Optimering

Lineær programmering

Projektrapport 2. semester
Gruppe A400b

Aalborg Universitet
Basis - Matematik
Strandvejen 12-14
DK-9000 Aalborg

Aalborg Universitet - Basis
Matematik
Strandvejen 12-14
9000 Aalborg
http://math.aau.dk

Titel:

Optimering - Lineær programmering

Projekt:

P2-projekt

Projektperiode:

4. februar 2019 - 27. maj 2019

Projektgruppe:

A400b

Deltagere:

Yasmin Belal Abouarabi
Sofie Broch Christensen
Frederik Grove Hougaard
Tea Caroline Konnerup
Bastian Peitersen
Marcus Johan Schytt
Anders Lindgaard Sørensen

Vejledere:

Lisbeth Fajstrup
Bettina Dahl Søndergaard

Sidetal: 95 + forside
Afsluttet 27. maj 2019

Synopsis:

Projektet undersøger optimering gennem lineære programmeringsproblemer og deres løsninger. Der redegøres i kapitel 3 for omskrivningen af et virkelighedsnært problem til et lineært programmeringsproblem på standardform, og begreberne mulighedsmængde og mulig løsning defineres.

Derefter redegøres der i kapitel 4 for geometrien i lineære programmeringsproblemer, hvorfor teori om polyedre og konveksitet introduceres. Desuden defineres hjørnepunkter, kritiske punkter og basale mulige løsninger, som til slut i kapitlet kobles til optimalitet.

I kapitel 5 redegøres der for, hvordan den matematiske teori fører frem til Simplex-metoden som en systematisk løsningsmetode til lineære programmeringsproblemer. Her defineres mulige- og basale retninger, den reducerede omkostning samt optimale baser. Dette leder frem til en iteration i den naive Simplex-metode. Dette videreudvikles også i kapitel 5 til den Reviderede Simplex-metode og igen videre til Simplex på Tabelform. I kapitel 5 redegøres der også for To-Fase Simplex-metoden. Projektet afsluttes med kapitel 6 indeholdende eksempler på anvendelse af de forskellige implementeringer af Simplex.

Underskrifter

Yasmin Belal Abouarabe

Sofie Broch Christensen

Frederik Grove Hougaard

Tea Caroline Konnerup

Bastian Peitersen

Marcus Johan Schytt

Anders Lindgaard Sørensen

Forord

Dette projekt er skrevet af studerende i gruppe A400b på andet semesters Matematik ved det Ingeniør- og Naturvidenskabelige fakultet på Aalborg Universitet. Projektet er udarbejdet i foråret 2019.

Temaet for projektet er optimering og lineær programmering, samt matematiske modellering af - og løsning til - optimeringsproblemer. Formålet med rapporten er således at skabe kendskab til optimering af lineære programmeringsproblemer og Simplex-metoden som en systematisk løsningsmetode til lineære programmeringsproblemer.

Projektet er udarbejdet over et semester, hvor der er arbejdet med den bagvedliggende teori i forhold til en problemformulering. Dermed er projektet bygget op omkring problembaseret læring. Projektet er dog et teoretisk projekt og løser dermed et teoretisk problem frem for et virkelighedsnært problem. Der afsluttes dog med et kapitel indeholdende mindre eksempler på anvendelse af forskellige implementationer af Simplex-metoden.

Projektet er afgrænset af, at det henvender sig til førsteårsstuderende på matematik med grundlæggende kendskab til vektorregning og basal matrixregning fra lineær algebra, hvorfor der er afgrænset fra at redegøre for denne grundlæggende teori.

Projektet følger konventionerne for Vancouver-formatet til referering af kilder. Alle kilder nummereres med et tal [n] og kan altså findes i litteraturlisten under dette nummer. Sætninger, definitioner, korollarer, figurer og eksempler nummereres løbende efter kapitelnummeret. Efter forordet findes et symbolregister.

Slutteligt skal rettes en tak til henholdsvis vejleder og bivejleder for projektet Lisbeth Fajstrup og Bettina Dahl Søndergaard.

Symbolregister

A	Koefficientmatrix	$\vec{c}$	Omkostningsvektor
$\mathbb{R}^n$	n-dimensionelt vektorrum	x_j	Den j'te beslutningsvariabel
$\vec{A}_j$	Den j'te søjle i A	$I = 1, \ldots, m$	Indeksmængde
$\vec{e}_i$	Den i'te standardvektor	$J = 1, \ldots, n$	Indeksmængde
I_m	Identitetsmatrix $\in \mathbb{R}^m$	M_1	Delmængde af I $(\leq)$
$\vec{a}_i$	Den i'te rækkevektor i A	M_2	Delmængde af I $(\geq)$
$[A \mid \vec{b}\,]$	Totalmatrix	M_3	Delmængde af I $(=)$
A^{-1}	A's inverse	N_1	Delmængde af J $(\geq)$
$\mathrm{Span}(S)$	Span af vektorerne i mængden S	N_2	Delmængde af J $(\leq)$
W	Underrum	$\vec{x}$	Mulig løsning
$T_A : \mathbb{R}^n \to \mathbb{R}^m$	Matrixtransformation	s_i	Den i'te underskuds- eller overskudsvariabel
$\mathrm{Null}(A)$	Nulrummet af A	x_j^+, x_j^-	De j'te kunstige variable
$\mathrm{Col}(A)$	Søjlerummet af A	P	Polyeder
$\mathrm{Vm}(T_A)$	Værdimængde for T_A	H	Hyperplan

$\mathrm{Row}(A)$	Rækkerummet af A	H_l	Nedre halvrum
$\dim(W)$	Dimension af W	H_u	Øvre halvrum
$\mathrm{Rank}(A)$	Rangen af A	$B(1), \ldots, B(m)$	Basale indeks
$Z = \vec{c} \cdot \vec{x}$	Objektfunktion	$A_{B(1)}, \ldots, A_{B(m)}$	Basale søjler i A
$x_{B(1)}, \ldots, x_{B(m)}$	Basale variable	d_j	Den j'te komponent i $\vec{d}$
B	Basismatrix	$\tilde{c}_j$	Reducerede omkostning i den j'te retning
$\vec{x}_B$	Basisvektor	$\vec{c}_B$	Omkostningsvektor hørende til B
$\vec{d}$	Mulig retning fra $\vec{x}$	$\tilde{\vec{c}}$	Reducerede omkostningsvektor
$\tilde{B}$	Ny basismatrix	$\vec{u}$	$-\vec{d}_B$
$\vec{d}_B$	$-B^{-1}\vec{A}_j$	Q	Elementærmatrix

Indholdsfortegnelse

1 Problemanalyse

Forestil dig, at du er ejer af en virksomhed. En virksomhed, der producerer et tilfældigt produkt - møbler eksempelvis. En sådan virksomhed har ofte som mål i et eller andet omfang at tjene penge; at opnå profit. Gerne så meget profit, som det er muligt. Det kan også være, at dit ønske er, at produktionen skal foregå så bæredygtigt som muligt eller med så få omkostninger som muligt. Med andre ord: Du ønsker, at din virksomhed skal *optimere* produktionen og/eller outputtet. Et sådan problem kaldes i matematikken et *optimeringsproblem*, hvor det altså er målet enten at *minimere* eller *maksimere* et givet output; en objektfunktion. Og på baggrund af sådanne virkelighedsnære problematikker dannes der et grundlag for det, der kaldes *matematisk modellering*, hvor det virkelighedsnære problem - ofte langt mere kompliceret end dette - sættes i system og derfor kan løses rent matematisk.

Når man gerne vil optimere profitten eller minimere forbruget i for eksempel en produktion, er dette altid underlagt en række *begrænsninger*, der kan fortælle noget om de ressourcer, der er tilgængelige - det være sig mængden af træ eller antallet af arbejdstimer i din møbelproduktion. Men det er dog ofte ikke muligt at medtage alle *variable* for en given kompleks virkelighedsnær problemstilling. Matematisk modellering foregår således i grænsefeltet mellem matematisk teori og virkelighedens problematikker, og er på den måde i mange tilfælde en forenkling af virkeligheden. Der er altså i de fleste tilfælde tale om, at løsningen er en *approksimation* [1].

Hvilken metode der anvendes til løsningen af et optimeringsproblem afhænger af, hvordan objektfunktionen er opbygget, samt hvilke begrænsninger denne er underlagt. Nogle problemer, som eksempelvis produktion, vil ofte kunne modelleres med en *lineær* objektfunktion underlagt lineære begrænsninger. Men man kan også forestille sig problemer, der bedst modelleres med en ikke-lineær funktion underlagt ikke-lineære begrænsninger. Endvidere kan nogle problemer være afgrænset til kun at kunne løses med heltal - med andre ord, at du kun ønsker at producere hele møbler og ikke halve.

Lineær programmering

Problemer med en lineær objektfunktion af flere variable, underlagt lineære begrænsninger, kaldes for lineære optimeringsproblemer og er historisk blevet kaldt lineære programmeringsproblemer. En af de første til at arbejde med at løse sådanne problemer var Fourier, som i 1824 udviklede en algoritme til at løse lineære uligheder. Der skulle dog gå mange år, før denne problematik blev relevant. I slutningen af trediverne begyndte den russiske matematiker Kantorovich at arbejde med den optimale ressourcefordeling i en planøkonomi, hvor vi for første gang ser lineær programmering på den form, vi kender i dag [2]. Programmering blev også anvendt under 2. verdenskrig til blandt andet at beskrive processen til planlægning af operationer og ressourcetildeling. Der opstår altså konkrete problemer, som man ønsker løst, og i kølvandet på de tanker, der gøres i forbindelse med sådanne problemer, opstår der mere abstrakte idéer, der efterfølgende kan oversættes til en mere generel løsningsmodel. I 1940 fandt man således ud af, at programmering kunne anvendes til at løse andre specifikke optimeringsproblemer, givet med en lineær objektfunktion, underlagt lineære begrænsninger - deraf lineær programmering.

Senere samme årti, i 1947, opstillede George Dantzig *Simplex-metoden*, som netop er en generel løsning til lineære programmeringsproblemer. Denne blev siden den fundamentale teknik til løsning af lineære optimeringsproblemer [2]. Et tidligt eksempel på anvendelsen af Simplex-metoden var under blokaden af Berlin i 1948, hvor man ved hjælp af Simplex fik optimeret transporten af forsyninger til Berlin. I dag er Simplex-metoden et udbredt værktøj, der anvendes af mange forskellige fagfolk udover matematikere, da den er tilgængelig i pakkeprogrammer til diverse regneark [2].

Projektets formål

Dette projekt handler dermed om lineær programmering, matematisk modellering og løsningsmetoden Simplex. Dog ønsker vi ikke at løse en konkret virkelighedsnær problemstilling - som eksempelvis, hvordan du profitoptimerer din imaginære møbelvirksomhed. Det er derimod den underlægggende matematiske forståelse, der er det interessante for os.
Vi vender derfor tilbage til at arbejde med den teoretiske matematik, der ligger til grund for lineære problemstillinger. Vi ønsker med andre ord at redegøre for, hvordan man kommer fra et lineært modelleret matematisk problem til, *hvorfor* og *hvordan* løsningsmetoden Simplex fungerer til at løse denne type problemer *generelt*. Dette leder os frem til følgende problemformulering:

1.1 Problemformulering

Hvordan kan den bagvedliggende geometri i lineære programmeringsproblemer anvendes til at finde en algoritmisk løsningsmetode?

Vores problemformulering vil blive besvaret ud fra følgende fire problemstillinger:

- Hvad er et lineært programmeringsproblem?
- Hvordan kan et lineært programmeringsproblem betragtes geometrisk?
- Hvordan udledes Simplex-metoden fra den matematiske teori, og hvorfor virker metoden?
- Hvordan videreudvikler man Simplex-metoden til en algoritme?

