in \mathbb{R}^V$ *von* D*.*

Satz 1.44

Sei $D = (V, A)$ *ein Digraph. Dann ist* $c \in \mathbb{R}^A$ *genau dann konservativ, wenn es ein c-Potenzial für* D *gibt.*

Pfade

Definition 1.45

Sei $D = (V, A)$ ein Digraph und $c \in \mathbb{R}^A$. Ein l-Tupel

$$P = \big((v_0, v_1), (v_1, v_2), (v_2, v_3), \dots, (v_{l-1}, v_l)\big) \in A^l$$

von Bögen in A heißt ein v_0-v_l-Pfad in D (dabei dürfen einzelne Knoten und einzelne Bögen beliebig oft auftreten) der c-**Länge**

$$c(P) := \sum_{i=1}^{l} c_{v_{i-1}, v_i} \, .$$

Seine **(kombinatorische) Länge** ist l.

Bemerkung 1.46

In einem Digraphen $D = (V, A)$ mit konservativen Bogenlängen $c \in \mathbb{R}^A$ gilt $c(P) \geq \operatorname{dist}_c(s, t)$ für jeden s-t-Pfad P.

Der Bellman-Ford Digraph

Definition 1.47

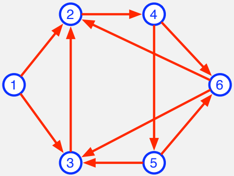
Sei $D = (V, A)$ ein Digraph, $c \in \mathbb{R}^A$. Der Graph $BF(D) = (V', A')$ hat die Knotenmenge $V' = V \times \{0, 1, \ldots, |V|\}$ und die Bogenmenge

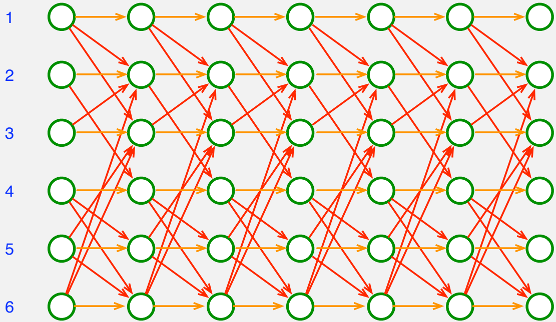
$$A' = \left\{((v, i-1), (w, i)) \,\middle|\, (v, w) \in A,\ i \in [|V|]\right\} \\ \cup \left\{((v, i-1), (v, i)) \,\middle|\, i \in [|V|]\right\}.$$

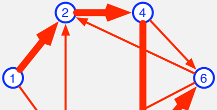
Die Mengen $\{(v, i) : i \in \{0, 1, \ldots, |V|\}\}\,(v \in V)$ sind die **Zeilen** von $BF(D)$. Analog sind $\{(v, i) : v \in V\}\,(i \in \{0, 1, \ldots, |V|\})$ die **Spalten** von $BF(D)$. Die Bögen, welche Knoten einer Zeile verbinden, heißen **waagerecht**. Wir definieren $c' \in \mathbb{R}^{A'}$ via

$$c'_{(v,i),(w,i+1)} = \left\{ \begin{array}{lll} c_{vw} & , & \text{falls} \quad (v, w) \in A \\ 0 & , & \text{falls} \quad v = w \end{array} \right\}.$$

$BF(D)$







Kürzeste Wege in $BF(D)$ (I)

Definition 1.48

- Für $s \in V$ und $v \in \mathrm{R}_D(s)$ sei $P_s(v) \subseteq A'$ der c'-kürzeste $(s,0)$-$(v,|V|)$-Weg in $BF(D)$, der von Algorithmus 1.36 ($BF(D)$ ist azyklisch) berechnet wird, wenn man in Schritt 6 wenn möglich den waagerechten Nachbarn nimmt.
- Für $T \subseteq A'$ sei

  $$A(T) := \{(v,w) \in A \,|\, ((v,i-1),(w,i)) \in T \text{ für ein } i \in [|V|]\}$$

- Ein Weg $P \subseteq A'$ **betritt** Zeile w, wenn es einen Bogen $((v,i-1),(w,i)) \in P$ mit $v \neq w$ gibt.

