Voorbeeld van een uitwerking van een lineaire programmering

ir. G.C.Meijerman

Het volgende voorbeeld is ontleend aan Earl O'Heady and Wilfred Candler "Linear programming methods" page 68.

Er is van uitgegaan dat er drie beperkende productiemiddelen zijn, te weten land, arbeid in periode I en arbeid in periode II. De "beschikbaarheden" aan land en arbeid zijn quantitatief bekend.

Op het bedrijf kunnen 4 gewassen worden geteeld en wel maïs, haver, sojabonen en tarwe. Al deze gewassen doen aanspraken op de beschikbaarheden (beperkingen) en leveren een saldo op. Deze aanspraken worden in één of andere éénheid - in dit geval een éénheid van land - uitgedrukt.

De beschikbaarheden en de aanspraken van de gewassen (activiteiten) daarop worden in onderstaande tabel weergegeven.

	Beschikbaar	Maïs	Haver	Sojabonen	Tarwe
Land	100	1	1	1	1
Arbeid periode I	100	0	1	0	0,5
Arbeid periode II	80	1	0	2	0

De saldi (C) van de eventueel te telen gewassen zijn voor maïs, haver, sojabonen en tarwe achtereenvolgens 30, 10, 40 en 12.

Het probleem van de lineaire programmering nu is het bepalen van een optimaal bouwplan (optimale combinatie van activiteiten), zodat:

- Het totaal saldo (voor het gehele bedrijf) maximaal is;
- De beschikbaarheden (aan land en arbeid) niet worden overschreden.

We noemen de gewassen achtereenvolgens P_1, P_2, P_3 en P_4. Dit zijn vectoren.

$$P_1 = \begin{pmatrix}1\\0\\1\end{pmatrix}, \quad P_2 = \begin{pmatrix}1\\1\\0\end{pmatrix}, \quad P_3 = \begin{pmatrix}1\\0\\2\end{pmatrix} \quad \text{en } P_4 = \begin{pmatrix}1\\0{,}5\\0\end{pmatrix}$$

De kolom van beschikbaarheden is de vector P_0 of wel $\begin{pmatrix}100\\100\\80\end{pmatrix}$

106/0561/150

We hebben nu een <u>stelsel van ongelijkheden</u>, namelijk

$$P_0 \geqslant x_1P_1 + x_2P_2 + x_3P_3 + x_4P_4$$

x_1 tot en met x_4 zijn de niveaus, waarop de activiteiten worden ontplooid.

Bovenstaande vergelijking is ook op te schrijven als

$$\begin{pmatrix}100\\100\\80\end{pmatrix} \geqslant x_1 \begin{pmatrix}1\\0\\1\end{pmatrix} + x_2 \begin{pmatrix}1\\1\\0\end{pmatrix} + x_3 \begin{pmatrix}1\\0\\2\end{pmatrix} + x_4 \begin{pmatrix}1\\0{,}5\\0\end{pmatrix}$$

of meer gebruikelijk als

$$\begin{aligned}100 &\geqslant x_1.1 + x_2.1 + x_3.1 + x_4.1\\ 100 &\geqslant x_1.0 + x_2.1 + x_3.0 + x_4.0{,}5\\ 80 &\geqslant x_1.1 + x_2.0 + x_3.2 + x_4.0\end{aligned}$$

Van dit stelsel van ongelijkheden maken we een stelsel van gelijkheden door toevoeging van de vectoren P_5, P_6, P_7 (de kunstmatige activiteiten).

Daarbij is $P_5 = \begin{pmatrix}1\\0\\0\end{pmatrix}$, $P_6 = \begin{pmatrix}0\\1\\0\end{pmatrix}$ en $P_7 = \begin{pmatrix}0\\0\\1\end{pmatrix}$

Dit betekent respectievelijk het "niet gebruiken" of "ter beschikking houden" van één eenheid land, één eenheid arbeid im periode I en één eenheid arbeid in periode II. Dit levert uiteraard een saldo C = 0 op.