1.2 Problemafgrænsning

Projektet her er et teoretisk projekt, som dermed henvender sig til det videnskabelige samfund. Hovedvægten vil derfor være på at redegøre for teorien, fremfor hvordan denne kan anvendes. Desuden skrives projektet med det formål, at andre førsteårs matematikstuderende med grundlæggende viden inden for lineær algebra skal kunne læse det, og skabe sig en forståelse af emnet. Projektet fokuserer derudover på det geometriske aspekt i lineær programmering. Derfor er projektets andet kapitel kun en opsummering af den teori, der anvendes indenfor lineær algebra, og de nødvendige resultater fra lineær algebra i kapitel 2 bevises derfor ikke i dette projekt. Der er dog ved alle sætninger henvist til kilder, hvor beviserne kan læses.

I projektet bearbejder vi udelukkende optimeringsproblemer stillet ved lineære funktioner underlagt lineære begrænsninger. Projektet omtaler derfor ikke ikke-lineære optimeringsproblemer og deres løsningsmetoder.

I projektet betragter vi alle reelle løsninger til et lineært programmeringsproblem, hvorfor heltalsprogrammering ikke omtales i projektet.

2 | Lineær algebra

I dette kapitel opsummeres nogle af de definitioner og resultater fra lineær algebra, der er relevante for indeværende projekt. Kapitlets resultater er afgørende for, at vi kan betragte et lineært optimeringsproblem geometrisk og derfra udvikle Simplex-metoden.

2.1 Linearkombinationer

Et vigtigt begreb i lineær algebra er *linearkombinationer*, som vi her definerer:

Definition 2.1 Linearkombinationer

Lad $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k \in \mathbb{R}^n$ og $t_1, t_2, \dots, t_k \in \mathbb{R}$. Vektoren givet på formen

$$t_1\vec{u}_1 + t_2\vec{u}_2 + \dots t_k\vec{u}_k$$

er en *linearkombination* af vektorerne $\vec{u}_1, \vec{u}_2, \dots, \vec{u}_k$, med *koefficienterne* $t_1, t_2, \dots, t_k$ [3, s. 14].

Vi husker, at skalarproduktet mellem to vektorer er en linearkombination af vektorernes komponenter. Hvis vi ønsker at beskrive en general linearkombination af nogle skalarer $u_1, \dots, u_k$, kan vi opstille dem i en vektor $\vec{u} = (u_1, \dots, u_k)^T$. Skalarproduktet mellem $\vec{u}$ og en vektor $\vec{t} \in \mathbb{R}^k$ vil da være givet ved

$$\vec{t} \cdot \vec{u} = \sum_{i=1}^{k} t_i u_i = t_1 u_1 + t_2 u_2 + \dots + t_k u_k.$$

Det bemærkes her, at vektoren $\vec{t}$ indeholder linearkombinationens koefficienter.

Ligeledes kan vi betragte en linearkombination af nogle vektorer $\vec{A}_1, \vec{A}_2, \dots, \vec{A}_n \in \mathbb{R}^m$, som er konstrueret ved at summere et multiplum af hver vektor. Vi opstiller her en $m \times n$-matrix A med vektorerne $\vec{A}_1, \vec{A}_2, \dots, \vec{A}_n$, som søjler:

$$A = [\vec{A}_1 \ \vec{A}_2 \ \cdots \ \vec{A}_n].$$

Vi kan udtrykke en linearkombination af søjlerne i matricen A ved et *matrix-vektor-produkt* [3, s. 19]:

Definition 2.2 Matrix-vektor-produkt

Lad A være en $m \times n$-matrix og $\vec{x}$ være en vektor i $\mathbb{R}^n$. Således er *matrix-vektor-produktet* $A\vec{x}$ en linearkombination af søjlerne i A, hvis koefficienter er komponenterne i $\vec{x}$:

$$A\vec{x} = x_1\vec{A}_1 + x_2\vec{A}_2 + \cdots + x_n\vec{A}_n \text{ [3, s. 19].}$$

Eksempel 2.3.
Givet en matrix A og en vektor $\vec{x}$

$$A = \begin{bmatrix} 2 & 7 & 0 \\ 1 & 3 & 4 \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} -1 \\ 2 \\ 5 \end{bmatrix},$$

kan vi udregne matrix-vektor-produktet $A\vec{x}$ på følgende måde:

$$A\vec{x} = \begin{bmatrix} 2 & 7 & 0 \\ 1 & 3 & 4 \end{bmatrix} \begin{bmatrix} -1 \\ 2 \\ 5 \end{bmatrix} = (-1)\begin{bmatrix} 2 \\ 1 \end{bmatrix} + 2\begin{bmatrix} 7 \\ 3 \end{bmatrix} + 5\begin{bmatrix} 0 \\ 4 \end{bmatrix} = \begin{bmatrix} (-1)\cdot 2 + 2\cdot 7 + 5\cdot 0 \\ (-1)\cdot 1 + 2\cdot 3 + 5\cdot 4 \end{bmatrix} = \begin{bmatrix} 12 \\ 25 \end{bmatrix}.$$

◀

Bemærk at antallet af søjler i matricen skal være det samme, som antallet af komponenter i vektoren $\vec{x}$ for, at et matrix-vektor-produkt er defineret. Resultatet bliver således en vektor i $\mathbb{R}^m$ og er på matrix-form da givet ved

$$A\vec{x} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}.$$

Matrix-vektor-produktet har en række egenskaber, som benævnes i følgende sætning:

Sætning 2.4

Lad A og D være $m \times n$-matricer, lad $\vec{u}$ og $\vec{v}$ være vektorer i $\mathbb{R}^n$, og lad c være en reel skalar. Da gælder følgende:

(a) $A(\vec{u} + \vec{v}) = A\vec{u} + A\vec{v}$.

(b) $A(c\vec{u}) = c(A\vec{u}) = (cA)\vec{u}$.

(c) $(A + D)\vec{u} = A\vec{u} + D\vec{u}$.

(d) $A\vec{e}_j = \vec{A}_j$, for $j = 1, 2, \ldots, n$, hvor $\vec{e}_j$ er den j'te standardvektor i $\mathbb{R}^n$ og $\vec{A}_j$ er den j'te søjlevektor i $\mathbb{R}^m$.

(e) Hvis $D\vec{u} = A\vec{u}$, så er $D = A$.

(f) $A\vec{0}$ er nulvektoren i $\mathbb{R}^m$.

(g) Hvis O er $m \times n$-nulmatricen, så er $O\vec{v}$ nulvektoren i $\mathbb{R}^m$.

(h) $I_n\vec{v} = \vec{v}$, hvor I_n er identitetsmatricen - se definition 2.28 [3, s. 24].

Beviset for punkt (a) i sætning 2.4 kan findes i [3, s. 24].

Matrix-vektor-produktet kan anvendes til at opstille en *matrix-vektor-ligning*. Antag at A er en $m \times n$-matrix, hvis søjler er $\vec{A}_1, \dots, \vec{A}_n$. Hvis vi ønsker at finde de koefficienter, som opfylder, at en linearkombination af A's søjler er lig en vektor $\vec{b} \in \mathbb{R}^m$, kan vi opstille følgende ligning:

$$A\vec{x} = \vec{b}.$$

Ved at løse den overstående ligning, vil vi finde en vektor $\vec{x}$, hvis indgange vil være koefficienterne i den ønskede linearkombination. Vi vil i afsnit 2.3 se på andre anvendelser af matrix-vektor-ligninger, samt betragte hvornår en sådan ligning har løsninger.

Vi vil nu afslutte dette afsnit med at betragte en specifik egenskab, som en mængde af vektorer kan have. Hvis ingen vektor i en mængde af vektorer kan udtrykkes som en linearkombination af de øvrige vektorer, kaldes mængden *lineært uafhængig*. Dette begreb definerer vi herunder:

Definition 2.5 Lineær uafhængighed

Mængden bestående af vektorerne $\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_k \in \mathbb{R}^n$ kaldes *lineært uafhængig*, hvis

$$t_1\vec{u}_1 + t_2\vec{u}_2 + \cdots + t_k\vec{u}_k = \vec{0},$$

kun er sandt, når $t_1, t_2, \dots, t_k = 0$. Hvis dette ikke er tilfældet kaldes mængden *lineært afhængig* [3, s. 75-76].

Bemærk at søjlerne i en $m \times n$-matrix A da vil være lineært uafhængige, hvis matrix-vektor-ligningen

$$A\vec{x} = \vec{0}$$

kun har løsningen $\vec{x} = \vec{0}$. Der vil altså være en entydig løsning til denne matrix-vektor-ligning.

2.2 Matrix-transformationer

Ethvert matrix-vektor-produkt bygger på konstruktionen af en linearkombination af nogle vektorer - opstillet som søjler i en $m \times n$-matrix A. En specifik linearkombination bygger på en vektor $\vec{x} \in \mathbb{R}^n$, hvis elementer er linearkombinationens koefficienter. Vi kan forestille os, at vi kan opstille en funktion, som tager denne vektor $\vec{x}$ som argument, og har den tilhørende linearkombination som funktionsværdi. Denne funktion er *matrix-transformationen* tilknyttet A.

Definition 2.6 Matrix-transformation

Lad A være en $m \times n$-matrix. Da er *matrix-transformationen* $T_A : \mathbb{R}^n \to \mathbb{R}^m$ defineret som

$$T_A(\vec{x}) = A\vec{x},$$

for $\vec{x} \in \mathbb{R}^n$ [3, s. 168].

Vi bemærker, at denne funktions værdimængde er alle mulige linearkombinationer af A's søjler. Mængden af alle mulige linearkombinationer af nogle vektorer, kaldes også disse vektorers *span*. Dette definerer vi herunder:

Definition 2.7 Span

Lad S være en ikke-tom mængde af vektorer $\vec{u}_1, \vec{u}_2, \ldots, \vec{u}_k \in \mathbb{R}^n$. Span($S$) er da mængden af alle linearkombinationer af $\vec{u}_1, \vec{u}_2, \cdots, \vec{u}_k \in \mathbb{R}^n$. Mængden Span($S$) kaldes *span* af vektorerne, hvor S er en udspændende mængde af Span(S) [3, s. 66].

Vi ser her, at værdimængden af en matrix-transformation af en $m \times n$-matrix A er span af A's søjler. Vi siger her, at A's søjler *udspænder* værdimængden.

Vi kan forestille os, at værdimængden af matrix-transformationen tilhørende A nødvendigvis ikke er hele vektorrummet $\mathbb{R}^m$. Vi siger da, at værdimængden af matrix-transformationen er et *underrum* af $\mathbb{R}^m$.

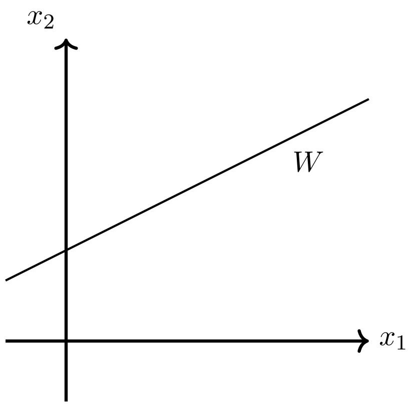
Definition 2.8 Underrum

En mængde $W \subseteq \mathbb{R}^n$ kaldes et *underrum* af $\mathbb{R}^n$, hvis den opfylder følgende kriterier:

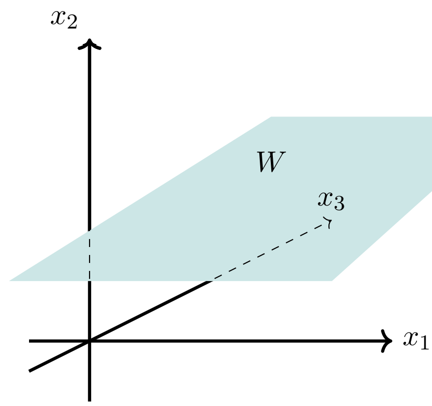
(a) $\vec{0} \in W$;

(b) $\vec{x}_1, \vec{x}_2 \in W \Rightarrow \vec{x}_1 + \vec{x}_2 \in W$;

(c) $\vec{x}_1 \in W, k \in \mathbb{R} \Rightarrow k\vec{x}_1 \in W$ [3, s. 227].



