

Tentamen Combinatorische Optimalisatie 29-05-2015.

* Boeken, aantekeningen etc. zijn niet toegestaan.

* Rekenmachines niet toegestaan.

* Tentamen mag worden meegenomen, Antwoorden staan na tentamen op BB.

* Beschrijf bij elke opgave (behalve bij 1) steeds het belangrijkste idee. Een gedetailleerd bewijs wordt niet gevraagd. Ook notatie en exacte formulering is van minder belang. Een tekening kan soms handig zijn.

Opgave	1	2	3	4	5	6	7	Σ
Max pntn	5	10	10	10	5	10	15	65

Cijfer = punten/6.5

Er zijn meerdere definities van α -*approximatie*. Voor de duidelijkheid, in dit tentamen gaan we uit van de volgende (standaard) definitie. Een algoritme is:

- een α -*approximatie* voor een minimization problem als de waarde van de gevonden oplossing altijd hooguit α maal de optimale waarde is ($\alpha \geq 1$),
 - een α -*approximatie* voor een maximization problem als de waarde van de gevonden oplossing altijd minimaal α maal de optimale waarde is, ($\alpha \leq 1$).
-

Opgave 1 (5 punten).

Beantwoord de volgende vragen alleen met ja (waar), nee (niet waar) of laat open. Goed antwoord: +1 punt; Fout antwoord: -0.5 punt; Geen antwoord: +0 punt. (Minimaal aantal punten is 0 punten.)

- (a) $4n^2 + (\log n)^3 = O(n^2)$
- (b) Als er een polynomiaal algoritme voor TSP bestaat dan moet gelden $P = NP$.
- (c) Als $P \neq NP$ dan bestaat is géén polynomiaal algoritme voor TSP.
- (d) Het volgende probleem kan in polynomiale tijd beslist worden: Gegeven een graaf $G = (V, E)$, is het waar dat elke cykel in G minimaal 100 punten heeft? (cykel wordt circuit genoemd in het dictaat van Schrijver)
- (e) Laat Π_1 en Π_2 beslissingsproblemen in NP zijn. Als Π_1 reduceert tot Π_2 en Π_2 is NP-compleet dan moet gelden dat Π_1 ook NP-compleet is.

Opgave 2 (5+5=10 punten) In het college zijn vijf verschillende 2-approximatie algoritmes voor het metrische TSP probleem gegeven.

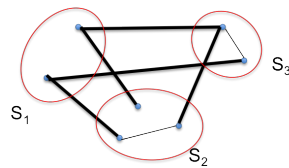
- Beschrijf er twee. Een uitleg van het belangrijkste idee is voldoende. Pseudo code wordt niet gevraagd.
- Pas de twee algoritmes uit (a) toe op onderstaande instantie. Ga uit van Euclidische afstanden, dus afstanden zijn zoals getekend op papier. Als het algoritme begint met één punt, neem dan punt 1.



Opgave 3 (3+7=10 punten) In het (ongewogen) maximale snede probleem (max-cut problem) is het de bedoeling om voor een graaf $G = (V, E)$ een deelverzameling $S \subseteq V$ van de punten te vinden zodat het aantal lijnen van S naar $V \setminus S$ maximaal is.

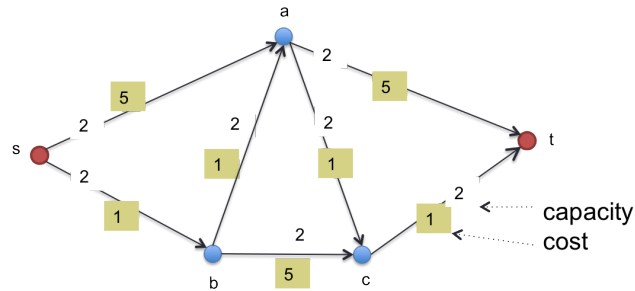
- Beschrijf een randomized 0.5-approximatie algoritme. (Alleen beschrijving. Bewijs wordt in (b) gevraagd.)

In het maximale k -snede probleem ($k \geq 2$) is het de bedoeling om de punten van V in k verzameling $S_1, \dots, S_k$ te verdelen zodat het aantal lijnen dat tussen verschillende verzamelingen loopt maximaal is. In onderstaand voorbeeld zijn deze lijnen vet gedrukt.



