

## Prototype Tentamen

### Opgave 1

Beschouw het LP-probleem

$$\begin{array}{ll}\max & 2x_1 + 5x_2 + 9x_3 \\ \text{onder} & 9x_1 + 6x_2 + 3x_3 \leq 10 \\ & x_1 + 15x_2 + 6x_3 \leq 25 \\ & x_1, x_2, x_3 \geq 0\end{array}$$

- (a) Breng dit LP-probleem in de standaard basisvorm. Geef de startoplossing voor de simplex-algoritme en pas **één** iteratie toe om vanuit deze startoplossing een verbeterde basisoplossing te vinden. Geef op overzichtelijke en beargumenteerde wijze aan hoe u deze verbeterde basisoplossing construeert en wat de bijbehorende basisvorm van het LP-probleem is.
- (b) Wat zijn de nieuwe basisvariabelen? Wat zijn hun waarden en wat is de nieuwe criteriumwaarde? Hoe kun je van de nieuwe basisvorm aflezen of de gevonden oplossing optimaal is? Is dit hier het geval?
- (c) Formuleer van het oorspronkelijke LP-probleem in ongelijkheidsvorm het duale LP-probleem. Wat kun je zeggen over de minimale criteriumwaarde van het duale probleem?
- (d) Zonder het primale helemaal op te lossen, laat zien, door inspectie van het duale probleem, of het primale probleem onbegrensd is.

**Opgave 2.** *Bedenk je goed dat je voor goede onderdelen van het model in deze opgave al punten kunt verdienen*

Manon R. maakt voor het bedrijf Burger Meister elke week  $N$  soorten hamburger sauzen, genummerd  $j = 1, 2, \dots, N$ . Elke saus wordt gemaakt door het mengen van  $M$  ingrediënten, genummerd  $i = 1, 2, \dots, M$ . Er geldt dat  $N > M$ . Hoewel een zekere mate van vrijheid is toegestaan bij het maken van de sauzen, moet er voldaan worden aan de volgende eisen: elke saus moet voor tenminste 10% bestaan uit ingrediënt 1. Saus 2 moet voor tenminste 25% uit ingrediënt 3 bestaan. Verder eist Burger Meister dat er 4 maal zoveel van saus 1 geleverd moet worden als van saus 2. Per week kan ten

hoogste  $b_i$  liter van ingrediënt  $i$  besteld worden,  $i = 1, 2, \dots, M$ . Manon kan tegen een prijs van  $w_j$  euro per liter van saus  $j$  net zoveel afzetten van de sauzen als dat zij produceert,  $j = 1, 2, \dots, N$ . Ingrediënt  $i$  kost een bedrag  $c_i$  euro per liter om in te kopen,  $i = 1, 2, \dots, M$ .

Formuleer dit probleem als een lineair programmeringsprobleem zodat Manon haar netto winst op de sauzen kan maximaliseren.

*Hint: introduceer voor  $i = 1, 2, \dots, M, j = 1, 2, \dots, N$  de beslissingsvariabelen:*

$x_{ij}$  = aantal liters van ingrediënt  $i$  dat Manon per week inkoopt voor saus  $j$ .

**Opgave 3.** *Bedenk je goed dat je voor goede onderdelen van het model in deze opgave al punten kunt verdienen*

Holland Seinen heeft een contract getekend om 1000 nachtkijkers te leveren voor een vredesmissie. Voor de productie van de nachtkijkers heeft het bedrijf vier locaties ter beschikking. Vanwege lokale omstandigheden zijn de kosten voor het produceren van een nachtkijker verschillend per locatie. De benodigde arbeidskracht en het benodigde materiaal voor het maken van een nachtkijker is ook verschillend per locatie. Deze gegevens betreffende de productie per nachtkijker staan in de volgende tabel:

| locatie | kosten | arbeidsuren | eenheden materiaal |
|---------|--------|-------------|--------------------|
| 1       | 15     | 2           | 3                  |
| 2       | 10     | 3           | 4                  |
| 3       | 9      | 4           | 5                  |
| 4       | 7      | 5           | 6                  |

Om de locaties gereed te maken voor het produceren van verrekijkers moeten bepaalde ruimtes stofvrij gemaakt worden. Dit vraagt, voor ieder van de locaties, eenmalig een aantal uren arbeid, indien besloten wordt in de locatie te produceren, onafhankelijk van het aantal verrekijkers dat dan op de locatie gemaakt zal worden. Dit zijn 10 uren voor Locatie 1, 15 voor Locatie 2, 8 voor Locatie 3, en 12 voor Locatie 4.