Figur 2.1: Linjen W er et underrum af $\mathbb{R}^2$.



Figur 2.2: Ligeledes er planen W et underrum af $\mathbb{R}^3$.

Bemærk at matrix-transformationens værdimængde opfylder de samme kriterier, som kriterierne for et underrum, da det er en *lineær funktion*:

Definition 2.9 Lineære funktioner

En funktion $f : \mathbb{R}^n \to \mathbb{R}^m$ siges at være en *lineær funktion*, hvis alle par af vektorer $\vec{x}, \vec{y} \in \mathbb{R}^n$, og alle skalarer $k \in \mathbb{R}$ opfylder *linearitetsbetingelserne*:

(a) $f(\vec{x} + \vec{y}) = f(\vec{x}) + f(\vec{y})$.

(b) $f(k\vec{x}) = k \cdot f(\vec{x})$ [3, s. 171].

Derudover gælder det, at matrix-transformationens værdimængde altid, som konsekvens af følgende resultat, vil være et underrum.

Sætning 2.10

En endelig delmængde af $\mathbb{R}^n$ udspænder et underrum af $\mathbb{R}^n$ [3, s. 231].

Et bevis for sætning 2.10 kan findes i [3, s. 231].

Da mængden af alle linearkombinationer af en matrix' søjler er et specifikt underrum tilhørende en specifik matrix, vælger vi herunder at definere dette:

Definition 2.11 Søjlerum

Lad A være en $m \times n$-matrix. Da udspænder søjle-vektorerne $\vec{A}_j$, $j = 1, \ldots, n$ mængden $\mathrm{Col}(A)$, som kaldes matricens *søjlerum*. Søjlerummet for en $m \times n$-matrix A kan angives eksplicit ved

$$\mathrm{Col}(A) = \{A\vec{x} \mid \vec{x} \in \mathbb{R}^n\} \quad [3, \text{ s. } 233].$$

Vi ser, at søjlerummet af A er ækvivalent med værdimængden af matrix-transformationen T_A:

$$\mathrm{Vm}(T_A) = \mathrm{Col}(A).$$

Vi kan se, at et muligt underrum af $\mathbb{R}^n$ kan være $\mathbb{R}^n$ selv. Dette vil være tilfældet for et søjlerum af en matrix, hvis søjler udspænder en mængde bestående af enhver linearkombination i det tilhørende vektorrum. Ligeledes kan et underrum af $\mathbb{R}^n$ også være en mængde kun bestående af nulvektoren. Dette er tilfældet for søjlerummet af en nul-matrix, da enhver linearkombination af dens søjler altid vil give nul-vektoren. Disse to eksempler på underrum kalder vi de *trivielle underrum* af $\mathbb{R}^n$ [3, s. 228].

Der findes også andre typer af underrum tilhørende en matrix. Et andet eksempel på et underrum af en matrix, vil være matricens *rækkerum*. Vi betragter her alle linearkombinationer af matricens rækker stillet som søjlevektorer:

Definition 2.12 Rækkerum

Lad A være en matrix. Da udspænder dens rækker $\vec{a}_i$, $i = 1, \dots m$ mængden $\text{Row}(A)$, som kaldes matricens *rækkerum* [3, s. 236].

Endnu et eksempel på et underrum tilhørende en matrix er dens *nulrum*:

Definition 2.13 Nulrum

Lad A være en $m \times n$-matrix. Da kaldes løsningsmængden til $A\vec{x} = \vec{0}$ for matricens *nulrum* og angives med $\text{Null}(A)$. Nulrummet af en $m \times n$-matrix A kan implicit gives

$$\text{Null}(A) = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{0}\} \quad [3, \text{ s. } 232].$$

Nulrummet til en matrix A består da af alle de vektorer, som afbildes over i nulvektoren af matrix-transformationen T_A. Bemærk at hvis nulrummet af A kun indeholder nul-vektoren, da er A's søjler lineært uafhængige. Omvendt, hvis nulrummet af A består af flere vektorer, da vil A's søjler være lineært afhængige. Det følgende resultat bekræfter at ethvert nulrum er et underrum:

Sætning 2.14

Lad A være en $m \times n$-matrix. Da er $\text{Null}(A)$ et underrum af $\mathbb{R}^n$ [3, s. 233].

Beviset for sætning 2.14 kan findes i [3, s. 233].

Vi så, at søjlerummet tilhørende en matrix var intuitivt, da det er lig værdimængden af matricens tilhørende matrix-transformation. Det samme er ikke tilfældet for de to øvrige underrum. Vi vil dog, i de næste kapitler, se dem anvendt til at beskrive geometrien bag et lineært programmeringsproblem.

2.2.1 Basis og dimension

Vi kan opstille en vektor i et underrum som en linearkombination af vektorerne i en udspændende mængde. Vi kan forestille os, at det ville være smart at bruge en udspændende mængde S, der har så *få* vektorer som muligt.

Definition 2.15 Baser for underrum

Lad W være et underrum af $\mathbb{R}^n$ forskellig fra $\{\vec{0}\}$. Vektorerne $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ danner en *basis* for W, hvis

(a) $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ er lineært uafhængige,

(b) $\vec{v}_1, \vec{v}_2, \dots, \vec{v}_n$ udspænder W [3, s. 241].

Vi ved, at søjlerne i en matrix danner en udspændende mængde for søjlerummet. Hvis disse søjler er lineært uafhængige, vil det medføre, at de er en basis for matricens søjlerum.

Vi kan forestille os, at vi kan støde på matricer, som har lineært afhængige søjler. Ud fra en sådan matrix kan vi konstruere en ny matrix bestående af lineært uafhængige søjler. For at opnå lineært uafhængige søjler, skal vi naturligvis fjerne de lineært afhængige søjler. Dette viser sig at være en mulig metode til at konstruere en basis.

Sætning 2.16

Lad S være en endelig udspændende mængde for et underrum W af $\mathbb{R}^n$ forskellig fra $\{\vec{0}\}$. Så kan S *udtyndes* til en basis for W ved at fjerne de lineært afhængige vektorer fra S [3, s. 243].

Et bevis for sætning 2.16 kan ses i [3, s. 243].

Modsat kan vi også støde på matricer, hvis søjler udspænder et ikke-trivielt underrum. Vi kan ud fra en sådan matrix konstruere en ny matrix, hvis søjler udspænder hele vektorrummet. Vi kan altså udvide til en basis:

Sætning 2.17

Lad S være en lineært uafhængig delmængde af et underrum W af $\mathbb{R}^n$ forskellig fra $\{\vec{0}\}$. Så kan S *udvides* til en basis for W ved at supplere med flere lineært uafhængige vektorer [3, s. 245].

Et bevis for sætning 2.17 kan ses i [3, s. 245].

Som konsekvens af dette må alle underrum have en basis. Vi kan forestille os, at vi kan udvide en mængde til en basis ved uendeligt mange forskellige vektorer. Der vil da eksistere uendeligt mange baser for det samme underrum. Disse baser vil dog have det til fælles, at de består af det samme antal af vektorer. Resultatet er givet ved følgende sætning:

Sætning 2.18

Lad W være et underrum af $\mathbb{R}^n$ forskellig fra $\{\vec{0}\}$. Ethvert par af baser for W indeholder det samme antal af vektorer [3, s. 245].

Et bevis for sætning 2.18 kan ses i [3, s. 245].

Da antallet af vektorer i enhver basis for et underrum er ens, kan vi tilskrive ethvert underum en *dimension*:

Definition 2.19 Dimension

Antallet af vektorer i en basis for et underrum W af $\mathbb{R}^n$ forskelligt $\{\vec{0}\}$, kaldes *dimensionen* af W og skrives $\dim(W)$.
Dimensionen af det trivielle underrum bestående af nulvektoren i $\mathbb{R}^n$ er 0 [3, s. 246].

Ligeledes vil det trivielle underrum $\mathbb{R}^n$ have dimensionen n.

Vi afslutter dette afsnit med at definere, hvad det vil sige at betragte *rangen* og *nulliteten* af en matrix - samt et resultatet heraf.

Definition 2.20 Rang

Rangen af en matrix A er dimensionen af dens søjlerum. Dette skrives $\text{rank}(A)$ [3, s. 48].

Rangen af en matrix er da givet ved

$$\text{rank}(A) = \dim(\text{Col}(A)).$$

Vi bemærker, at antallet af lineært uafhængige søjler i en matrix A vil give $\text{rank}(A)$. Desuden gælder det, at $\text{rank}(A)$ er lig $\text{rank}(A^T)$. Dette betyder, at antallet af uafhængige søjler er lig antallet af uafhængige rækker [3, s. 47].

Definition 2.21 Nullitet

Nulliteten af en matrix A er dimensionen af dens nulrum. Dette skrives $\text{nullitet}(A)$ [3, s. 48].

Nullitet af en matrix er da givet ved

$$\text{nullitet}(A) = \dim(\text{Null}(A)).$$

Sætning 2.22

Lad A være en $m \times n$-matrix, da gælder

$$\text{rank}(A) + \text{nullitet}(A) = n \quad [3, \text{s. } 47].$$

Ud fra dette resultat kan vi da tilskrive dimensionen af nulrummet, som differencen mellem n og antallet af uafhængige søjler i A:

$$\text{nullitet}(A) = n - \text{rank}(A).$$

2.3 Lineære ligningssystemer

De førnævnte matrix-vektor-ligninger kan også bruges til at beskrive et *lineært ligningsystem* bestående af flere lineære ligninger. Vi husker, at en *lineær ligning* med variable $x_1, x_2, \dots, x_n$ skrives på formen

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = b,$$

hvor skalarene $a_1, a_2, \dots a_n$ er ligningens koefficienter, og b er ligningens *konstant* [3, s. 27]. Et lineært ligningssystem er en samling af lineære ligninger, hvori de samme variable indgår:

Definition 2.23 Lineære ligningssystemer

Et *lineært ligningssystem* bestående af m lineære ligninger med n variable, skrives på formen

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n &= b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n &= b_2\\
&\vdots\\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n &= b_m,
\end{aligned}$$

hvor a_{ij} betegner koefficienten til x_j i ligning i, for $i = 1, 2, \ldots, m$ og $j = 1, 2, \ldots, n$ [3, s. 27-28].

Det er muligt at udtrykke et lineært ligningssystem som en *matrix-vektor-ligning*. Vi kan konstruere en $m \times n$-matrix A - bestående af koefficienterne i systemets lineære ligninger, en vektor $\vec{x}$ - bestående af systemets variable, og en vektor $\vec{b}$ - bestående af de lineære ligningers konstanter:

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n}\\ a_{21} & a_{22} & \cdots & a_{2n}\\ \vdots & \vdots & \ddots & \vdots\\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix}, \quad \vec{x} = \begin{bmatrix} x_1\\ x_2\\ \vdots\\ x_n \end{bmatrix} \quad og \quad \vec{b} = \begin{bmatrix} b_1\\ b_2\\ \vdots\\ b_m \end{bmatrix}.$$

Ligningssystemets i'te ligning vil da være

$$\vec{a}_i \cdot \vec{x} = b_i,$$

hvor $\vec{a}_i$'s komponenter svarer til A's indgange i den i'te række og b_i den i'te konstant. Matricen A kaldes for ligningssystemets *koefficientmatrix*. Ligeledes kan vi opstille en *totalmatrix*, som indeholder alle ligningssystemets koefficienter og konstanter:

$$[A \mid \vec{b}\,] = \left[\begin{array}{cccc|c} a_{11} & a_{12} & \cdots & a_{1n} & b_1\\ a_{21} & a_{22} & \cdots & a_{2n} & b_2\\ \vdots & \vdots & \ddots & \vdots & \vdots\\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{array}\right].$$

Vi vil senere se, at et ligningssystems totalmatrix giver tilstrækkelige oplysninger for, at vi kan løse ligningssystemet. Her følger et eksempel på et ligningssystem, der opstilles som henholdsvis en koefficientmatrix og en totalmatrix.