Het stelsel van gelijkheden is nu:

$$P_0 = x_1P_1 + x_2P_2 + x_3P_3 + x_4P_4 + x_5P_5 + x_6P_6 + x_7P_7$$

of wel

$$\begin{pmatrix}100\\100\\80\end{pmatrix} = x_1 \begin{pmatrix}1\\0\\1\end{pmatrix} + x_2 \begin{pmatrix}1\\1\\0\end{pmatrix} + x_3 \begin{pmatrix}1\\0\\2\end{pmatrix} + x_4 \begin{pmatrix}1\\0{,}5\\0\end{pmatrix} + x_5 \begin{pmatrix}1\\0\\0\end{pmatrix} + x_6 \begin{pmatrix}0\\1\\0\end{pmatrix} + x_7 \begin{pmatrix}0\\0\\1\end{pmatrix}$$

of meer gebruikelijk geschreven

$$\begin{aligned}100 &= x_1.1 + x_2.1 + x_3.1 + x_4.1 + x_5.1 + x_6.0 + x_7.0\\ 100 &= x_1.0 + x_2.1 + x_3.0 + x_4.0{,}5 + x_5.0 + x_6.1 + x_7.0\\ 80 &= x_1.1 + x_2.0 + x_3.2 + x_4.0 + x_5.0 + x_6.0 + x_7.1\end{aligned}$$

We hebben hier drie vergelijkingen met 7 onbekenden. Er zijn dus vele oplossingen. Het gaat er nu om die oplossing te kiezen (die niveaus van

x_1, x_2 enz. t/m x_7) zodat het totaal saldo maximaal is, of wel $x_1C_1 + x_2C_2 + x_3C_3 + x_4C_4 + x_5C_5 + x_6C_6 + x_7C_7$ = maximaal.

De tabel kan als volgt worden opgeschreven:

C →		30	10	40	12			
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
land	100	1	1	1	1	1	0	0
arbeid periode I	100	0	1	0	0,5	0	1	0
arbeid periode II	80	1	0	2	0	0	0	1

Wat er nu gebeurt is essentieel. We gaan elk van deze (kolom) vectoren (P_0 t/m P_7) uitdrukken als een lineaire combinatie van drie vectoren (de basis). In eerste instantie kiezen we hiervoor P_5, P_6 en P_7. Dan krijgen we

C →		30	10	40	12			
	P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
P_5	100	1	1	1	1	1	0	0
P_6	100	0	1	0	0,5	0	1	0
P_7	80	1	0	2	0	0	0	1

Hier staat onder andere:

$$P_0 = 100\ P_5 + 100\ P_6 + 80\ P_7$$

of wel:

$$\begin{pmatrix}100\\100\\80\end{pmatrix} = 100\begin{pmatrix}1\\0\\0\end{pmatrix} + 100\begin{pmatrix}0\\1\\0\end{pmatrix} + 80\begin{pmatrix}0\\0\\1\end{pmatrix}$$

Ook staat er $P_2 = 1\ P_5 + 1\ P_6 + 0\ P_7$

of wel:

$$\begin{pmatrix}1\\1\\0\end{pmatrix} = 1\begin{pmatrix}1\\0\\0\end{pmatrix} + 1\begin{pmatrix}0\\1\\0\end{pmatrix} + 0\begin{pmatrix}0\\0\\1\end{pmatrix}$$

Voorts $P_1 = 1\ P_5 + 0\ P_6 + 1\ P_7$ en $P_5 = 1\ P_5 + 0\ P_6 + 0\ P_7$

$P_3 = 1\ P_5 + 0\ P_6 + 2\ P_7$ $\qquad P_6 = 0\ P_5 + 1\ P_6 + 0\ P_7$

$P_4 = 1\ P_5 + 0{,}5\ P_6 + 0\ P_7$ $\qquad P_7 = 0\ P_5 + 0\ P_6 + 1\ P_7$

In de P_0 kolom staat nu het eerste programma. De kunstmatige activiteiten P_5, P_6 en P_7 worden ontplooid op niveaus van respectievelijk 100, 100 en 80.

Dit betekent:

- 100 éénheden land worden niet gebruikt;
- 100 éénheden arbeid in periode I worden niet gebruikt;
- 80 éénheden arbeid in periode II worden niet gebruikt.

Vanzelfsprekend levert dit eerste (tussen) programma geen saldo op, of wel C = 0.

Dit plan gaan we nu verbeteren. Daartoe werden twee regels aan de tabel toegevoegd, die we de Z en de Z-C regel noemen.