Kürzeste Wege in $BF(D)$ (II)

Bemerkung 1.49

Betritt $P_s(v)$ keine Zeile mehrmals (und Zeile s gar nicht), so ist $A(P_s(v))$ ein c-kürzester (s, v)-Weg in D.

Lemma 1.50

Besucht $P_s(v)$ die Knoten (w, i) und (w, j) $(i < j)$ und verläuft der Teilweg $Q \subseteq A'$ von $P_s(v)$, der von (w, i) nach (w, j) führt, nicht vollständig in Zeile w, so ist $c'(Q) < 0$.

Kürzeste Wege in $BF(D)$ (III)

Bemerkung 1.51

Wählt man in Lem. 1.50 (w, i) und (w, j) so, dass $P_s(v)$ zwischen diesen beiden Knoten keinen waagerechten Bogen hat und keine Zeile mehrmals und die zu w gehörende Zeile gar nicht betritt, so ist $A(Q)$ ein Kreis negativer c-Länge in D.

Satz 1.52

Für einen Digraphen $D = (V, A)$ und konservative Bogenlängen $c \in \mathbb{Q}^A$ kann man das Kürzeste-Wege Problem in $\mathrm{O}(|V||A|)$ Zeit mit Algorithmus 1.36 auf $BF(D)$ lösen.

Entscheiden von Konservativität

Satz 1.53

Für einen Digraphen $D = (V, A)$ und $c \in \mathbb{Q}^A$ kann man in $O(|V||A|)$ *Zeit ein c-Potenzial oder einen Kreis negativer c-Länge finden, indem man Algorithmus 1.36 auf $BF(\tilde{D})$ laufen lässt, wobei $\tilde{D}$ aus D durch Hinzufügen eines Knotens s und der Bögen (s, v) mit Länge* 0 *für alle $v \in V$ entsteht.*

Das 0/1-Knapsack Problem

Problem 1.54 (0/1-Knapsack Problem)

Instanz: $a \in \mathbb{N}^n$, $\beta \in \mathbb{N}$, $c \in \mathbb{N}^n$

Aufgabe: *Löse* $\max\{\langle c, x\rangle : \langle a, x\rangle \leq \beta,\ x \in \{0,1\}^n\}$

Definition 1.55

Für eine endliche Menge M und $N \subseteq M$ heißt $\chi(N) \in \{0,1\}^M$ mit $\chi(M)_N = \mathbb{1}_N$ und $\chi(M)_{M\setminus N} = \mathbb{O}_{M\setminus N}$ der **charakteristische Vektor** oder der **Inzidenzvektor** von N.

Dynamische Programmierung für 0/1-Knapsack (I)

- (Das 0/1-Knapsack Problem ist *NP* schwer.)
- Definiere einen azyklischen Digraphen $D = (V, A)$ mit