Eksempel 2.24.
Givet et lineært ligningssystem med ligningerne

$$\begin{aligned}
L_1 :&\quad x_1 - x_2 - x_3 = 3\\
L_2 :&\quad x_1 - 6x_2 - 5x_3 = 3\\
L_3 :&\quad 2x_1 - x_2 = 0,
\end{aligned}$$

opstilles koefficientmatricen og totalmatricen. For opstilling af koefficientmatricen aflæser vi koefficienterne foran variablene i ligningssystemet. I L_1 aflæser vi koefficienterne til at være 1, -1 og -1. I L_2 er koefficienter 1, -6 og -5 og i L_3 henholdsvis 2, -1 og 0, da x_3 ikke er repræsenteret. Dermed får vi koefficientmatricen

$$A = \begin{bmatrix} 1 & -1 & -1 \\ 1 & -6 & -5 \\ 2 & -1 & 0 \end{bmatrix}.$$

Konstanterne b udgør ligningernes højreside. Da får vi totalmatricen

$$[A \mid \vec{b}\,] = \left[\begin{array}{ccc|c} 1 & -1 & -1 & 3 \\ 1 & -6 & -5 & 3 \\ 2 & -1 & 0 & 0 \end{array}\right].$$

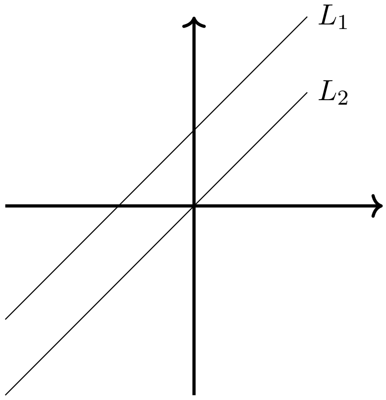
◀

Vi så tidligere på matrix-vektor-ligningen $A\vec{x} = \vec{0}$, hvor A er en matrix med lineært uafhængige søjler. I det tilfælde så vi fra definition 2.5, at $\vec{x} = \vec{0}$ var en entydig løsning. Vi kan opstille samme matrix-vektor-ligning som et lineært ligningssystem. Dette lineære ligningssystem vil ligeledes have en entydig løsning. Vi vil nu præsentere et resultat, som viser, at det ikke altid er tilfældet, at et lineært ligningssystem har præcis én løsning.

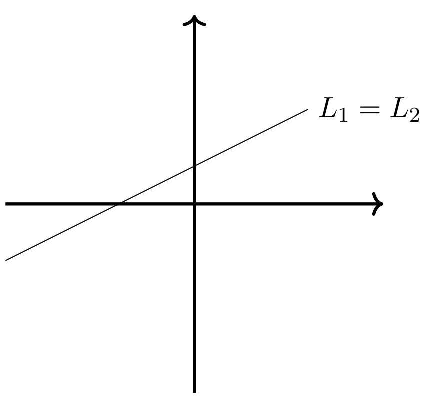
Sætning 2.25

Et lineært ligningssystem har enten ingen løsning, præcis én løsning eller uendelig mange løsninger [3, s. 29].

De tre mulige udfald kan vi med fordel betragte geometrisk. Løsningen til en lineær ligning med to variable kan repræsenteres ved en linje i $\mathbb{R}^2$. To forskellige lineære ligninger med to variable vil da repræsentere to linjer i $\mathbb{R}^2$. De punkter i planen, hvor de to linjer mødes, vil være der, hvor en løsning til begge linjer findes.

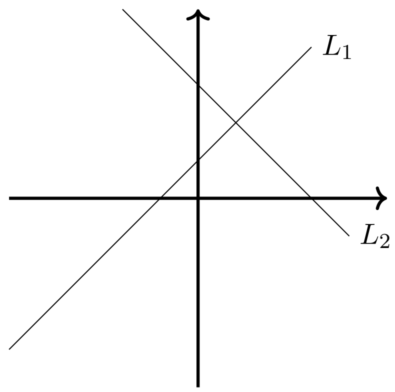


Figur 2.3: L_1 og L_2 er parallelle. Ingen løsning.



Figur 2.4: L_1 og L_2 er ens. Uendelig mange løsninger.

Betragt figur 2.3. Vi ser her to parallelle linjer, som viser løsningsmængden til to lineære ligninger. Et kendt resultat er, at to parallelle linjer i planen aldrig vil krydse. Af den grund vil ligningssystemet, bestående af disse to ligninger, ikke have nogen løsning. Betragt nu figur 2.4. Det ses, at løsningsmængden til begge ligninger her er ens. Vi ser, at ligningssystemets løsninger er alle punkter på linjen. Der vil da være uendelig mange løsninger til ligningssystemet.



Figur 2.5: L_1 og L_2 er ikke parallelle.
Præcis én løsning.

Betragt nu det tredje og sidste tilfælde i figur 2.5. Vi ser her, at løsningsmængderne til hver ligning krydser i ét entydigt punkt. Ligningssystemet bestående af disse to ligninger vil da have én entydig løsning [3, s. 29].

Bemærk at hvis løsningsmængden til to ligninger er ens, så må de to ligninger også være ens. Dette er tilfældet, hvis den ene lignings koefficienter og konstant er en linearkombination af koefficienterne og konstanten i den anden ligning. Dette vil afspejle sig i ligningssystemets totalmatrix, hvor de to rækker vil være lineært afhængige.

Da vi fra afsnit 2.2 ved, at antallet af lineært uafhængige rækker i en matrix er lig antallet af lineært uafhængige søjler, kan vi bruge denne viden til at bestemme antallet af løsninger til et lineært ligningssystem. Vi har lige set, at hvis A's søjler er lineært afhængige, da vil der være uendeligt mange løsninger. Vi har i definition 2.5 også set, at hvis A har lineært uafhængige søjler, så er løsningen entydig.

Et lineært ligningssystem kan, ved et lille antal variable, nemt løses ved at isolere variable og substituere dem i de andre ligninger - men ved et større antal variable kan det blive mere kompliceret. Vi ønsker derfor en metode, der gør dette nemmere. En sådan metode vil vi senere introducere i afsnit 2.5.

2.4 Matrix-multiplikation

For at kunne konstruere den førnævnte metode, skal vi først have kendskab til *matrix-multiplikation* og dets konsekvenser. Det at *gange* matricer sammen definerer vi herunder:

Definition 2.26 Matrix-multiplikation

Lad A være en $m \times n$-matrix og D en $n \times k$-matrix. Da er deres produkt AD en $m \times k$-matrix, hvis indgange er givet ved

$$ad_{ij} = \sum_{l=1}^{n} a_{il} d_{lj} = \vec{a}_i \cdot \vec{D}_j$$

hvor $\vec{a}_i$, for $i = 1, \ldots, m$, er den i'te række i A og $\vec{D}_j$, $j = 1, \ldots, k$, er den j'te søjle i D [2, s. 28].

Matrix-multiplikation er associativt så $(AD)C = A(DC)$. Til gengæld er det ikke altid kommutativt. Det vil sige, at ligningen $AD = DA$ ikke altid er gældende [2, s. 28].

Eksempel 2.27.
Givet matricerne A og D, hvor

$$A = \begin{bmatrix} \mathbf{2} & \mathbf{3} & \mathbf{1} \\ 4 & 0 & 1 \\ 0 & 2 & 6 \end{bmatrix} \quad \text{og} \quad D = \begin{bmatrix} \mathbf{5} & 1 & 3 \\ \mathbf{1} & 6 & 2 \\ \mathbf{1} & 0 & 4 \end{bmatrix}$$

vil deres produkt AD være givet ved

$$\begin{aligned} AD &= \begin{bmatrix} \mathbf{2 \cdot 5 + 3 \cdot 1 + 1 \cdot 1} & 2 \cdot 1 + 3 \cdot 6 + 1 \cdot 0 & 2 \cdot 3 + 3 \cdot 2 + 1 \cdot 4 \\ 4 \cdot 5 + 0 \cdot 1 + 1 \cdot 1 & 4 \cdot 1 + 0 \cdot 6 + 1 \cdot 0 & 4 \cdot 3 + 0 \cdot 2 + 1 \cdot 4 \\ 0 \cdot 5 + 2 \cdot 1 + 6 \cdot 1 & 0 \cdot 1 + 2 \cdot 6 + 6 \cdot 0 & 0 \cdot 3 + 2 \cdot 2 + 6 \cdot 4 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{14} & 20 & 16 \\ 21 & 4 & 16 \\ 8 & 12 & 28 \end{bmatrix}. \end{aligned}$$

Her er række $\vec{a}_1$ i A, søjle $\vec{D}_1$ i D, samt indgang ad_{11} i produktet AD fremhævet for at tydeliggøre fremgangsmåden i udregningen. ◀

Vi kan ud fra definitionen af matrix-multiplikation bemærke, at vi også kan beregne et matrix-produkt én søjle ad gangen - i stedet for én indgang ad gangen. Lad igen A være en $m \times n$-matrix og D en $n \times k$-matrix. Matrix-produktet AD kan da betragtes

$$AD = \begin{bmatrix} A\vec{D}_1 & A\vec{D}_2 & \cdots & A\vec{D}_k \end{bmatrix}.$$

Vi vil nu betragte en *identitetsmatrix*, der har nogle egenskaber, vi senere vil gøre brug af. Denne definerer vi herunder:

Definition 2.28 Identitetsmatricer

Den kvadratiske $n \times n$-matrix, som har skalaren 1 i dens diagonale indgange og skalaren 0 i de resterende, kaldes *identitetsmatricen* I_n [3, s. 22].

Bemærk at identitetsmatricen I_n har standardvektorerne $\vec{e}_i$, for $i = 1, \ldots, n$, som søjler [3, s. 22]. Identitetsmatricen I_3 ser da ud som følgende:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Vi kan nu betragte identitetsmatricens egenskab ved at udregne produktet mellem en vilkårlig $m \times n$-matrix A og identitetsmatricen I_n:

$$AI_n = \begin{bmatrix} A\vec{e}_1 & A\vec{e}_2 & \cdots & A\vec{e}_n \end{bmatrix} = \begin{bmatrix} \vec{A}_1 & \vec{A}_2 & \cdots & \vec{A}_n \end{bmatrix} = A.$$

Bemærk at produktet mellem identitetsmatricen I_m og A har samme egenskab:

$$I_mA = \begin{bmatrix} I_m\vec{A}_1 & I_m\vec{A}_2 & \cdots & I_m\vec{A}_n \end{bmatrix} = \begin{bmatrix} \vec{A}_1 & \vec{A}_2 & \cdots & \vec{A}_n \end{bmatrix} = A.$$

Vi kan nu bruge identitetsmatricen til at definere den *inverse matrix* til en matrix:

Definition 2.29 Inverse matricer

Lad A og D være kvadratiske $m \times m$-matricer og lad $AD = DA = I_m$, hvor I_m er identitetsmatricen. A er da invertibel og D er A's inverse. Dette noteres som $D = A^{-1}$ [3, s. 122].