Z zijn de zogenaamde "alternatieve kosten", namelijk datgene, wat we moeten opofferen om een éénheid van een activiteit te ontplooien. In het tussenprogramma hoeven we bij ontplooiïng van andere activiteiten alleen het "niet gebruiken" van land en arbeid op te offeren. Dit kost uiteraard niets (C_s). In de begintabel (het begintableau) is Z dus voor elke activiteit gelijk aan 0. Op deze Z wordt het saldo C in mindering gebracht, namelijk wat het oplevert, wanneer we de betrokken activiteit op niveau 1 ontwikkelen. Zolang Z-C voor enige activiteit negatief is zal ontplooiïng van die activiteit het totaal saldo verhogen en is het programma dus (nog) niet optimaal.

Ons begintableau ziet er nu als volgt uit:

	C →		30	10	40	12	0	0	0
C_s ↓		P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	P_5	100	1	1	1	1	1	0	0
0	P_6	100	0	1	0	0,5	0	1	0
0	P_7	80	1	0	2	0	0	0	1
	Z	0	0	0	0	0	0	0	0
	Z-C	0	-30	-10	-40	-12	0	0	0

Nu kunnen we beginnen met de verbetering van het bovenstaande plan 0. We kiezen daarvoor de activiteit, die de hoogste bijdrage per éénheid oplevert. Dat is P_3, waarvoor de absolute waarde van Z-C het grootst is.

- P_3 op niveau 100 put alle land (P_5) uit;
- Wat arbeid in periode I (P_6) betreft kan P_3 op onbeperkt niveau worden ontwikkeld. Hier is dus geen beperking.
- P_3 op niveau 40 put alle arbeid in periode II (P_7) uit. Dit is de effectieve beperking (het knelpunt); meer dan 40 éénheden P_3 zullen

nooit kunnen worden ontwikkeld, althans zolang we niet meer arbeid in periodeII ter beschikking hebben. We kunnen derhalve 40 éénheden P_3 in het (tussen)programma brengen.

Bij de eerste iteratietrap brengen we P_3 in de basis (in het programma) en verwijderen we P_7 (de beperking) uit het programma.

We gaan daarna alle activiteiten uitdrukken als een lineaire combinatie van de vectoren in de nieuwe basis dus als een lineaire combinatie van P_5, P_6 en P_3.

Voor de P_0 kolom is dit zeer eenvoudig; als we P_3 op niveau 40 ontplooien is P_7 zoals gezegd geheel op; van P_5 blijven 60 éénheden over en P_6, waarop geen aanspraak wordt gedaan blijft intact op niveau 100 in het programma.

Het uitdrukken van de werkelijke activiteiten in de nieuwe basis is iets ingewikkelder. Hiertoe gaan we eerst de kunstmatige activiteiten(P_5,P_6 en P_7)in de nieuwe basis uitdrukken en met behulp daarvan vervolgens de werkelijke activiteiten.

- P_5 verandert niet $\quad P_5 = 1\ P_5 + 0\ P_6 + 0\ P_7$
- P_6 evenmin $\quad P_6 = 0\ P_5 + 1\ P_6 + 0\ P_7$
- P_7 is nu uit de basis gehaald en P_7 moet worden uitgedrukt als een lineaire combinatie van P_5, P_6 en P_3.

Dit gaat als volgt:

P_3 was gelijk aan $1\ P_5 + 0\ P_6 + 2\ P_7$ (begintableau)

dus $2\ P_7 = -\ P_5 - 0\ P_6 + P_3$

of $P_7 = -\ 0{,}5\ P_5 + 0\ P_6 + 0{,}5\ P_3$

Daaruit volgt dat

$P_1 = 1\ P_5 + 0\ P_6 + 1\ P_7$ (begintableau)

$P_1 = 1\ P_5 + 0\ P_6 - 0{,}5\ P_5 + 0\ P_6 + 0{,}5\ P_3$

$P_1 = 0{,}5\ P_5 + 0\ P_6 + 0{,}5\ P_3$

Op dezelfde wijze valt te berekenen dat

$P_2 = 1\ P_5 + 1\ P_6 + 0\ P_3$ (deze verandert niet)

$P_3 = 0\ P_5 + 0\ P_6 + 1\ P_3$ (in zichzelf thans uitgedrukt)