$$V = ([n] \times \{0, 1, \ldots, \beta\}) \cup \{(0, 0)\}$$

und

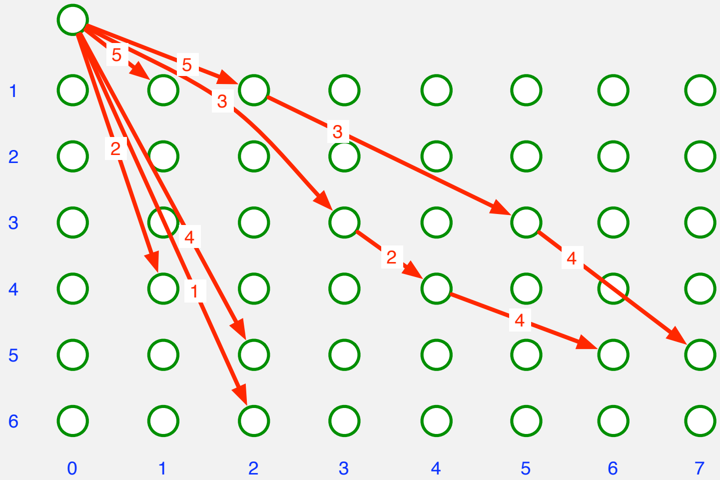
$$((i, g), (j, h)) \in A$$

$$\Longleftrightarrow$$

$$i < j \text{ und } g + a_j = h$$

Für $(v, w) \in A$ mit $v = (i, g)$, $w = (j, h)$ setze $c_{vw} := c_j$.

DP-Graph für 0/1-Knapsack



$$a = (2, 1, 3, 1, 2, 2),\ c = (5, 1, 3, 2, 4, 1),\ \beta = 7$$

Dynamische Programmierung für 0/1-Knapsack (II)

- Für $s = (0,0)$ und $v = (i,g)$ ist dann die c-Länge eines c-längsten s-v-Weges $P \subseteq A$ in D gleich

$$\max\left\{\langle c_{[i]}, \tilde{x}\rangle : \langle a_{[i]}, \tilde{x}\rangle = g,\ \tilde{x} \in \{0,1\}^{[i]}\right\} \qquad (1)$$

  und mit

$$J := \{j \in [n] : P \text{ besucht einen Knoten } (j,h)\}$$

  ist $\chi(J) \in \{0,1\}^n$ eine Optimallösung von (1).
- Ein c-längster s-v-Weg in D (über alle $v \in V$) liefert also eine Optimallösung des 0/1-Knapsackproblems.

Satz 1.56

Das 0/1-Knapsackproblem kann man in $O(n^2\beta)$ Zeit mit dynamischer Programmierung lösen.

Minimum Mean Cycle Problem

Definition 1.57

Für einen Digraphen $D = (V, A)$ und $c \in \mathbb{R}^A$ sei

$$\mu_c(D) := \min \left\{ \frac{c(C)}{|C|} : C \subseteq A \text{ Kreis in } D \right\}$$

($\mu_c(D) = \infty$, falls D azyklisch).

Problem 1.58 (Minimum Mean Cycle Problem)

Instanz: *Digraph $D = (V, A), c \in \mathbb{Q}^A$*

Aufgabe: *Finde einen Kreis $C \subseteq A$ mit $\frac{c(C)}{|C|} = \mu_c(D)$ oder stelle fest, dass D azyklisch ist.*

Der Digraph $K(D)$

- Seien $D = (V, A)$ ein Digraph und $c \in \mathbb{R}^A$, $n := |V|$.
- Definiere $\tilde{D} := (\tilde{V}, \tilde{A})$ mit $\tilde{V} := V \uplus \{s\}$, $\tilde{n} := n + 1$, und

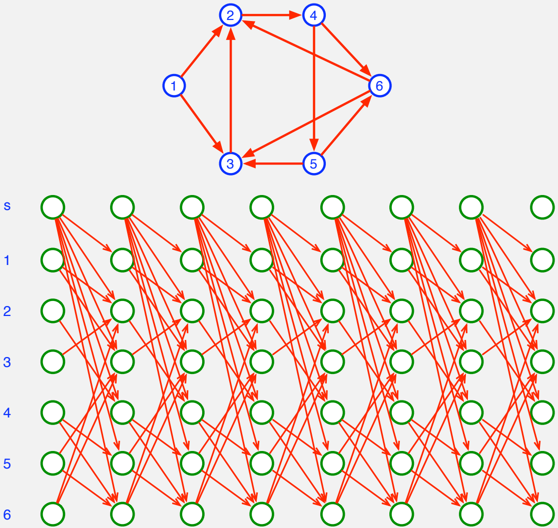
$$\tilde{A} := A \cup \{(s, v) \mid v \in V\}$$

