

## Combinatorische Optimalisatie (X401067), Tentamen 30-05-2013.

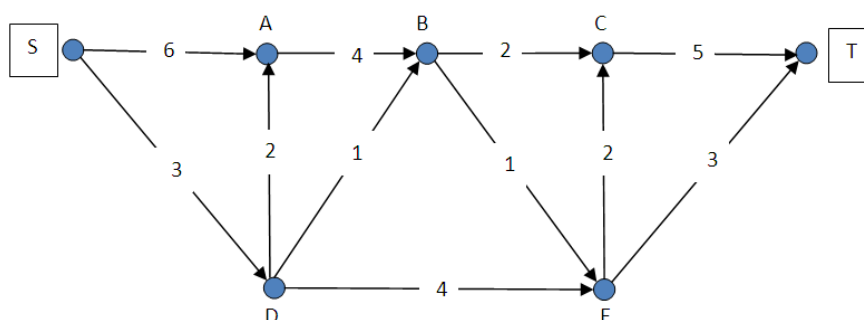
Het tentamen bestaat uit 8 vragen die opgesplitst zijn in 12 subvragen. Voor elke subvraag zijn maximaal 5 punten te behalen. Het resultaat wordt verkregen door het totaal door 6 te delen.

Tijdens het examen zijn GEEN boeken, aantekeningen, rekenmachines etc. toegestaan. Alleen pen en papier.

Opgavenblad mag meegenomen worden. Antwoorden worden (uiterlijk) de dag na het tentamen op de webpagina gezet.

| 1 | 2 | 3a | 3b | 4a | 4b | 5 | 6 | 7a | 7b | 8a | 8b | $\Sigma$ |
|---|---|----|----|----|----|---|---|----|----|----|----|----------|
| 5 | 5 | 5  | 5  | 5  | 5  | 5 | 5 | 5  | 5  | 5  | 5  | 60       |

### Opgave 1.



In bovenstaand netwerk staat op elke pijl de capaciteit van de pijl. Geef een minimale  $S, T$ -snede en een maximale  $S, T$ -stroom. (Geef niet alleen de waarde maar geef ook aan hoe de snede en stroom loopt. Bijvoorbeeld door de figuur over te nemen. )

**Opgave 2.** Onderstaande matrix geeft de afstanden tussen de punten in het gerichte netwerk met punten  $a, b, c, d$ . Vind de de lengtes van de kortste (gerichte) paden tussen elk tweetal punten door het Floyd-Warshall algoritme toe te passen.

| $c_{ij}$ | $a$      | $b$      | $c$      | $d$      |
|----------|----------|----------|----------|----------|
| $a$      | $\infty$ | 5        | -1       | 2        |
| $b$      | 4        | $\infty$ | 3        | 1        |
| $c$      | 1        | 6        | $\infty$ | 9        |
| $d$      | 10       | 3        | 5        | $\infty$ |

### Opgave 3. .

#### VERTEX COVER:

*Instantie:* Een graaf  $G = (V, E)$

*Oplossing:* Een verzameling  $S \subseteq V$  zodat elke lijn minstens één eindpunt heeft in  $S$ .

*Doel:* Minimaliseer  $|S|$ .

(a) Geef een integer linear programming (ILP) formulering voor het vertex cover probleem. Geef ook de LP-relaxatie.

(b) Geef  **twee**  verschillende 2-approximatie algoritmes voor het vertex cover probleem. Toon voor minimaal één van de twee aan dat dit inderdaad een 2-approximatie is.

### Opgave 4. .

Het beslissingsprobleem INDEPENDENT SET is als volgt gedefinieerd.

#### INDEPENDENT SET:

*Instantie:* Graaf  $G = (V, E)$  en een geheel getal  $K$

*Vraag:* Bestaat er een  $U \subseteq V$  met  $|U| \geq K$  en zodat er geen lijn bestaat met beide eindpunten in  $U$ .

(a) Geef voor het vertex cover probleem uit opgave 3 het bijbehorende beslissingsprobleem en beargumenteer dat vertex cover in  $\mathcal{NP}$  zit.

(b) Stel dat we weten dat INDEPENDENT SET  $\mathcal{NP}$ -compleet is. Toon aan dat VERTEX COVER ook  $\mathcal{NP}$ -compleet is.

*Hint:* Laat eerst het volgende zien: Als  $S$  een vertex cover in  $G = (V, E)$  is dan is  $U = V \setminus S$  een onafhankelijke verzameling (independent set) in  $G$  en omgekeerd.

### Opgave 5. .

#### RESTRICTED MAX-CUT:

*Instantie:* Een graaf  $G = (V, E)$  op  $|V| = 3k$  punten.