$P_4 = 1\ P_5 + 0{,}5\ P_6 + 0\ P_3$

$P_0 = 60\ P_5 + 100\ P_6 + 40\ P_3$ (was al eerder op een andere wijze bepaald)

Deze getallem vullen we in het nieuwe tableau in, dat er nu als volgt uitziet:

C_s		P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
0	P_5	60	0,5	1	0	1	1	0	- 0,5
0	P_6	100	0	1	0	0,5	0	1	0
40	P_3	40	0,5	0	1	0	0	0	+ 0,5
	Z	1600	20	0	40	0	0	0	20
	Z-C	1600	- 10	- 10	0	- 12	0	0	20

(De hele bewerking komt neer op het inverteren van de matrix, bestaande uit de in de basis opgenomen vectoren of activiteiten)

Dit eerste tussenprogramma bestaat uit
- het niet gebruiken van 60 éénheden land (P_5);
- het niet gebruiken van 100 éénheden arbeid in periode I (P_6);
- het telen van 40 éénheden sojabonen (P_3).

Het levert een saldo op, namelijk 1600 (regel Z, kolom P_0), namelijk 40 (niveau P_3) x 40 (saldo P_3).

Beschouwen we nu P_1, dan blijkt, dat Z hier een waarde heeft gekregen, namelijk 20. Het is immers zo dat P_1 hetzelfde beslag legt op de beperkingen als de lineaire combinatie 0,5 P_5 + 0 P_6 + 0,5 P_3. Deze lineaire combinatie levert een saldo van 0,5 x 0 + 0 x 0 + 0,5 x 40 (saldo P_3) op, ofwel 20. Dit zijn de "alternatieve kosten" voor P_1. P_1 moet immers bij ontplooiïng op niveau 1 meer dan 20 opbrengen; anders is het niet aantrekkelijk P_1 in het programma te brengen. P_1 levert echter een C op van 30. De opbrengst ervan is dus hoger dan de alternatieve kosten ervoor. Z-C = - 10, dus het is aantrekkelijk P_1 in het programma te brengen. Het is evenwel nog voordeliger P_4 in te voeren. De op dezelfde wijze als voor P_1 berekende alternatieve kosten zijn voor deze activiteit 0, waar een saldo van 12 tegenover staat.

Bij de tweede iteratietrap brengen we derhalve P_4 in de basis. Deze loopt vast op het land (P_5); meer dan 60 eenheden kunnen niet in het programma worden gebracht.

P_4 komt dus in de basis en P_5 gaat eruit. Op dezelfde wijze als bij de eerste iteratietrap gaan we nu alle activiteiten (P_0 inclusis) uitdrukken als een lineaire combinatie van P_4, P_6 en P_3. Het tableau, dat we dan krijgen is als volgt:

C		P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
12	P_4	60	0,5	1	0	1	1	0	- 0,5
0	P_6	70	- 0,25	0,5	0	0	- 0,5	1	0,25
40	P_3	40	0,5	0	1	0	0	0	0,5
	Z	2320	26	12	40	12	12	0	14
	Z-C	2320	- 4	2	0	0	12	0	14

Nu blijkt alleen P_1 nog een negatieve waarde Z-C te hebben. Het beslag van P_1 op de beperkingen is gelijk aan het beslag van 0,5 P_4 - 0,25 P_6 + + 0,5 P_3. Deze combinatie levert een saldo op van 26; P_1 echter een saldo van 30. Merkwaardig is dat P_3, die in de eerste iteratietrap in de basis werd opgenomen er nu weer uit verdwijnt ten gunste van P_1. P_1 loopt immers op P_3 vast; P_3 blijkt voor P_1 de effectieve beperking (het knelpunt) te zijn. Wat beperking P_4 betreft zou P_1 op niveau 120 kunnen worden opgenomen, terwijl P_6 in het geheel niet beperkend is. Het knelpunt P_3 maakt dat P_1 ten hoogste op niveau 80 kan worden ontplooid.