Voor de vier productielocaties tezamen zijn in totaal 3300 arbeidsuren ter beschikking en 4000 eenheden materiaal. De opdrachtgever

heeft in het contract geëist dat tenminste 400 nachtkijkers worden geproduceerd op locatie drie.

Formuleer het probleem van het vinden van de oplossing die de totale winst maximaliseert als een Geheeltallig Linear Programmerings probleem.

#### Opgave 4: Branch & Bound

Beschouw het volgende 0-1 knapzak probleem.

$$\begin{array}{ll}\max & 2.5y_1 + 3y_2 + 5y_3 + 8y_4 + 5y_5 \\ \text{onder} & 3y_1 + y_2 + 3y_3 + 5y_4 + 4y_5 \leq 10 \\ & y_i \in \{0, 1\}, \quad i = 1, \dots, 5\end{array}$$

**4A)** Geef de oplossing van de LP Relaxatie van dit probleem [dat wil zeggen: geef de oplossing van het probleem als  $y_i \in \{0, 1\}$ ,  $i = 1, \dots, 5$ , vervangen zou worden door  $0 \leq y_i \leq 1$ ,  $i = 1, \dots, 5$ ]. Geef ook de bijbehorende optimale criteriumwaarde.

**4B)** Los het bovenstaande 0-1 knapzak probleem op met Branch en Bound en vind op deze manier de **twee** optimale oplossingen.

#### Opgave 5: Geheeltallige Programmering

Daan H. wil op een zo goedkoop mogelijke manier een computer aanschaffen door alle onderdelen los te kopen. Er zijn in totaal  $M$  verschillende onderdelen nodig [van elk onderdeel **precies één**], aangegeven met nummers 1 t/m  $M$ , die te koop zijn in  $N$  verschillende (internet)winkels.

Onderdeel  $i$  kost in winkel  $j$  een bedrag  $c_{ij}$  [ $i = 1, \dots, M$ ;  $j = 1, \dots, N$ ]. **Als** Daan in winkel  $j$  één of meerdere onderdelen koopt moet hij sowieso een bedrag  $v_j$  betalen [ $j = 1, \dots, N$ ] (bijvoorbeeld voor verzendkosten). Daan vraagt zich af in welke winkels hij de  $M$  onderdelen moet aanschaffen ten einde de totale aanschafkosten zo laag mogelijk te laten zijn.

**5A)** Om dit probleem te formuleren als een geheeltallig lineair programmerings- probleem, introduceren we binaire beslissingsvariabelen  $x_{ij}$  en  $y_j$ . Geef nu duidelijk de betekenis van de gekozen beslissingsvariabelen aan.

**5B)** Formuleer dit probleem als een geheeltallig lineair programmeringsprobleem.

**5C)** Welke restrictie(s) moet(en) worden toegevoegd als Daan bij winkel  $N$  hoogstens  $k$  onderdelen mag aanschaffen? Welke restrictie(s) moet(en) worden toegevoegd als Daan bij hoogstens  $R$  winkels onderdelen mag aanschaffen [met  $R$  een gegeven waarde en  $R < N$ ]? Welke restrictie(s) moet(en) worden toegevoegd als Daan onderdelen 3 en 4 bij winkel 6 wil [moet] aanschaffen indien Daan onderdeel  $M$  **niet** bij winkel 2 aanschafft?

**5D) [BONUS]** Welke restrictie(s) moet(en) worden toegevoegd als Daan bij **tenminste** 3 verschillende winkels **maximaal** voor een bedrag van  $b$  per winkel aan onderdelen wil kopen?

### Opgave 6: Dynamische Programmering

Beschouw opnieuw het oorspronkelijke probleem van Daan H. uit **Opgave 5**.

**6A)** Formuleer dit probleem als een dynamisch programmeringsprobleem. Beschrijf daarvoor in ieder geval de toestand, de interpretatie van de waardefunctie in woorden, de recursierelatie, de start-recursie en wat je moet berekenen in termen van de waardefunctie.