*Oplossing:* Deelverzamelingen  $S \subset V$  met  $|S| = k$ .

*Doel:* Maximaliseer het aantal lijnen in de snede  $(S, \bar{S})$ . Dat wil zeggen: maximaliseer het aantal lijnen dat precies één eindpunt in  $S$  heeft.

Het restricted max-cut probleem is dus gelijk aan het standaard max-cut probleem met de restrictie dat  $|V|$  deelbaar is door 3 en dat  $|S| = |V|/3$ .

Geef een (randomized)  $\frac{4}{9}$ -approximatie algoritme voor dit probleem. (Toon dit ook aan)

**Opgave 6.** Beschouw het scheduling probleem  $1|r_j|\sum_j C_j$ . (Ter herinnering: Er is 1 machine en elke job  $j$  heeft een gegeven process time  $p_j$  en een release time  $r_j$ . Een job mag niet eerder starten dan zijn release time. Doel is om een schedule te vinden dat de som van de completeringstijden  $C_j$  minimaliseert.)

Neem voor het gemak aan dat er minimaal 1 job  $j$  is met  $r_j = 0$ . Beschouw het volgende SPT algoritme (Shortest Process Time first): Op tijdstip 0, start de job met de kortste process time onder de jobs die al beschikbaar (released) zijn. Op het eerste moment dat de machine weer vrij is en er jobs beschikbaar zijn, start weer de job met de kleinste process time onder de jobs die al beschikbaar zijn, enz.

Voorbeeld:  $p_1 = 1, r_1 = 0$  en  $p_2 = 3, r_2 = 0$ , en  $p_3 = 2, r_3 = 1$ . Het SPT algoritme zet de jobs in de (optimale) volgorde job 1, job 3, job 2.

Laat met een ander voorbeeld zien dat dit algoritme niet altijd optimaal is.

**Opgave 7.** Beschouw het online paging probleem. (Ter Herinnering : De cache kan  $k$  pages bevatten. Een input sequence  $\sigma = (p_1, p_2, \dots)$  arriveert online. Als een page  $p_j$  al in de cache zit dan zijn er geen kosten voor deze page. Als deze niet in de cache zit dan moet deze eerst in de cache gebracht worden en (indien de cache vol was) er een andere page uit de cache gehaald worden. De kosten voor deze page zijn in dat geval 1. )

FIFO: First-In-First-Out

LIFO: Last-In First-Out.

(a) Laat met een voorbeeld sequence zien dat FIFO niet beter dan  $k$ -competitive is voor het paging probleem.

(b) Voor FIFO kan men aantonen dat het niet slechter dan  $k$ -competitive is. Voor LIFO geldt dit niet. Laat met een voorbeeld sequence zien dat LIFO strict slechter dan  $k$ -competitive is voor het paging probleem.

### Opgave 8.

#### MAX-DIRECTED CUT:

*Instantie:* Een gerichte graaf  $G = (V, A)$ .

*Oplossing:* Een verzameling  $S \subset V$ .

*Doel:* Maximaliseer het aantal lijnen in de snede  $(S, \bar{S})$ , dat wil zeggen, het aantal lijnen  $(i, j)$  met beginpunt  $i \in S$  en eindpunt  $j \in \bar{S} = V \setminus S$ .

$$\begin{aligned} \max \quad & \sum_{(i,j) \in A} z_{ij} \\ \text{s.t.} \quad & z_{ij} \leq x_i \quad \text{voor alle } (i, j) \in A \\ & z_{ij} \leq 1 - x_j \quad \text{voor alle } (i, j) \in A \\ & x_i \in \{0, 1\} \quad \text{voor alle } i \in V \\ & 0 \leq z_{ij} \leq 1 \quad \text{voor alle } (i, j) \in A. \end{aligned}$$

**a.** Laat zien dat bovenstaand mixed ILP een exacte formulering is voor het Max-Directed Cut probleem. (Met andere woorden, laat zien dat er voor elke maximale snede er een oplossing voor het mixed-ILP bestaat met gelijke waarde en laat anderzijds zien dat er voor elke optimale oplossing van het mixed-ILP er een snede bestaat met gelijke waarde.)

**b.** Beschouw het volgende randomized approximatie algoritme. Los de LP-relaxatie van het ILP probleem op. Voor elke  $i \in V$ , kies  $i \in S$  met kans  $1/4 + x_i/2$ . Laat zien dat dit een randomized  $\frac{1}{2}$ -approximatie geeft. *Hint:* Laat zien dat  $\Pr(\text{lijn } (i, j) \text{ in snede}) \geq z_{ij}/2$ .