  und $\tilde{c} \in \mathbb{R}^{\tilde{A}}$ mit $\tilde{c}_A := c$, $\tilde{c}_{\tilde{A} \setminus A} := \mathbb{O}_{\tilde{A} \setminus A}$.
- Definiere $K(D) := (V', A')$ mit $V' := \tilde{V} \times \{0, 1, \ldots, \tilde{n}\}$,

$$A' := \{((v, i-1)(w, i)) \mid (v, w) \in \tilde{A},\ i \in [\tilde{n}]\}$$

  und $c' \in \mathbb{R}^{A'}$ mit $c'_{((v,i-1),(w,i))} := \tilde{c}_{vw}$ für alle $(v, w) \in \tilde{A}$, $i \in [\tilde{n}]$.
- Wege der kombinatorischen Länge l in $K(D)$ entsprechen Pfaden der kombinatorischen Länge l in $\tilde{D}$.
- Für $k \in [\tilde{n}]$ und $v \in \tilde{V}$ sei $F_k(v) := \mathrm{dist}_{c'}((s, 0), (v, k))$.

$K(D)$



Der Min Mean Cycle Satz

Satz 1.59

Sei $D = (V, A)$ ein nicht azyklischer Digraph, $c \in \mathbb{R}^A$. Dann gilt

$$\mu_c(D) = \mu_{\tilde{c}}(\tilde{D}) = \min\left\{\max\left\{\frac{F_{\tilde{n}}(v) - F_k(v)}{\tilde{n} - k} \,\middle|\, 0 \leq k \leq \tilde{n} - 1\right\} \,\middle|\, v \in \tilde{V}\right\}. \quad (2)$$

Finden eines Minimum Mean Cycle (I)

- Stelle fest, ob D azyklisch ist.
- Falls D nicht azyklisch ist, sei $v \in \tilde{V}$ ein Knoten, für den das Minimum in Gleichung (2) angenommen wird.
- Sei $\tilde{Q} \subseteq A'$ ein $(s, 0) - (v, \tilde{n})$-Weg mit $c'(\tilde{Q}) = F_{\tilde{n}}(v)$.
- $\tilde{Q}$ muss wenigstens eine Zeile mehrmals besuchen, er besteht also aus einem Anfangsstück $Q_1 \subseteq A'$, einem Endstück $Q_2 \subseteq A'$, und einem Mittelstück $Q \subseteq A'$, das von (w, i) zu $(w, j)\,(i < j)$ führt und keine Zeile mehrmals betritt.

Finden eines Minimum Mean Cycle (II)

- Dann ist $C := \tilde{A}(Q) \subseteq \tilde{A}$ ein Kreis und mit $k^\star := \tilde{n} - |Q|$ gilt

$$c'(Q_1) + c'(Q_2) \geq F_{k^*}(v)$$

(weil man durch Anhängen einer um $|Q|$ Spalten nach links verschobenen Kopie von Q_2 an Q_1 einen $(s,0)$-$(v,k^\star)$-Weg der c'-Länge $c'(Q_1) + c'(Q_2)$ erzeugen kann), also

$$\begin{aligned} \frac{\tilde{c}(C)}{|C|} &= \frac{c'(Q)}{|Q|} = \frac{F_{\tilde{n}}(v) - (c'(Q_1) + c'(Q_2))}{|Q|} \\ &\leq \frac{F_{\tilde{n}}(v) - F_{k^*}(v)}{\tilde{n} - k^*} \leq \mu_{\tilde{c}}(\tilde{D}) \end{aligned}$$

(wobei die letzte Ungleichung wegen Satz 1.59 und der Wahl von v gilt)

- Wegen $\delta^{\text{ein}}(s) = \emptyset$ ist $C \subseteq A$ sogar ein Kreis in D mit

$$\frac{c(C)}{|C|} = \mu_c(D) = \mu_{\tilde{c}}(\tilde{D})\,.$$

Laufzeit des Min Mean Cycle Algorithmus

Satz 1.60

Das Minimum Mean Cycle Problem kann in $O(|V||A|)$ *Zeit gelöst werden.*