

## Diskrete Optimierung im Verkehr (WS 2014)

### Übungsblatt3

Abgabe: Fr, 07. November 2014, in der Übung

#### Aufgabe 1.

5+5 Punkte

Betrachte den Digraphen  $D$  in Abb. 1.

- Bestimme einen maximalen  $(1, 11)$ -Fluss in  $D$ .
- Gib ein Argument an das beweist, dass der in Teilaufgabe a) bestimmte Wert maximal ist.

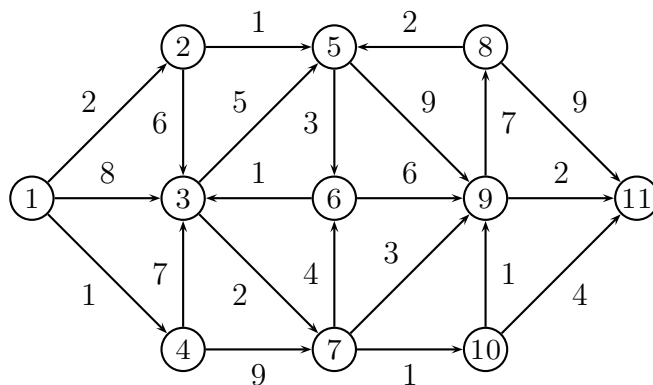


Abbildung 1: Digraph  $D$  mit Kantenkapazitäten.

#### Aufgabe 2.

10 Punkte

Beweisen Sie die Äquivalenz der Bogen- und der Pfadformulierung des Mehrdepot-Umlaufplanungsproblems.

#### Aufgabe 3.

10 Punkte

Betrachte die Bogenformulierung des Umlaufplanungsproblems und seine LP-Relaxierung.

- Konstruiere ein Beispiel, in dem LP- und IP-Optimalwert nicht übereinstimmen.
- Wie schlecht kann das Verhältnis sein?

#### Aufgabe 4.

2+2+2+2+2 Punkte

Wie kann man die Zeitlinienformulierung des Umlaufplanungsproblems modifizieren oder erweitern, so dass folgende Anforderungen behandelt werden können:

- a) Bei langen Wenden mit einer gewissen Mindestdauer können die Fahrzeuge in einer (räumlich von der Endhaltestelle entfernten) Abstellanlage zwischengeparkt werden.
- b) Mindestwendezeiten an den Endhaltestellen dürfen nicht unterschritten werden.
- c) Lange Wenden sollen bevorzugt werden.
- d) Mindestwendezeiten an den Endhaltestellen dürfen bei jeder zweiten Wende nicht unterschritten werden.

Geben Sie die Komplexität des jeweiligen Modells an.

#### Aufgabe 5.

Präsenzübung

Wir verwenden für verschiedene Übungen das niederländische Intercity-Netz aus einem GAMS-Modell von Michael Bussieck. Das Netz ist in Abb. 2 abgebildet.

Die Daten sind in verschiedenen Dateien enthalten (in dieser Übung werden noch nicht alle Daten benötigt):

- *edges.dat* enthält die Bögen des Graphen. Diese sind gerichtet und die Daten enthalten Vorwärts- und Rückwärtsbögen.
- *cap.dat* enthält Kapazitäten für die Bögen.
- *lines.dat* enthält die Knoten von Linien. Die Linien sind ungerichtet, d.h. eine Linie überdeckt immer die Vorwärts- und die Rückwärtsrichtung.
- *load.dat* enthält das Fahrgastaufkommen auf einem Bogen.
- *nodes.dat* enthält eine geordnete Liste von Knoten des Graphen (dies ist nützlich, um die OD-Matrix einzulesen).
- *od.dat* enthält das Fahrgastaufkommen auf den OD-Relationen.
- *times.dat* enthält Fahrzeiten für die Bögen.