**Aanwijzing:** *Ga 1 voor 1 de winkels langs! Beschrijf nu eerst stadium (fase)  $j$ .*

**6B)** Formuleer het bovenstaande probleem opnieuw als een dynamisch programmerings- probleem wanneer er de **extra eis** is dat er bij hoogstens  $R$  winkels onderdelen mogen worden aangeschaft. Beschrijf daarvoor in ieder geval de toestand, de interpretatie van de waardefunctie in woorden, de recursierelatie, de start-recursie en wat je moet berekenen in termen van de waardefunctie.

**Aanwijzing:** *Gebruik de gelijkheid:*

$$\min_{A \subseteq S} \{g(S \setminus A)\} = \min \left\{ \min_{A \subseteq S, A \neq \emptyset} \{g(S \setminus A)\}, g(S) \right\}$$

### Opgave 7: Netwerken

**7A)** Beschouw een gerichte graaf  $G = (X, E)$  waarbij  $X$  de verzameling steden is in het netwerk:  $X = \{1, \dots, N\}$ . Een pijl van stad  $i$  naar stad  $j$  wordt genoteerd met  $(i, j)$ . Laat  $E \subseteq X \times X$  (Cartesisch Product) de verzameling pijlen zijn waarover het mogelijk is om over te reizen. Laat het niet-negatieve getal  $c_{ij}$  met  $(i, j) \in E$  de afstand zijn om van stad  $i$  rechtstreeks naar stad  $j$  te gaan. Formuleer nu het algemene kortste pad probleem als een **geheeltallig lineair programmeringsprobleem** zodat het kortste pad berekend wordt van stad 1 naar stad  $N$ . Geef duidelijk de betekenis van de beslissingsvariabelen aan en geef een korte toelichting op de constraints.

**7B)** Veronderstel dat je in een gericht netwerk met  $n$  knooppunten  $N_1, \dots, N_n$  het meest veilige pad wil bepalen voor het verzenden van een boodschap van een gegeven knooppunt  $N_s$  naar een ander knooppunt  $N_e$ . De boodschap gaat verloren met kans  $p_{ij}$  wanneer de boodschap langs de directe verbinding van knooppunt  $N_i$  naar knooppunt  $N_j$  gezonden wordt. Hoe kun je dit probleem terugbrengen tot het standaard kortste-pad probleem van Dijkstra met 'afstanden'  $c_{ij}$ ?

### Opgave 8

De rijen van de volgende tabel representeren personen en de kolommen representeren jobs. Een getal in rij  $i$  en kolom  $j$  geeft aan wat de kosten zijn indien persoon  $i$  job  $j$  doet. Los het toewijzingsprobleem op. Laat daarbij duidelijk de stappen zien die je maakt om tot de oplossing te komen.

|   | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | 9 | 5 | 4 | 7 |
| 2 | 4 | 3 | 3 | 9 |
| 3 | 1 | 3 | 2 | 5 |
| 4 | 5 | 8 | 1 | 4 |

### Opgave 9.

**Opgave 9a:** Beschrijf formeel het algoritme van Prim voor het oplossen van het Minimum Opspannende Boom probleem.

**Opgave 9b:** Leid de rekentijd van dit algoritme af als functie van het aantal knopen van de graaf. Geef uiteindelijk de rekentijd weer in termen van de grote  $O$  notatie.

### Opgave 10.

Gegeven is de ongerichte graaf op 11 knopen met kanten en lengtes op de kanten zoals aangegeven in de volgende tabel. Bepaal de optimale Chinese Postbode route op deze graaf. Laat duidelijk zien welke stappen je maakt om tot de oplossing te komen.

| kant         | lengte | kant               | lengte |
|--------------|--------|--------------------|--------|
| $(v_1, v_2)$ | 5      | $(v_6, v_5)$       | 1      |
| $(v_1, v_3)$ | 5      | $(v_6, v_7)$       | 1      |
| $(v_1, v_4)$ | 5      | $(v_6, v_8)$       | 2      |
| $(v_2, v_3)$ | 2      | $(v_6, v_9)$       | 3      |
| $(v_2, v_5)$ | 3      | $(v_7, v_{10})$    | 3      |
| $(v_2, v_6)$ | 2      | $(v_8, v_9)$       | 2      |
| $(v_3, v_6)$ | 5      | $(v_8, v_{11})$    | 5      |
| $(v_3, v_7)$ | 2      | $(v_9, v_{10})$    | 2      |
| $(v_4, v_7)$ | 3      | $(v_9, v_{11})$    | 4      |
| $(v_5, v_8)$ | 5      | $(v_{10}, v_{11})$ | 4      |