Vi kan anvende den inverse matrix til at løse specifikke ligningssystemer. Lad A være en invertibel $m \times m$-matrix og $\vec{b}$ være en vektor i $\mathbb{R}^m$. Betragt nu ligningssystemet givet ved A og $\vec{b}$:

$$\begin{aligned} A\vec{x} &= \vec{b} \Leftrightarrow \\ A^{-1}A\vec{x} &= A^{-1}\vec{b} \Leftrightarrow \\ I_m\vec{x} &= A^{-1}\vec{b} \Leftrightarrow \\ \vec{x} &= A^{-1}\vec{b}. \end{aligned}$$

Vi kan heraf udlede, at ligningssystemer, givet ved invertible matricer, har entydige løsninger, der nemt kan findes ved at udregne matrix-vektor-produktet mellem systemets inverse matrix og løsningen. Husk at vi tidligere fandt ud af, at en entydig løsning vil føre til, at ligningssystemts matrix har lineært uafhængige søjler. Dette tilfælde og mere opsummerer vi i det følgende stærkere resultat:

Sætning 2.30

Lad A være en $m \times m$-matrix. Følgende er da ækvivalente:

(a) A er invertibel;

(b) Ligningssystemet $A\vec{x} = \vec{b}$ har en entydig løsning $\vec{x}$ for alle $\vec{b} \in \mathbb{R}^m$;

(c) A har lineært uafhængige søjler;

(d) A's søjler er en basis for $\mathbb{R}^m$;

(e) A's søjler udspænder $\mathbb{R}^m$;

(f) $\text{Rank}(A) = m$;

(g) $\text{Null}(A) = \{\vec{0}\}$ [3, s. 138].

Et bevis for sætning 2.30 kan findes i [3, s. 138-139].

2.5 Elementære rækkeoperationer

Vi vender nu tilbage til at beskrive den førnævnte metode til at løse lineære ligningsystemer, ved at manipulere med dets totalmatrix. For et lineært ligningssystem, givet som en totalmatrix, kan vi foretage tre forskellige operationer. Disse kaldes *elementære rækkeoperationer* [3, s. 31].

Definition 2.31 Elementære rækkeoperationer

De følgende tre operationer udført på en matrix, kaldes *elementære rækkeoperationer*.

(a) **Rækkeombytning** ($\vec{r_i} \leftrightarrow \vec{r_j}$): Der foretages en ombytning af den j'te og den i'te række i matricen.

(b) **Skalarmultiplikation** ($c\vec{r_i} \rightarrow \vec{r_i}$): Hver indgang i den i'te række af matricen multipliceres med den samme skalar $c \neq 0$.

(c) **Rækkeudskiftning** ($c\vec{r_i} + \vec{r_j} \rightarrow \vec{r_j}$): Der adderes et multiplum af den j'te række i matricen til en den i'te række [3, s. 32].

Vektoren $\vec{r_i}$ betegner den i'te række.

Foretager vi elementære rækkeoperationer på en matrix A, dannes der en ny matrix D. De typer af rækkeoperationer, der bruges på en matrix A, til at danne den nye matrix D, kan omvendt også benyttes på D for igen at opnå A. Bemærk at rækkeombytningen tydeligvis ikke har nogen indflydelse på ligningssystemets løsninger, da vi blot vælger at opskrive dets ligninger i en anden rækkefølge. Ligeledes vil de resterende rækkeoperationer heller ikke ændre på ligningssystemets løsninger [3, s. 33].

Ved brug af elementære rækkeoperationer kan der opstå nogle særlige rækker, som vi kalder *nul-rækker* med skalaren 0 i alle indgange. Modsat er *ikke-nul-rækker* de rækker, hvor mindst én indgang indeholder en skalar $k \neq 0$. En *ledende indgang* er den første indgang fra venstre i en række med en skalar $k \neq 0$, hvis og kun hvis skalarene i indgangene under er lig nul. En ledende indgang kaldes også for en *pivot-indgang*. Hvis en søjle eller række indeholder en pivot-indgang, kaldes disse henholdsvis *pivot-søjle* og *pivot-række* [3, s. 41]. De introducerede begreber vil i nedenstående eksempel 2.32 blive anskueliggjort.

Eksempel 2.32.
Vi får givet en 3×3-matrix

$$A = \begin{bmatrix} 1 & 3 & 1 \\ 0 & 5 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

I matrix A er pivot-indgangene markeret med en cirkel. Pivot-søjlerne er dermed søjle 1 og søjle 2 og pivot-rækkerne er række 1 og række 2. Desuden er række 1 og række 2 også ikke-nul-rækker, mens række 3 er en nul-række. ◀

2.5.1 Løsninger på lineære ligningssystemer

For at løse et lineært ligningssystem bruges et antal elementære rækkeoperationer på en totalmatrix A, sådan at vi når frem til en totalmatrix D på såkaldt *trappeform* og dernæst på *reduceret trappeform*. Disse to matrixformer defineres her:

Definition 2.33 Trappeform og reduceret trappeform

En matrix siges at være på *trappeform*, hvis den opfylder følgende tre betingelser:

(a) Hver ikke-nul-række ligger over enhver nul-række.

(b) Pivot-indgangen i enhver pivot-række ligger til højre for pivot-indgangen i enhver forrige række.

(c) Alle indgange under pivot-indgangen i pivot-søjlen er 0.

Hvis en matrix yderligere opfylder følgende to betingelser, siges den at være på *reduceret trappeform*.

(d) Alle indgange forskellig fra pivot-indgangen i pivot-søjlen er 0.

(e) Pivot-indgangen er 1 [3, s. 33].

På trappeform kan *antallet* af løsninger aflæses ved at betragte antallet af pivot-søjler i totalmatricen. På reduceret trappeform kan *løsningen* aflæses. Dette forklares herunder.

Vi så i sætning 2.25, at der enten er én, ingen eller uendelig mange løsninger til et lineært ligningsystem. Dette kobler vi nu til begreberne, der blev anskueliggjort i eksempel 2.32.
Lad D være en $m \times n + 1$-totalmatrix på trappeform. Da gælder følgende:

- Hvis totalmatricen D har pivot i sidste søjle, har ligningsystemet *ingen* løsninger og siges at være *inkonsistent*.

- Hvis totalmatricen D *ikke* har pivot i sidste søjle og har n pivot-indgange i alt, har ligningsystemet en *entydig* løsning og siges at være *konsistent*.
- Hvis totalmatricen D *ikke* har pivot i sidste søjle og har færre end n pivot-søjler, har ligningssystemet uendeligt mange løsninger og er konsistent [3, s. 34-37].

De variable, der ikke tilhører en pivot-søjle, kaldes *frie variable* og de variable, der tilhører en pivot-søjle, kan udtrykkes ved de frie variable. I det tilfælde, hvor der er uendelig mange løsninger, idet der er frie variable i ligningssystemet, kan en *generel løsning* opstilles [3, s. 35]. Den generelle løsning kan skrives på vektor-form som en parameterfremstilling. For at illustrere hvordan vi udfører dette, gennemfører vi her et eksempel.

Eksempel 2.34.
Givet et ligningssystem

$$\begin{aligned} x_1 + 2x_2 - x_3 + 2x_4 + x_5 &= 2 \\ -x_1 - 2x_2 + x_3 + 2x_4 + 3x_5 &= 8 \\ 2x_1 + 4x_2 - 3x_3 + 2x_4 &= 3 \\ -3x_1 - 6x_2 + 2x_3 + 3x_5 &= 9, \end{aligned}$$

opstiller vi totalmatricen, hvorpå vi gennemfører en følge af elementære rækkeoperationer for at nå frem til en matrix på reduceret trappeform:

$$\begin{bmatrix} 1 & 2 & -1 & 2 & 1 & 2 \\ -1 & -2 & 1 & 2 & 3 & 8 \\ 2 & 4 & -3 & 2 & 0 & 3 \\ -3 & -6 & 2 & 0 & 3 & 9 \end{bmatrix} \sim \cdots \sim \begin{bmatrix} 1 & 2 & 0 & 0 & -1 & -5 \\ 0 & 0 & 1 & 0 & 0 & -3 \\ 0 & 0 & 0 & 1 & 1 & 2 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

Vi kan ud fra vores totalmatrix på reduceret trappeform se, at der ikke er pivot i sidste søjle, samt at der er flere variable end pivot-søjler og dermed frie variable. Ligningssystemet er derfor konsistent og har uendeligt mange løsninger. Vi opstiller nu en generel løsning, hvor den reducerede matrix kan skrives som følgende ligningssystem:

$$\begin{aligned} x_1 + 2x_2 - x_5 &= -5 \\ x_3 &= -3 \\ x_4 + x_5 &= 2. \end{aligned}$$

Vi ser ud fra vores reducerede matrix, at x_2 og x_5 er frie variable, da de ikke tilhører pivot-søjler. Vi omskriver derfor den generelle løsning til

$$\begin{aligned} & x_1 = -2x_2 + x_5 - 5 \\ & x_2 \ \ fri \\ & x_3 = -3 \\ & x_4 = -x_5 + 2 \\ & x_5 \ \ fri. \end{aligned}$$

Vi har derved fundet frem til den generelle løsning for vores oprindelige ligningssystem, som vi kan opskrive på vektor-form

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} -2x_2 + x_5 - 5 \\ x_2 \\ -3 \\ -x_5 + 2 \\ x_5 \end{bmatrix} = x_2 \begin{bmatrix} -2 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + x_5 \begin{bmatrix} 1 \\ 0 \\ 0 \\ -1 \\ 1 \end{bmatrix} + \begin{bmatrix} -5 \\ 0 \\ -3 \\ 2 \\ 0 \end{bmatrix}.$$

◀

2.5.2 Elementærmatricer

Vi vil nu se, at vi kan bruge matrix-multiplikation til at udføre elementære rækkeoperationer på en vilkårlig matrix. Til dette skal vi først definere *elementærmatricer*:

Definition 2.35 Elementærmatricer

En kvadratisk $m \times m$-matrix Q, som adskiller sig fra identitsmatricen I_m, med én elementær rækkeoperation, kaldes en *elementærmatrix* [3, s. 126].

Det viser sig, at hvis vi udfører en elementær rækkeoperation på en identitetsmatrix I_m, og ganger den på en vilkårlig $m \times n$-matrix A fra venstre - svarer det til at udføre samme elementære rækkeoperation på A. Vi vil nu give et eksempel på denne egenskab:

Eksempel 2.36.
Betragt følgende 3×3-matrix

$$A = \begin{bmatrix} 3 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Antag at vi ønsker at udføre rækkeudskiftningen $\vec{r}_1 + 3\vec{r}_3 \to \vec{r}_1$ på A. Lad os i stedet udføre denne rækkeoperation på identitetsmatricen I_3 og angive den tilhørende elementærmatrix Q:

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 3 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = Q.$$

Lad os nu betragte produktet mellem elementærmatricen Q og A,

$$QA = \begin{bmatrix} Q \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} & Q \begin{bmatrix} 2 \\ 0 \\ 0 \end{bmatrix} & Q \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} \end{bmatrix} = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}.$$

Bemærk at den ønskede elementære rækkeoperation, som vi udførte på I_3, nu, efter multiplikationen, også er udført på A. ◀

Hvis vi ønsker at udføre flere rækkeoperationer på en vilkårlig matrix ved matrix-multiplikation, kan dette gøres ved at gange forskellige elementærmatricer på fra venstre. En følge af k rækkeoperationer udført på en vilkårlig matrix A, med tilhørende elementærmatricer $Q_1, \ldots Q_k$, vil da også kunne gives som følgende produkt:

$$Q_k Q_{k-1} \ldots Q_2 Q_1 A.$$

For at gøre notationen enkel lader vi en følge af rækkeoperationer, givet som et produkt af elementærmatricer, være lig en matrix Q. Det overstående produkt vil da være ækvivalent med QA.

Bemærk at hvis vi ved hjælp af elementære rækkeoperationer kan gå fra en kvadratisk $m \times m$-matrix A til identitetsmatricen I_m, vil det tilsvarende produkt af elementærmatricer Q opfylde

$$QA = I_m.$$

Det kan verificeres, at dette betyder at $Q = A^{-1}$.

