

1. Gegeven is het onderstaande LP-probleem (P)

$$\max z = 4x_1 + 3x_2 + 7x_3$$

$$2x_1 - x_2 + 3x_3 \geq 25$$

$$4x_1 + 2x_2 - x_3 = 52$$

$$6x_1 + 3x_2 + x_3 \leq 80$$

$$x_1 \geq 0, x_2 \geq 0, x_3 \geq 0.$$

Het programma SIMOPT geeft de onderstaande output voor het probleem (P).

Last Iteration

	X1	X2	X3	Surp.1	Art.1	Art.2	Slack3	RHS
Rel.Cost	??	??	??	??	??	??	??	??
X1	??	??	??	-0.25	0.25	0.50	-0.25	??
X2	??	??	??	0.50	-0.50	-0.80	0.70	??
X3	??	??	??	0.00	0.00	-0.60	0.40	??

Value Objective Function : 60.3000

Variable	Activity Level	Reduced Cost
X1 :	12.2500	0.0000
X2 :	1.9000	0.0000
X3 :	0.8000	0.0000

	Slack of Surplus	Shadow prices
Constraint 1	0.0000	-0.5000
Constraint 2	0.0000	-4.6000
Constraint 3	0.0000	3.9000

Objective Coefficient Ranges

Variable	Current Coefficient	Allowed Interval
X1	4.0000	[-Infinity , 6.0000]
X2	3.0000	[2.0000 , Infinity]
X3	7.0000	[????? , ?????]

Right-Hand-Side Ranges

Constraint	Current RHS	Allowed Interval
1.	25.0000	[????? , ?????]
2.	52.0000	[27.5000 , 53.3333]
3.	80.0000	[78.0000 , 129.0000]

- (a) Formuleer het duale probleem (D) van (P) en geef de *volledige* oplossingen van zowel (P) als (D). Vul ook het eindtableau volledig in.
- (b) Wijzig in (P) de rechterkantcoëfficiënt van de eerste bijvoorwaarde van 25 in $25 + \theta$ en bereken de nieuwe volledige oplossingen van zowel (P) als (D) als functie van θ behorende bij het huidige eindtableau. Bereken ook het toegelaten interval voor de rechterkantcoëfficiënt van de eerste bijvoorwaarde.
- (c) Wijzig in (P) de opbrengstcoëfficiënt van x_3 van 7 in $7 + \eta$ en bereken de nieuwe volledige oplossingen van (P) en (D) als functie van η behorende bij het huidige eindtableau. Bereken weer het toegelaten interval voor de opbrengstcoëfficiënt van x_3 .
- (d) We wijzigen nu (uitgaande van (P)) de opbrengstcoëfficiënten van x_1 en x_2 van 4 en 3 in resp. $4 + 4\eta$ en $3 - 5\eta$. Bereken de nieuwe volledige oplossingen van (P) en (D) als functie van η en bereken voor welke waarden van η de optimale oplossing *niet wezenlijk* verandert.

2. In Bovenland bevinden zich N universiteiten. Elke universiteit bestaat uit K faculteiten. In faculteit i van universiteit j studeren m_{ij} studenten ($i = 1, 2, \dots, K; j = 1, 2, \dots, N$). Laat $m_j := \sum_{i=1}^K m_{ij}$ het aantal studenten op universiteit j zijn en $M := \sum_{j=1}^N m_j$ het totale aantal studenten in Bovenland. Voor het samenstellen van een landelijke studentenraad, waarin alle universiteiten vertegenwoordigd moeten zijn, vraagt men zich af hoeveel studenten van elke afzonderlijke faculteit op te nemen. Enerzijds wil men dat grote faculteiten ruim vertegenwoordigd zijn, maar men wil voorkomen dat een kleine universiteit oververtegenwoordigd wordt doordat er toevallig een klein aantal grote faculteiten aanwezig zijn. Zowel de grootte van een universiteit als de grootte van een faculteit wordt gedefinieerd als het totale aantal studenten op de universiteit resp. faculteit. De studentenraad bestaat uit 150 leden. Stel men besluit uit faculteit i van universiteit j een aantal van x_{ij} studenten in de raad op te nemen. Dan zijn er in totaal $y_j := \sum_{i=1}^K x_{ij}$ studenten van universiteit j in de raad vertegenwoordigd. Men streeft er nu in eerste instantie naar dat het aantal vertegenwoordigers van universiteit j (dus y_j) dicht ligt bij het 'ideale aantal' $\frac{m_j}{M} 150$. Precies gezegd, men wil van elke universiteit zoveel studenten in de raad opnemen dat de grootste absolute afwijking van het ideale aantal zo klein mogelijk is.