Die Namen der Stationen werden wie folgt abgekürzt:

Ah	Arnhem	Lls	Lelystad Centrum
Apd	Apeldoorn	Lw	Leeuwarden
Asd	Amsterdam CS	Mt	Maastricht
Asdz	Amsterdam Zuid WTC	Odzg	Oldenzaal Grens
Asn	Assen	Rsdg	Rosendaal Grens
Bd	Breda	Rtd	Rotterdam CS
Ehv	Eindhoven	Shl	Schiphol
Gn	Groningen	Std	Sittard
Gv	Den Haag HS	Ut	Utrecht CS
Gvc	Den Haag CS	Zl	Zwolle
Hgl	Hengelo	Zvg	Zevenaar Grens
Hr	Heerenveen		

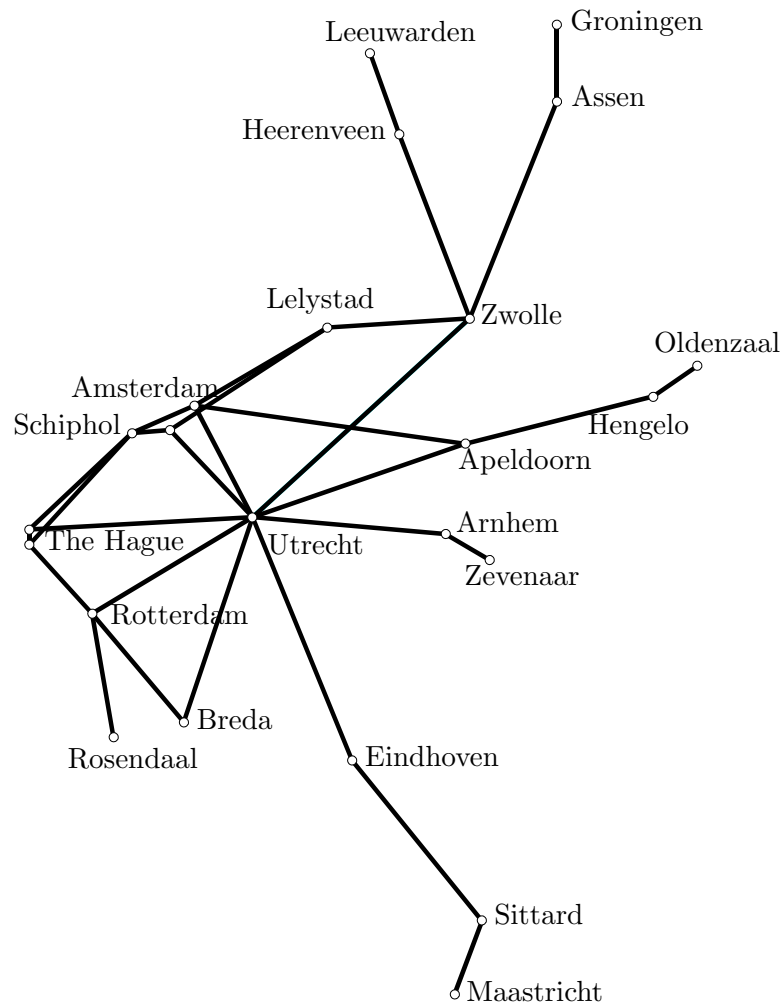


Abbildung 2: Das niederländische Intercity-Netz nach M. Bussieck.

Verwende ZIMPL und SCIP, um die folgenden Optimierungsprobleme zu lösen.

- a) Berechne einen maximalen Fluss bezüglich der gegebenen Kapazitäten zwischen Amsterdam (Asd) und Breda (Bd) im niederländischen Intercity-Netz. Zeichne den Fluss in die Karte ein.
- b) Finde einen Mehrgüterfluss, der für jedes Gut  $d \in T$  eine Nachfrage  $v^d$  von einer Quelle  $s^d$  zur Senke  $t^d$  befriedigt und die Gesamtreisezeit minimiert.
- c) Löse folgende Variante von b). Maximiere einen Skalierungsfaktor  $\lambda$ , so dass für jedes Gut  $d \in T$  eine Nachfrage von  $\lambda v^d$  transportiert werden kann. Wie groß kann  $\lambda$  sein?

Hier sind einige Hinweise, die die Verarbeitung der Daten mit ZIMPL erleichtern sollen:

- Benutze das Kommando `read` um Dateien zu lesen. Lies z.B. die Bögen wie folgt ein:

```
set E := { read "edges.dat" as "<1s,2s>" comment "#" };
```

Auch Parameter können eingelesen werden. Lies beispielsweise die Kapazitäten wie folgt ein:

```
param cap[E] := read "cap.dat" as "<1s,2s> 3n" comment "#";
```

Die OD-Matrix kann wie folgt gelesen werden:

```
param d[0*0] := read "od.dat" as "n+" comment "#";
```

Dabei sind 0 die OD-Knoten, die aus der Datei *od.dat* eingelesen werden müssen.