De derde iteratietrap levert op

C		P_0	P_1	P_2	P_3	P_4	P_5	P_6	P_7
12	P_4	20	0	1	- 1	1	1	0	-1
0	P_6	90	0	0,5	0,5	0	- 0,5	1	0,5
30	P_1	80	1	0	2	0	0	0	1
	Z	2640	30	12	48	12	12	0	18
	Z-C	2640	0	2	8	0	12	0	18

We hebben nu het optimale plan verkregen. Z-C is namelijk voor iedere activiteit positief, wat betekent, dat invoering van deze activiteiten het saldo zal verlagen. Men offert dan meer op (Z) dan men terugkrijgt (C).

Het optimale plan bevat

- teelt van 20 eenheden tarwe;
- "niet gebruiken" van 90 éénheden arbeid in periode I;
- teelt van 80 eenheden maïs

en levert een totaal saldo op van 2640.

106/056/1/1/66/67 gegeven bewerking bij de eerste iteratietrap kunnen de volgende <u>rekenregels</u> worden afgeleid:

1. Zoek de activiteit met de hoogste negatieve Z-C waarde,

2. Bepaal het knelpunt door de elementen van de P_0 vector te delen door de overeenkomstige (op dezelfde rij) van de gekozen (te ontplooien) (in de basis te brengen) activiteit. Het kleinste positieve quotient bepaalt het knelpunt. (N.B. de elementen van P_0 mogen nooit negatief worden en voorts een negatief getal in een activiteit verruimt de beperking).

3. Maak het schema voor het nieuwe tableau; de kop daarvan is gelijk aan die van het oude.Op de omschrijving van de regels verandert echter iets. De beperking, die het knelpunt vormde (laagste quotient zie 2), de zogenaamde"uitgaande rij"wordt vervangen door de"inkomende rij" dit is de"uitgaande kolom"of wel de activiteit met de hoogste negatieve Z-C waarde. In het nu komende tableau wordt deze activiteit een zogenaamde één-heidsvector (bestaande uit één 1 en voor de rest O'en; hij wordt namelijk in zichzelf uitgedrukt).

4. Deel de uitgaande rij (bij de eerste iteratietrap P_7 en bij de tweede en derde trappen respectievelijk P_5 en P_3) term voor term door het getal, dat staat op de kruising van de activiteit (kolom) met de hoogste negatieve Z-C waarde en de uitgaande rij. Dit is de voor de betrokken iteratietrap "algemene noemer". (Bij de eerste iteratietrap 2 en bij de 2de en 3de trap respectievelijk +1 en +0,5). Schrijf de uitkomsten in het nieuwe tableau op dezelfde regel (die nu een andere naam heeft) als in het oude tableau.

5. Om de andere rijen in het nieuwe tableau te berekenen (de rijen, die dezelfde aanduiding houden) trekken we van de rij in het oude tableau af de eerst behandelde rij uit het oude tableau (het knelpunt), vermenigvuldigd met een breuk, waarvan de <u>teller</u> bestaat uit de term op de kruising van de te behandelen rij met de kolom met de hoogste negatieve Z-C waarde (de kolom, die in de basis is gebracht) en de <u>noemer</u> uit de term op de kruising van de uitgaande rij (oude tableau) en de in de basis gebrachte kolomvector (hoogste Z-C waarde), de in 4 bedoelde "algemene noemer".

[N.B. Daar de eerste rij in het nieuwe tableau bestaat uit de overeen. komstige rij in het oude tableau in zijn geheel (term voor term) gedeeld door deze "algemene noemer" kan men een andere rij ook als volgt berekenen: "Trek van de oude rij af de eerst verkregen rij in het nieuwe tableau (dat is de oude rij gedeeld door de algemene noemer), vermenigvuldigd met het getal, dat voorkomt op de kruising van de te behandelen rij met de kolom met de hoogste negatieve Z-C waarde.

6. De Z en Z-C regels kunnen op dezelfde wijze worden berekend als de andere regels. Voor deze regels geldt derhalve ook het onder 5 behandelde rekenvoorschrift, waarbij we er echter wel rekening mee moeten houden, dat we om de Z regel te krijgen moeten vermenigvuldigen met het getal op de kruising van de oude Z-C regel en de activiteit met de laagste Z-C waarde. Voor de Z-C regel geldt het rekenvoorschrift ongewijzigd. Alleen Z-C is interessant, omdat deze regel bepaalt welke activiteit bij de volgende iteratie in de basis wordt gebracht. Het heeft weinig zin de Z regel nog verder te berekenen. We slaan deze regel dus in het algemeen over.