- (a) Formuleer dit probleem als een DP-(dynamisch programmerings)probleem. Beschrijf duidelijk alle elementen van de DP-algoritme (stadia, toestand, beslissing, etc.)

Vervolgens is bij elke universiteit nog de vraag hoeveel studenten te betrekken uit de diverse faculteiten. Ook nu kunnen we spreken van ideale aantallen: voor faculteit i van universiteit j is dit ideale aantal $\frac{m_{ij}}{m_j} y_j$ (merk op dat de y_j nu vast liggen, omdat de 'optimale' waarden reeds in onderdeel (a) bepaald zijn). Men wil nu de som van *al* deze afwijkingen zo klein mogelijk maken. M.a.w. we willen het aantal studenten van universiteit j uit faculteit i (dit is x_{ij}) zó kiezen dat

$$\sum_{i=1}^K \sum_{j=1}^N \left| x_{ij} - \frac{m_{ij}}{m_j} y_j \right| \quad (*)$$

minimaal is.

- (b) Formuleer dit probleem als een DP-probleem met K stadia. Beschrijf expliciet de toestand in elk stadium, de begintoestand, en de eindtoestand, evenals de beslissingen, etcetera.

Omdat het resultaat van deze 'getrapte' aanpak (eerst de y_j bepalen en daarna pas de x_{ij}) niemand bevalt besluit men nu *simultaan* de y_j en de x_{ij} te bepalen onder minimalisering van de hierboven geformuleerde criteriumfunctie (*).

- (c) Formuleer dit probleem als een DP-probleem met KN stadia. Geef opnieuw expliciet alle relevante elementen. [Aanw. Neem als stadium nu (i, j) ; laat de achtereenvolgende stadia zijn $(1, 1), (2, 1), \dots, (K, 1), (1, 2), (2, 2), \dots, (K, 2), \dots, (1, N), (2, N), \dots, (K, N)$.]
3. Gokverslaafde Anton B. heeft 500 euro tot zijn beschikking om in te zetten voor het eenmalig spelen van een bijzonder kaartspel tegen de beruchte weekeraarster Diana K. Eerst wordt Anton gevraagd zijn geld te verdelen (niet noodzakelijk het volledige bedrag) over zes kaarten, Negen, Tien, Jack, Queen, King en Aas. Vervolgens kiest Diana een getal i ($i = 1, 2, 3, 4$) en haar keuze bepaalt nu hoeveel Anton krijgt uitbetaald: voor elke euro ingezet op kaart j ($j = \text{Negen, Tien, Jack, Queen, King, Aas}$) krijgt hij a_{ij} euro van Diana in geval zij het getal i gekozen heeft (als a_{ij} negatief is, dan betaalt Anton aan Diana het bedrag $-a_{ij}$). Omdat Anton niet weet welk getal Diana gaat kiezen wil hij zijn geld zó verdelen dat het laagste totaalbedrag dat hij zal ontvangen zo groot mogelijk is. In onderstaande tabel staan de getallen a_{ij} . Formuleer het probleem van Anton als een lineair programmeringsprobleem.

Diana's keuze i	winst (of verlies) per euro ingezet op kaart j					
	Negen	Tien	Jack	Queen	King	Aas
1	8	2	-6	4	-7	15
2	-2	7	5	-3	9	4
3	4	-5	3	-9	10	-8
4	5	4	12	6	-2	2

4. Een gerichte graaf bestaat uit N knopen $1, 2, \dots, N$. Met elke kant (i, j) is een positief geheel getal q_{ij} verbonden. We zijn geïnteresseerd in een zogenaamde *maximale toegelaten stroomvector* van knoop $s = 1$ (de bron) naar knoop $t = N$ (de put).
- (a) Beschrijf dit probleem als een mathematisch programmeringsprobleem.
- (b) Beschrijf in detail de algoritme waarmee dit probleem efficiënt kan worden opgelost. Hoe heet deze algoritme?

Bouwbedrijf Triplex heeft voor de komende M maanden een K -tal projecten aangeboden gekregen. Elk project wordt gekenmerkt door drie gegevens ($i = 1, 2, \dots, K$),

- vroegste startdatum van project i is het begin van maand m_i ,
- uiterste opleverdatum van project i is het eind van maand n_i ,
- het uitvoeren van project i vereist d_i maanden, $1 \leq d_i \leq n_i - m_i + 1$.