Elementære rækkeoperationer kan yderligere bruges til at finde den inverse matrix til en invertibel $m \times m$-matrix A. Vi starter da med at opstille en matrix, som indeholder A og identitetsmatricen I_m. Dette skrives som $[A \mid I_m\,]$. Vi vil derefter benytte elementære rækkeoperationer for at opnå $[I_m \mid A^{-1}\,]$ [3, s. 135-136]. Vi vil herunder gennemgå et eksempel:

Eksempel 2.37.
Vi ønsker at finde den inverse til følgende invertible 3×3-matrix:

$$A = \begin{bmatrix} 1 & 5 & 0 \\ 2 & 6 & 4 \\ 3 & 8 & 2 \end{bmatrix}.$$

For at gøre dette, opstilles matricen $[A \mid I_3\,]$:

$$[A \mid I_3\,] = \left[\begin{array}{ccc|ccc} 1 & 5 & 0 & 1 & 0 & 0 \\ 2 & 6 & 4 & 0 & 1 & 0 \\ 3 & 8 & 2 & 0 & 0 & 1 \end{array}\right].$$

Dernæst benytter vi elementære rækkeoperationer indtil matricen er på reduceret trappeform. Da får vi, at

$$[I_3 \mid A^{-1}\,] = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & -2/3 & 1/3 \\ 0 & 1 & 0 & -1/12 & 7/12 & 1/12 \\ 0 & 0 & 1 & 1/6 & -1/6 & -1/6 \end{array}\right].$$

Vi kan derved konkludere, at A invers er givet ved

$$A^{-1} = \begin{bmatrix} 0 & -2/3 & 1/3 \\ -1/12 & 7/12 & 1/12 \\ 1/6 & -1/6 & -1/6 \end{bmatrix}.$$

◀

Vi kan formulere et mere generelt resultat, hvis vi ønsker at konstruere en elementærmatrix, der udelukkende udfører rækkeudskiftninger. Hvis vi ønsker at udføre rækkeudskiftningen: $c\vec{r}_i+\vec{r}_j \to \vec{r}_j$, for $i \neq j$, i en $m\times n$-matrix A, er det det samme som at gange elementærmatricen Q på fra venstre, hvor $Q = I_m + D$. Matricen D er en $m \times m$-matrix, med 0 i alle indgange, pånær indgangen d_{ji}, som har skalaren c [2, s. 96]. Altså får vi, at produktet $QA = (I_m+D)A$ opfylder vores ønske.

3 | Lineære programmeringsproblemer

I dette kapitel afklarer vi, hvad et lineært programmeringsproblem er, samt hvad der skal opfyldes, før vi kan finde en mulig løsning til det. Vi vil også beskrive, hvordan lineære programmeringsproblemer kan omskrives på standardform, hvis konstruktion gør det muligt for os at bruge de gængse værktøjer fra lineær algebra, som introduceret i kapitel 2, til at løse dem.

Vi starter kapitlet ud med en grafisk fortolkning af en løsning på et lineært programmeringsproblem [4, s. 10-13].

3.1 Introducerende eksempel: Blandingsmodellen

I eksemplet her vil vi se på et lineært programmeringsproblem, der involverer kombinationen af to variable, som kaldes blandingsmodellen. Problemet består i at bestemme, hvor meget af hver ingrediens, der skal kombineres, sådan at den totale omkostning minimeres, mens kombinationen imødekommer basale krav til indholdet.

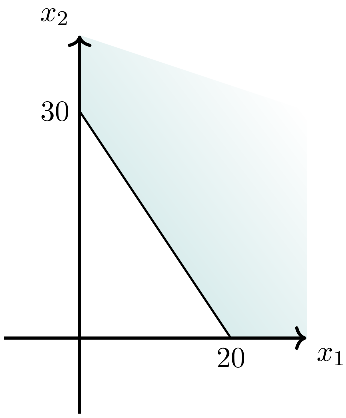
Vi antager her, at en landmand skal fodre sine dyr, og at vi kender prisen og næringsstofferne A, B og C, som måles i enhed/kg, i fodret. Dette er givet i nedenstående tabel.

Foder	A	B	C	Pris (kr/kg)
Foder 1	3	7	3	10
Foder 2	2	2	6	4

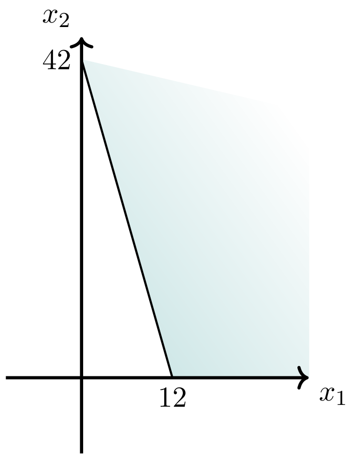
Derudover ved vi, at dyrene skal mindst have 60 enheder af A, mindst 84 enheder af B og mindst 72 enheder af C. Landmanden vil så gerne opnå at fodre med en sammensætning af fodret, der opfylder dyrenes næringsbehov og samtidig er billigst muligt.

Hvis vi oversætter dette til matematik, kan vi forstå det således, at der skal være x_1 antal kg af foder 1 og x_2 antal kg af foder 2. På den måde bliver det klart, at x_1 og x_2 skal være *ikke-negative*. Dermed må en diæts indhold af næringsstof A, bestående af hhv. foder 1 og 2, kunne beskrives som $3x_1 + 2x_2 = A$. Her ved landmanden dog, at diæten skal bestå af *mindst* 60 enheder af næringsstof A, hvorfor udtrykket kan omskrives til $3x_1+2x_2 \geq 60$. Dette kaldes en *lineær begrænsning*. Vi kan opstille ligningerne for nærringstoffernde B og C på samme måde, hvor vi får følgende begrænsninger B: $7x_1 + 2x_2 \geq 84$ og C: $3x_1 + 6x_2 \geq 72$. Her er dog gjort en række antagelser, som eksempelvis at dyrene ikke kan få for meget af et næringsstof, hvorfor problemet naturligvis er simplificeret.

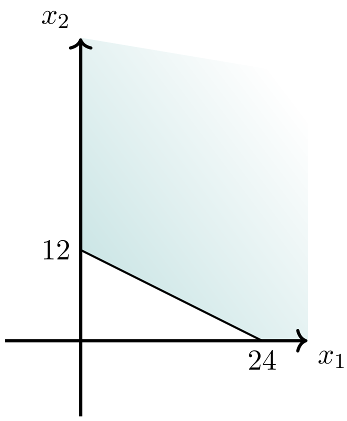
Dette kan illustreres grafisk, hvor vi ser på alle de punkter (x_1, x_2), der opfylder ulighederne for begrænsningerne til næringstofferne A, B og C - se figur 3.1, 3.2 og 3.3.



Figur 3.1: A: $3x_1 + 2x_2 \geq 60$

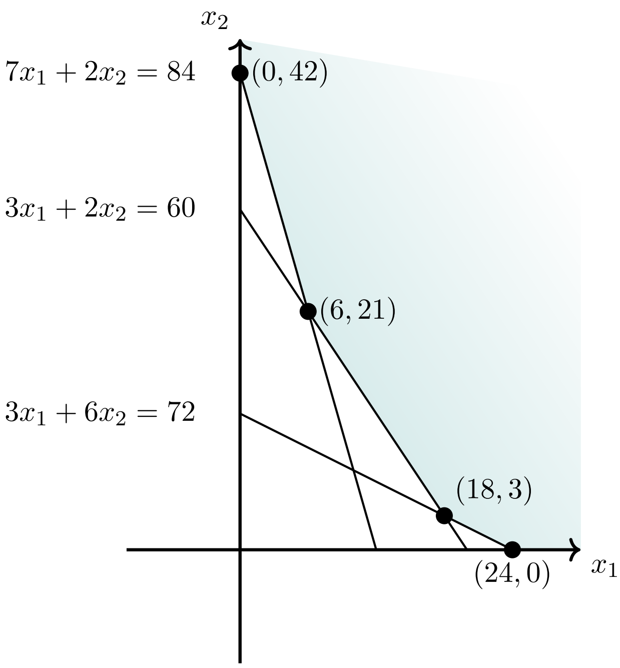


Figur 3.2: B: $7x_1 + 2x_2 \geq 84$



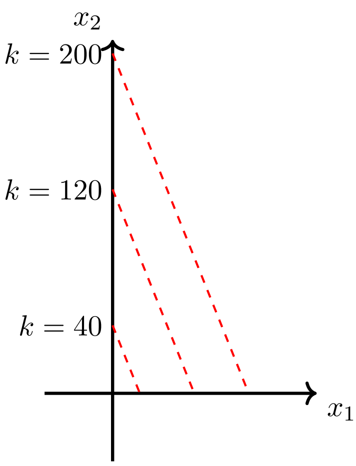
Figur 3.3: C: $3x_1 + 6x_2 \geq 72$

Det blå område over linjerne, for hvert næringsstof, er altså det område, hvor der findes en løsning - hvor begrænsningen opfyldes. Hvis vi sætter de tre begrænsninger i samme figur, ses det, at de danner et fælles område - det er altså begrænsningernes fællesmængde. Indenfor dette område må alle krav til næringsindhold være opfyldt. Figur 3.4 illustrerer dette.



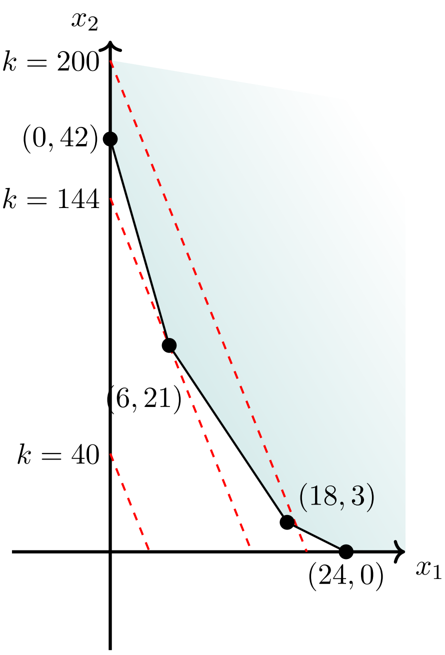
Figur 3.4: Afbildning af begrænsningernes fællesmængde. I det blå område er alle begrænsninger opfyldt.

Hver sammensætning af fodret - hver diæt - har en given samlet pris, som kan skrives som en lineær ligning. Denne kaldes for *omkostningsfunktionen*, og er altså den samlede omkostning, som ønskes minimeret. I vores tilfælde vil omkostningen være givet ved $f(x_1, x_2) = 10x_1 + 4x_2$ - se tidligere tabel.
Vi indtegne en række parallelle linjer, som kaldes *niveaukurver*, for omkostningsfunktionen $f(x_1, x_2) = k$, for nogle konstanter k. Disse ses i figur 3.5.



Figur 3.5: Niveaukurverne for omkostningsfunktionen. Konstanten k varierer, sådan at et større k flytter linjerne.

Da niveaukurverne altså er et udtryk for den samlede omkostning og denne ønskes mindst muligt, så vil den mindste omkostning derfor findes *første gang* en niveaukurve rører fællesmængden i et punkt (x_1, x_2). Dette sker, når $k = 144$, som det ses i figur 3.6.



Figur 3.6: Grafisk løsning.

Fra figur 3.6 kan det aflæses, at den optimale sammensætning af fodret består af 6 enheder af foder 1 og 21 enheder af foder 2. Det svarer til en samlet mindste omkostning på 144 kr - svarende til $f(6, 21) = 10 \cdot 6 + 4 \cdot 21 = 144$.

Denne grafiske fortolkning af en løsning virker dog kun for lineære programmeringsproblemer med højst tre variable, men kan anvendes både på minimerings- og maksimeringsproblemer.