- Geef een randomized $(1 - 1/k)$ -approximatie algoritme en beargumenteer waarom de ratio $(1 - 1/k)$ is.

Opgave 4 (10 punten) In onderstaande netwerk laten we een stroom van waarde 2 lopen over het pad s, a, t . De kosten van de deze stroom zijn $2 \cdot (5 + 5) = 20$. Dit is duidelijk niet de goedkoopste s, t stroom met waarde 2. Op college is een algoritme gegeven dat bij een beginstroom met waarde v , een stroom met waarde v van minimale kosten vindt. Pas dit algoritme toe om voor het netwerk in de figuur de minimale kosten stroom met waarde 2 te vinden. Neem als beginstroom een stroom over s, a, t met waarde 2.



Opgave 5 (5 punten) Pas het greedy algoritme voor het k -center probleem toe op onderstaande instantie. Ga uit van de Euclidische afstanden. Neem $k = 4$ en neem punt 2 als eerste center. Je hoeft alleen de indices van de 4 centers in de gekozen volgorde op te schrijven: 2,

1	2	3	4
5	6	7	8
9	10	11	12

Opgave 6 (5+5=10 punten) Het unrelated machine scheduling probleem op m machines ($Rm||C_{max}$) is NP-hard. Op de slides en in het dictaat van Krumke wordt een algoritme gegeven welke altijd een oplossing vindt waarvan de waarde hooguit twee maal de optimale waarde is. (Een 2-approximatie algoritme). In stap 1 wordt een LP opgelost.

- Geef deze LP-formulering. (Deze vraag heb je al eerder gezien)
- Stel nu dat sommige paren jobs niet op dezelfde machine gedaan mogen worden. Deze restrictie is gegeven in de vorm van een graaf $G = (V, E)$: V zijn de jobs en er is een lijn $(j, k) \in E$ als job j en k niet aan dezelfde machine mogen worden toegewezen. Voeg deze restrictie toe aan het LP van (a).

Opgave 7 (5+2+5+3=15 punten) Beschouw de volgende twee optimaliseringsproblemen:

- Een *vertex cover* in een graaf $G = (V, E)$ is een verzameling punten $S \subseteq V$ zodat voor elke lijn $e \in E$ minstens één van de twee eindpunten in S zit. Het *minimum vertex cover problem* is het vinden van een vertex cover met zo min mogelijk punten.
- Een *independent set* in een graaf $G = (V, E)$ is een verzameling punten $T \subseteq V$ zodat geen enkele lijn $e \in E$ beide eindpunten in T heeft. Het *maximum independent set problem* is het vinden van een independent set met zo veel mogelijk punten.

Beide problemen kunnen ook als beslissingsprobleem geformuleerd worden: Gegeven een graaf G en getal k , dan is het de vraag of er een vertex cover (independent set) met waarde k bestaat.

- (a) (5ptn) Stel dat je al weet dat het vertex cover probleem NP-compleet is. Laat door middel van een reductie zien dat independent set ook NP-compleet is.

Stel je bezoekt een festival. Er zijn n voorstellingen en voorstelling i begint op tijd a_i en duurt tot tijd b_i . Je wilt zoveel mogelijk voorstellingen bijwonen maar het probleem is dat de tijdsintervallen elkaar kunnen overlappen. (Het kost geen tijd om van een locatie naar een andere locatie te gaan op het festival.)

- (b) (2ptn) Hoe zou je dit kunnen formuleren als een independent set probleem?
- (c) (5ptn) Geef een dynamisch programmeringsformulering voor dit probleem. Wat is de running time?
Hint: Neem aan dat $a_1 < a_2 < \dots < a_n$. Definieer $F(i)$ als het maximale aantal voorstellingen dat je kunt bijwonen als je op tijd a_i aankomt bij het festival en definieer $j_i = \min\{j \mid a_j \geq b_i\}$.
- (d) (3ptn) In (b) heb je laten zien dat het festival probleem te schrijven is als het (NP-complete) independent set probleem en in (c) heb je (als het goed is) een polynomiaal algoritme gegeven. Is dit niet tegenstrijdig met het feit dat NP-complete problemen naar alle waarschijnlijkheid niet in polynomiale tijd kunnen worden opgelost? Leg uit.