Men kan aan het eind van elke maand het werk aan een project voor één of meer maanden onderbreken. Tijdens maand j heeft Triplex b_j bouwvakkers tot zijn beschikking, en in elke maand waarin aan project i gewerkt wordt moeten *precies* c_i bouwvakkers aan project i werken. Een bouwvakker kan in een maand aan slechts één project tegelijk werken. De vraag is of bouwbedrijf Triplex alle projecten de komende M maanden kan uitvoeren.

- (c) Formuleer deze vraag met behulp van een mathematisch programmeringsprobleem.
- (d) Leg gedetailleerd uit hoe de vraag van Triplex kan worden beantwoord met de in onderdeel (b) beschreven algoritme.
5. Voor de komende M weken wil het bedrijf Multi-Item een optimaal voorraad/productie schema vaststellen. Multi-Item produceert L verschillende producten. De vraag naar product i in week j is aan het begin van de planningsperiode (begin van week 1) reeds bekend en gelijk aan β_{ij} eenheden ($i = 1, \dots, L; j = 1, \dots, M$). Om aan alle vraag te kunnen voldoen

wordt telkens aan het begin van een week besloten voor welke producten een productierun gestart wordt. Het is mogelijk voor meerdere producten op hetzelfde tijdstip een productierun te starten. Een productierun voor product i brengt opstartkosten K_i met zich mee. Elke productierun duurt minder dan een week, onafhankelijk van de hoeveelheden die in een run geproduceerd worden. Alles wat in week j geproduceerd wordt kan nog dezelfde week gebruikt worden om te voldoen aan de vraag in week j . Eventuele overschotten worden tegen betaling van opslagkosten in voorraad gehouden. De opslagkosten voor eenheden van product i bedragen h_i per product per week. Er mag geen enkele vraag verloren gaan en de variabele productiekosten worden gedurende de planningsperiode constant verondersteld. Het probleem is voor elk product vast te stellen, wanneer en hoeveel te produceren ten einde de totale kosten (opstartkosten plus opslagkosten) zo laag mogelijk te laten zijn.

- (a) Beargumenteer waarom het nooit optimaal kan zijn een productierun te starten die niet bestaat uit de som van een aantal weekvragen. Welke efficiënte DP-methode kunnen we nu gebruiken (voor elk product i apart)? Formuleer deze aanpak in algemene termen.
- (b) Formuleer voor product 1 de in onderdeel (a) bedoelde algoritme in *detail*.

Nu blijkt dat de variabele productiekosten van week tot week verschillen. Precies gezegd, het produceren van één eenheid van product i in week j kost een bedrag b_{ij} . Opnieuw willen we het optimale productieschema bepalen waarvoor de *totale* kosten over de M weken (opstartkosten plus opslagkosten plus variabele kosten) minimaal zijn.

- (c) Kunnen we nog steeds de methode uit onderdeel (a) gebruiken? Geef een korte (informele) argumentatie.

Tevens blijkt de opslagruimte beperkt te zijn: op elk moment is in totaal niet meer beschikbaar dan een oppervlakte A en één eenheid van product i vereist aan opslagruimte een oppervlakte α_i . Dit heeft tot gevolg dat Multi-Item in zijn productieruns beperkt is: men kan aan het begin van een week nooit besluiten meer te produceren dan de nog beschikbare ruimte toelaat (A minus de ruimte reeds in gebruik voor de totale voorraad van eerder geproduceerde, maar nog niet geleverde eenheden), dit ondanks het gegeven dat nieuw geproduceerde eenheden nog in dezelfde week kunnen worden gebruikt voor de vraag in die week.

- (d) Formuleer een geheeltallig programmeringsprobleem waarmee nu een optimaal productieschema kan worden berekend. *Aanw.* Neem als beslissingsvariabelen,

x_{ijk} = het aantal producten i te produceren in week j bestemd voor de vraag in week k ,

$$y_{ij} = \begin{cases} 1 & \text{als in week } j \text{ een productierun voor product } i \text{ gestart wordt,} \\ 0 & \text{anders.} \end{cases}$$

Puntenwaardering

opgave	a.	b.	c.	d.
1.	5	10	10	10
2.	10	10	5	-
3.	15	-	-	-
4.	5	5	5	10
5.	5	10	5	10

Het eindcijfer over de opgaven 1 t/m 5 wordt als volgt bepaald,

$$\text{eindcijfer} = \max \left(1, \min \left(10, \text{round} \left(\frac{\text{puntentotaal}}{11} \right) \right) \right).$$

Merk op dat voor een 10 slechts 105 punten vereist zijn (van de 130)! Een weggevertje!!