3.2 Generelle maksimerings- og minimeringsproblemer

Vi så i ovenstående introducerede eksempel, at lineær programmering kan bruges til at optimere en lineær funktion med flere variable, underlagt lineære begrænsninger. I et generelt problem opstilles da den funktion Z, som ønskes optimeret, kaldet *objektfunktionen*, samt et lineært ligningssystem, der betegner nogle begrænsninger [5, s. 1].

Et generelt lineært programmeringsproblem med n variable og m begrænsninger opstilles på følgende måde:

$$
\begin{array}{rl}
\text{Maksimér eller minimér} & Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{Underlagt} & a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \ (\leq, =, \geq)\ b_1 \\
& a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \ (\leq, =, \geq)\ b_2 \\
& \vdots \\
& a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \ (\leq, =, \geq)\ b_m \\
& x_1, x_2 \ldots, x_n (\leq, \geq) 0
\end{array}
\tag{3.1}
$$

Vektor $\vec{c}$ kaldes *omkostningsvektoren* og vektor $\vec{x}$'s komponenter kaldes *beslutningsvariable*. Vektoren $\vec{b}$ kaldes for *begrænsningsvektoren*. Det ses, at skalarproduktet mellem $\vec{c}$ og $\vec{x}$, som begge tilhører $\mathbb{R}^n$, giver objektfunktionen $Z = \vec{c} \cdot \vec{x} = \sum_{j=1}^{n} c_j x_j$ [2, s. 2].
Begrænsningerne, der er opstillet under objektfunktionen, opdeles i to typer:

1. *Hovedbegrænsninger.*
2. *Den ikke-negative-* eller *ikke-positive begrænsning.*

Den i'te hovedbegrænsning er $\vec{a}_i \cdot \vec{x} (\leq, =, \geq) b_i$, hvor $i = 1, 2, \ldots, m$. For en variabel x_k underlagt den ikke-negative begrænsning gælder det at $x_k \geq 0$. Modsat for en variabel x_k underlagt den ikke-positive begrænsning gælder det at $x_k \leq 0$ [5, s. 3]. Det skal dog bemærkes, at nogle beslutningsvariable kan være ubegrænsede. Dette forklares i afsnit 3.2.1.

Oftest er det den ikke-negative begrænsning vi arbejder med. Betragter vi et problem fra virkeligheden er dette klart, da virkelighedsnære variable ofte ikke kan være negative værdier, eksempelvis arbejdstimer - eller foderet fra ovenstående eksempel i afsnit 3.1. Af samme argument følger, at vi sjældent arbejder med den ikke-positive begrænsning.

Lad mængderne M_1, M_2 og M_3 være disjunkte delmængder af indeksmængden $I = \{1, 2, \ldots, m\}$, hvor $\cup_{k=1}^{3} M_k = I$. Lad også N_1 og N_2 være delmængder af indeksmængden $J = \{1, 2, \ldots, n\}$.

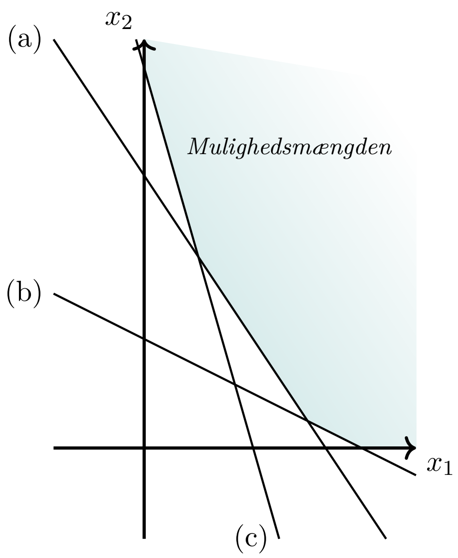
Med denne notation kan vi betragte et lineært programmeringsproblem på følgende form:

$$
\begin{array}{rlll}
\text{Maksimér eller minimér} & Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n & & \\
\text{Underlagt} & \vec{a}_i \cdot \vec{x} \ \leq \ b_i, & & i \in M_1 \\
& \vec{a}_i \cdot \vec{x} \ \geq \ b_i, & & i \in M_2 \\
& \vec{a}_i \cdot \vec{x} \ = \ b_i, & & i \in M_3 \\
& x_j \geq 0, & & j \in N_1 \\
& x_j \leq 0, & & j \in N_2
\end{array}
$$

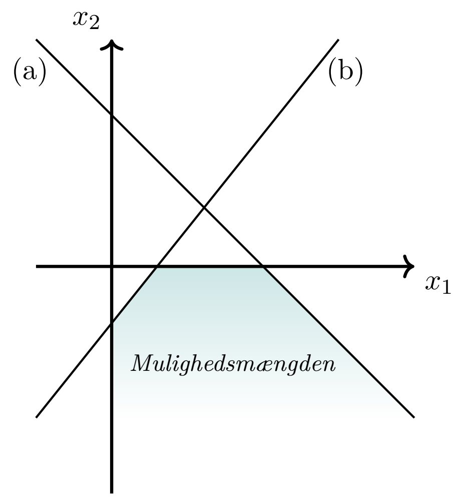
Her opdeler mængderne M_1, M_2 og M_3 begrænsningerne i "mindre end"- og "større end"-uligheder samt ligheder. For hvert element $i \in I$ er der tilknyttet en vektor $\vec{a}_i \in \mathbb{R}^n$ og en reel skalar b_i, som sammen med $\vec{x}$ danner den i'te hovedbegrænsning. Elementerne i N_1 og N_2 angiver, hvilke beslutningsvariable, der skal være henholdsvis ikke-negative eller ikke-positive [2, s. 3].

Definition 3.2 Mulige løsninger

En vektor $\vec{x}$, som opfylder alle begrænsninger, kaldes en *mulig løsning*. Mængden af alle mulige løsninger kaldes *mulighedsmængden* [2, s. 3].



Figur 3.7: En mulighedsmængde afgrænset af to variable, underlagt den ikke-negative begrænsning.

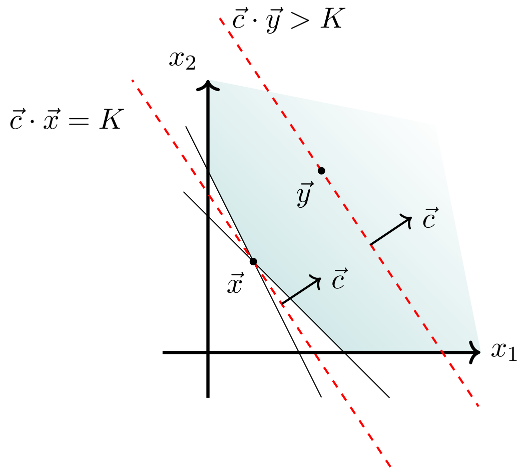


Figur 3.8: En mulighedsmængde afgrænset både af en variabel underlagt den ikke-positive begrænsning, og en variabel underlagt den ikke-negative begrænsning.

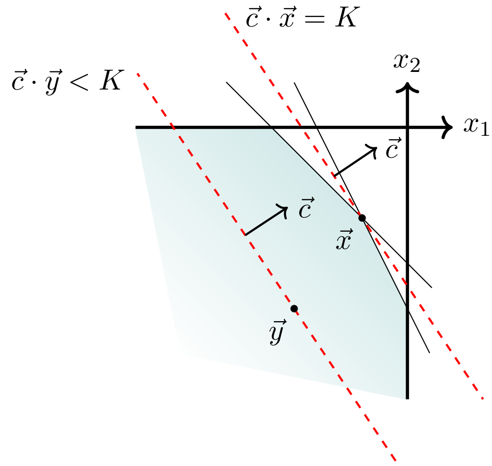
Antag at begrænsningerne i et lineært programmeringsproblem alle er stillet med positive koefficienter. Hvis vi betragter figur 3.7, er der tale om tre begrænsninger alle på formen $\vec{a}_i \cdot \vec{x} \geq b_i$, hvor $i \in M_2$. Alle de mulige løsninger vil befinde sig langs linjerne, som afgrænser det blå område, samt alle de punkter, som ligger inden for det blå område.
Hvis vi derimod ser på figur 3.8, er der tale om to begrænsninger, hvor begrænsning (a)'s indeks tilhører indeksmængden M_1 og begrænsning (b)'s indeks tilhører indeksmængden M_2. Desuden vil indekset $1 \in N_1$ og $2 \in N_2$, da x_1 er underlagt den ikke-negative begrænsning og x_2 er underlagt den ikke-positive begrænsning. Mulighedsmængden kan altså i $\mathbb{R}^2$ grafisk aflæses, i en given kvadrant, alt efter hvilke begrænsninger problemets beslutningsvariable er underlagt.

Betragt nu et lineært programmeringssproblem med objektfunktionen givet ved omkostningsvektoren $\vec{c}$ [2, s. 3]. Formålet med et minimeringsproblem er da at finde den mulige løsning $\vec{x}$, som opfylder $\vec{c} \cdot \vec{x} \leq \vec{c} \cdot \vec{y}$, for enhver mulig løsning $\vec{y}$. Modsat gælder det for et maksimeringsproblem om at finde den mulige løsning $\vec{x}$, som opfylder $\vec{c} \cdot \vec{x} \geq \vec{c} \cdot \vec{y}$, for enhver mulig løsning $\vec{y}$.

Vi har indtil nu betragtet lineære programmeringsproblemer, hvor der eksisterer en optimal værdi for objektfunktionen. Der kan dog opstå specialtilfælde, hvor objektfunktionen er ubegrænset. Dette redegøres for i nedenstående.



Figur 3.9: Maksimeringsproblem. Ubegrænset opadtil.



Figur 3.10: Minimeringsproblem. Ubegrænset nedadtil.

Givet et maksimeringsproblem, som afbildet i figur 3.9, vil den optimale værdi af objektfunktionen være ∞, hvis der gælder, at vi ikke kan finde et $K \in \mathbb{R}$, som opfylder, at $K \geq \vec{c} \cdot \vec{x}$ for alle $\vec{x}$ i mulighedsmængden. Det siges, at objektfunktionen er ubegrænset opadtil.
På samme måde kan vi se på et givet minimeringsproblem, som afbildet i figur 3.10. Her vil den optimale værdi af objektfunktionen være $-\infty$, hvis det gælder, at vi ikke kan finde et $K \in \mathbb{R}$, som opfylder, at $K \leq \vec{c} \cdot \vec{x}$ for alle $\vec{x}$ i mulighedsmængden. Det siges, at objektfunktionen er ubegrænset nedadtil [2, s. 3].

3.2.1 Generalisering af lineære programmeringsproblemer

Et lineært programmeringsproblem, med mange variable, kan med fordel beskrives med en koefficientmatrix, da vi ikke kan betragte og løse problemet grafisk, hvis vi har flere end tre variable. Vi kan derimod benytte værktøjer fra lineær algebra, som beskrevet i kapitel 2, til løsning af ligningssystemet [6, s. 4]. Vi opstiller en $m \times n$-matrix

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix},$$

hvis indgange er hovedbegrænsningernes koefficienter. Hvis det for et lineært programmeringsproblem gælder, at alle hovedbegrænsninger er på samme form, kan vi opskrive hovedbegrænsningerne på følgende måde:

$$A\vec{x}(\leq, =, \geq)\vec{b}.$$

Vi vil herunder introducere to former af lineære programmeringsproblemer, hvor dette er tilfældet.

Definition 3.3 Standard minimeringsproblemer

I et *standard minimeringsproblem* ønsker vi at bestemme den mulige løsning $\vec{x}$, der minimerer objektfunktionen $Z = \vec{c}\cdot\vec{x}$, underlagt hovedbegrænsningerne $A\vec{x} \geq \vec{b}$ og den ikke-negative begrænsning. Det gælder da at indeksmængderne $M_1 = M_3 = N_2 = \emptyset$ [6, s. 5].

