

Hertentamen combinatorische optimalisatie 09-07-2014.

Tentamen is met ‘gesloten boek’. Het tentamen wordt binnen 2 dagen nagekeken.

Beschrijf bij elke opgave steeds het belangrijkste idee. Notatie en exacte formulering is van minder belang. Een tekening kan soms handig zijn. Ook voor een half antwoord kun je punten krijgen.

Opgave	1	2	3	4	5	6	7	8	Σ
Max pntn	5	5	5	5	5	5	5	5	40

Opgave 1 (5 punten).

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

- (a) Het traveling salesman probleem (TSP) is NP -moeilijk (NP -hard).
- (b) $P \subseteq NP$.
- (c) Stel Π is een NP -compleet beslissingsprobleem. Dan weten we zeker dat Π niet in polynomiale tijd oplosbaar is. (Met andere woorden, deze stelling is ooit bewezen.)
- (d) Stel Π_1 en Π_2 zijn beslissingsproblemen en zijn beiden NP -compleet. Dan geldt dat Π_1 reduceert tot Π_2 en omgekeerd, Π_2 reduceert tot Π_1 .
- (e) Het kortste pad probleem (shortest path) is reduceerbaar tot het traveling salesman probleem (TSP).

Opgave 2 (5 punten) Bij scheduling problemen met due dates (deadlines) is de *lateness* L_j van een job j gedefinieerd als de completion time minus de due date: $L_j = C_j - d_j$. Beschouw onderstaande instantie van het probleem $1||L_{max}$. (Dwz, er is 1 machine, elke job j heeft een process time p_j en due date d_j en het doel is om een volgorde van de jobs te vinden zodat de maximale lateness ($\max_j L_j$) geminimaliseerd wordt. Geef een optimaal schedule en z'n waarde voor deze instantie en beschrijf ook hoe het optimale algoritme werkt.

jobs	1	2	3	4
p_j	3	4	5	6
d_j	11	5	3	12

Opgave 3 (5 punten)

Laat zien dat een *minimale* opspannende boom (MST) ook een opspannende boom is waarvan de maximale edge weight minimaal is. Met andere woorden, als T een MST is van graaf G is en e is een lijn in T met gewicht $w(e)$, dan heeft elke opspannende boom een lijn met een gewicht van minstens $w(e)$.

Opgave 4 (5 punten) Bij optimaliseringsproblemen kunnen we onderscheid maken tussen het kunnen vinden van een optimale oplossing en het kunnen oplossen van het beslissingsprobleem. Stel we hebben een algoritme A dat het volgende beslissingsprobleem oplost: Gegeven een graaf G , geeft het algoritme ons antwoord op de vraag of G wel of geen Hamiltonian cycle heeft. Leg uit hoe je met dit algoritme A in polynomiale tijd een Hamilton cycle kunt vinden in het geval G Hamiltonian is. Neem aan dat elke aanroep van A één tijdseenheid kost.

Opgave 5 (5 punten) Gegeven is een verzameling $A = \{a_1, a_2, \dots, a_n\}$ van n getallen en gegeven is een getal M . We willen weten of het mogelijk is om M te schrijven als een geheel-tallige lineaire combinatie van getallen in A . Met andere woorden, we willen weten of er getallen $x_1, \dots, x_n \in \{0, 1, 2, \dots\}$ zijn zodat $A = x_1 a_1 + \dots + x_n a_n$.

Voorbeeld: $A = \{5, 9, 13\}$. Als $M = 19$ dan is dit mogelijk want $19 = 2 \cdot 5 + 9$. Als $M = 26$ dan is dit ook mogelijk want $26 = 2 \cdot 13$. Maar bijvoorbeeld $M = 21$ heeft geen oplossing.

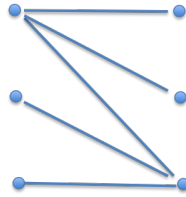
Geef een $O(nM)$ dynamic-programming algoritme voor dit probleem.

Hint: Voor alle $m \in \{0, 1, \dots, M\}$ definieer $H(m) = 1$ als het mogelijk is om m op zo'n manier te schrijven en laat $H(m) = 0$ als dat niet mogelijk is.

(Je hoeft geen algoritme uit te schrijven maar beschrijf alleen de essentie. Welke waarden staan er in de dynamic programming tabel? Hoe worden deze berekend? Waarom kost dit alles $O(nM)$ tijd?)

Opgave 6 (5 punten) Het max-flow probleem en het matching probleem zijn nauwverbonden. Ter herinnering: Het algoritme van Ford en Fulkerson vindt een maximale s, t -stroom in een gericht netwerk met capaciteiten op de pijlen. Een matching in een graaf $G = (V, E)$ is een verzameling lijnen $M \subseteq E$ zodat elk punt van V eindpunt is van maximaal 1 lijn van M .

Stel je hebt een algoritme (bijvoorbeeld Ford-Fulkerson) voor het max-flow probleem. Laat zien hoe je dit kunt gebruiken om in een bipartite graaf een maximale matching te vinden. Het volstaat om deze reductie toe te passen op onderstaand voorbeeld. (Neem over op je antwoordblad.) Met andere woorden, wat is het max-flow probleem bij onderstaande instantie van het maximale matching probleem?



Opgave 7 (5 punten) Beschrijf minimaal twee verschillende 2-approximatie algoritmes voor het TSP probleem. De beschrijving is voldoende. Er wordt *niet* gevraagd om de factor 2 te bewijzen.

Opgave 8 (5 punten: 1+2+2) Het volgende integer program (IP) is een exacte formulering voor het vinden van een grootste matching in een graaf $G = (V, E)$. (De notatie $v \in e$ betekent hier dat punt v een eindpunt is van edge e .)

$$\begin{aligned}
 \text{(IP) maximize } & Z = \sum_{e \in E} y_e \\
 \text{subject to } & \sum_{e: v \in e} y_e \leq 1 \quad \text{for all } v \in V, \\
 & y_e \in \{0, 1\} \quad \text{for all } e \in E.
 \end{aligned}$$

- Leg kort uit in woorden waarom dit IP een exacte formulering is van het grootste matching probleem.
- In de LP-relaxatie wordt $y_e \in \{0, 1\}$ vervangen door $y_e \geq 0$. Geef een graaf waarvoor de optimale waarde van de LP-relaxatie strict groter is dan de grootste matching in de graaf. Wat is de LP-oplossing en wat is de grootste matching in je voorbeeld?

Hint: Er is een voorbeeld mogelijk met slechts enkele punten.

- Neem nu aan dat $G = (V, E)$ een bipartite graaf is. Stel $Z^*, \mathbf{y}^*$ is een optimale oplossing voor de LP-relaxatie en neem aan dat het half-integer is, dat betekent, $y_e^* \in \{0, \frac{1}{2}, 1\}$ for all $e \in E$. Laat zien hoe je deze oplossing $\mathbf{y}^*$ stap voor stap kunt veranderen in een oplossing $\mathbf{y}$ met $y_e \in \{0, 1\}$ met dezelfde (optimale) waarde.

(*Hint: Bekijk de graaf $G^* = (V, E^*)$, waarbij E^* bestaat uit alle lijnen e waarvoor $y_e^* \in \{\frac{1}{2}, 1\}$. Laat eerst zien dat de maximale graad in G^* twee is. Laat vervolgens zien hoe je de y -waarden voor de lijnen in G^* kunt aanpassen.*)