Eksempel 3.4.
Ligningssystemet på formen

$$\begin{aligned} \text{Minimér} \quad & Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{Underlagt} \quad & A\vec{x} \ \geq \ b_i, \\ & x_j \geq 0, \quad j = 1, \dots, n \end{aligned}$$

siges at være et standard minimeringsproblem, da alle hovedbegænsningerne er begrænset ved uligheden $\geq$, og har ikke-negative variable x_j. ◀

Definition 3.5 Standard maksimeringsproblemer

I et *standard maksimeringsproblem* ønsker vi at bestemme den mulige løsning $\vec{x}$, der maksimerer objektfunktionen $Z = \vec{c} \cdot \vec{x}$, underlagt hovedbegrænsningerne $A\vec{x} \leq \vec{b}$ og den ikke-negative begrænsning. Det gælder da at indeksmængderne $M_2 = M_3 = N_2 = \emptyset$ [6, s. 4].

Eksempel 3.6.
Ligningssystemet på formen

$$\begin{aligned} \text{Maksimér} \quad & Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\ \text{Underlagt} \quad & A\vec{x} \ \leq \ b_i, \\ & x_j \geq 0, \quad j = 1, \dots, n \end{aligned}$$

siges at være et standard maksimeringsproblem, da alle hovedbegænsningerne er begrænset ved uligheden $\leq$, og har ikke-negative variable x_j. ◀

Da vi nu har set, at disse problemer kan skrives som matrix-vektor-ligninger, vil vi nu vise, hvordan vi kan skrive et arbitrært lineært programmeringsproblem om til et standard minimeringsproblem. Da der i et arbitrært lineært programmeringsproblem ikke er krav om, at problemets beslutningsvariable er begrænsede, definerer vi herunder disse tilfælde:

Definition 3.7 Ubegrænsede variable

Hvis en variabel x_j ikke opfylder hverken N_1 eller N_2, siges denne at være en ubegrænset variabel [2, s. 3].

Med henblik på dette, kan vi nu beskrive metoden til at skrive et givet problem på form som et standard minimeringsproblem:

1. Enhver begrænsning på formen $\vec{a}_i \cdot \vec{x} \leq b_i$ kan omskrives til en begrænsning i et standard minimeringsproblem, ved at gange igennem med -1.

2. Enhver begrænsning bestående af en lighed $\vec{a}_i \cdot \vec{x} = b_i$ kan omskrives til to lineære uligheder $\vec{a}_i \cdot \vec{x} \geq b_i$ og $-\vec{a}_i \cdot \vec{x} \geq -b_i$.
3. Enhver ubegrænset beslutningsvariabel x_j elimineres ved at substituere den med kunstige variable, så $x_j = x_j^+ - x_j^-$, i begrænsningerne og i objektfunktionen. De kunstige variable x_j^+ og x_j^- kan nu underlægges den ikke-negative begrænsning. Således vil x_j kunne repræsenteres ved differencen mellem to positive reelle tal og dermed opfylde N_1 [2, s. 5].
4. Enhver beslutningsvariabel, underlagt den ikke-positive begrænsning, kan blive underlagt den ikke-negative begrænsning ved at gange alle dens tilhørende koefficienter igennem med -1.

En lignende metode anvendes til at omskrive et arbitrært lineært programmeringsproblem til et standard maksimeringsproblem. Vi viser her et eksempel på omskrivning til et standard minimeringsproblem.

Eksempel 3.8.
Givet er et lineær programmeringsproblem:

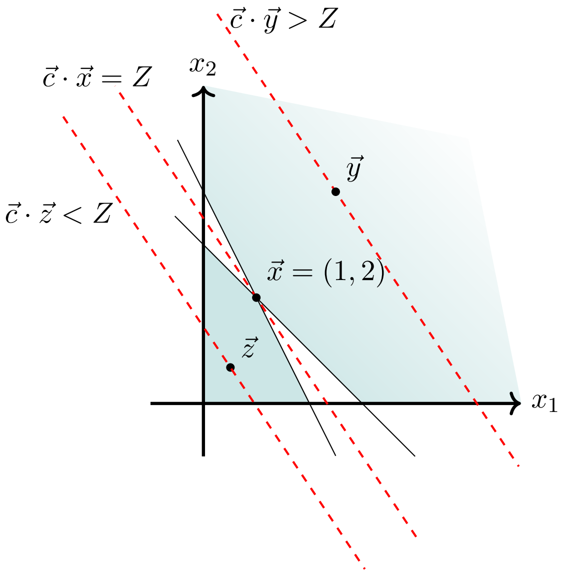
$$\begin{array}{lrcll} \text{Minimér} & Z = 5x_1 - 4x_2 + x_3 & & & \\ \text{Underlagt} & 2x_1 + 3x_2 & \geq & 8 & (a) \\ & 5x_1 + 2x_3 & = & 11 & (b) \\ & x_1 - 2x_2 & \leq & -3 & (c) \\ & x_1, x_2, x_3 \geq 0 & & & \end{array}$$

Vi ønsker at omskrive problemet til et standard minimeringsproblem. Hovedbegrænsning (a) er allerede på formen $\vec{a}_i \cdot \vec{x} \geq b_i$ og ændres derfor ikke. Hovedbegrænsning (b) er på formen $\vec{a}_i \cdot \vec{x} = b_i$ og omskrives derved ved at dele begrænsningen op i to lineære uligheder på formen $\vec{a}_i \cdot \vec{x} \geq b_i$ og $-\vec{a}_i \cdot \vec{x} \geq -b_i$. Hovedbegrænsning (c) er på formen $\vec{a}_i \cdot \vec{x} \leq b_i$ og omskrives derved ved at gange igennem med -1. Da får vi følgende:

$$\begin{array}{lrcl} \text{Minimér} & Z = 5x_1 - 4x_2 + x_3 & & \\ \text{Underlagt} & 2x_1 + 3x_2 & \geq & 8 \\ & 5x_1 + 2x_3 & \geq & 11 \\ & -5x_1 - 2x_3 & \geq & -11 \\ & -x_1 + 2x_2 & \geq & 3 \\ & x_1, x_2, x_3 \geq 0 & & \end{array}$$

◀

Ethvert standard maksimeringsproblem kan omskrives til et ækvivalent standard minimeringsproblem ved at gange objektfunktionen igennem med -1, eller ved at skrive begrænsningerne på formen $\sum_{j=1}^{n} a_{ij}x_j \leq b_i$ om til formen $\sum_{j=1}^{n}(-a_{ij})x_j \geq -b_i$. Det vil sige, at hvis vi ønsker at *maksimere* objektfunktionen $f(x_1, x_2) = Z$, er det det samme som at *minimere* $-f(x_1, x_2) = Z$. Dette kan ses på figur 3.11.



Figur 3.11: Afbildning af omskrivning fra standard maksimeringsproblem til standard minimeringsproblem. Vektor $\vec{x}$ er her den optimale løsning til både minimerings- og maksimeringsproblemet, hvorimod $\vec{z}$ er en mulig løsning til maksimeringsproblemet og $\vec{y}$ er en mulig løsning til minimeringsproblemet.

På figur 3.11 ser vi en mulighedsmængde tilhørende et standard minimeringsproblem, samt en mulighedsmængde tilhørende et standard maksimeringsproblem. Disse to problemer er ækvivalente. Desuden ses en løsning $\vec{x}$, som vi antager at være den optimale løsning. Sætter vi $\vec{x}$ ind i objektfunktionen, $10x_1 + 4x_2 = Z$, fra det introducerende eksempel i afsnit 3.1, får vi $10 \cdot 1 + 4 \cdot 2 = 18$ og $-10 \cdot 1 - 4 \cdot 2 = -18$. Vektoren $\vec{x}$ er da den optimale løsning for begge problemer.

3.3 Standardform

Når vi arbejder med optimeringsproblemer, kan de, som vi har set ovenfor, antage forskellige former - altså hhv. minimerings- og maksimeringsproblemer og begrænsninger der består af både ligheder og uligheder i begge retninger. Dette er dog problematisk i forhold til at løse lineære programeringsproblemer, da det vil blive mere omfattende. Derfor kan maksimerings- og minimeringsproblemer omskrives til ækvivalente problemer, lineære programmeringsproblemer på *standardform*, hvilket senere skal vise sig at være udgangspunktet for Simplex-metoden i kapitel 5.

At omskrive et vilkårligt lineært programmeringsproblem til standardform vil kræve, at der indføres nye begreber, samt at der udføres en række trin til elimination af uligheder og ubegrænsede beslutningsvariable i alle begrænsningerne. Disse beskrives i nedenstående, hvorefter standardform bliver endeligt defineret.

På standardform vil vi have elimineret begrænsningerne, som opfylder M_1 og M_2 og i stedet have dem alle til at opfylde M_3.

Elimination af ulighedsbegrænsninger:

1. *Underskudsvariable*:
   For alle begrænsninger med indeks $i \in M_1$, indfører vi *underskudsvariable* $s_i \geq 0$. Vi kan da omskrive uligheden til en lighed ved at lægge den ikke-negative værdi s_i til på venstresiden. Dette giver
   $$\sum_{j=1}^{n} a_{ij}x_j \leq b_i \Rightarrow \sum_{j=1}^{n} a_{ij}x_j + s_i = b_i.$$

2. *Overskudsvariable*:
   For alle begrænsninger med indeks $i \in M_2$ indfører vi *overskudsvariable* $s_i \geq 0$. Vi kan da omskrive uligheden til en lighed ved at trække den ikke-negative værdi s_i fra venstresiden. Dette giver
   $$\sum_{j=1}^{n} a_{ij}x_j \geq b_i \Rightarrow \sum_{j=1}^{n} a_{ij}x_j - s_i = b_i.$$

Begge tilføjes altid til venstresiden af uligheden, og om der er tale om en underskuds- eller overskudsvariabel er altså bestemt af, hvilken type ulighed begrænsningen er konstrueret ud fra [2, s. 5-6].

Vi ønsker nu at eliminere de ubegrænsede variable. Dette gøres som beskrevet i afsnit 3.2.1. Det samme gælder for de variable, som er underlagt den ikke-positive begrænsning. Til sidst ønsker vi at have problemet på formen af et minimeringsproblem. I tilfældet at problemet er på formen af et maksimeringsproblem, skal vi gange objektfunktionen igennem med -1. Gennem de ovenstående trin kan vi transformere alle lineære programmeringsproblemer til problemer på *standardform*.

Definition 3.9 Standardform

Et lineært programmeringsproblem på formen

$$\begin{array}{ll}
\text{Minimér} & Z = c_1x_1 + c_2x_2 + \cdots + c_nx_n \\
\text{Underlagt} & \quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n + s_1 \ = \ b_1 \\
 & \quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n + s_2 \ = \ b_2 \\
 & \quad \vdots \\
 & \quad a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n + s_m \ = \ b_m \\
 & \quad x_1, x_2 \ldots, x_n, s_1, s_2 \ldots s_m \geq 0
\end{array}$$

siges at være på *standardform*, da begrænsningerne, som objektfunktionen er underlagt, består af ligheder og den ikke-negative begrænsning [2, s. 4].

På standardform arbejder vi altså udelukkende med *minimerings*problemer. Det kan lade sig gøre, fordi maksimeringsproblemer, som sagt, kan omskrives til minimeringsproblemer.

Eksempel 3.10.
Givet er et lineært programmeringsproblem:

$$
\begin{array}{ll}
\text{Minimér} & Z = 2x_1 + 4x_2 - 4x_3 \\
\text{Underlagt} & 2x_1 + 3x_2 \ \leq \ 8 \\
 & 5x_1 \ \leq \ 11 \\
 & x_1 - 2x_2 \ \geq \ -3 \\
 & x_1, x_2 \geq 0
\end{array}
$$

Da kan vi omskrive det til standardform ved hjælp af ovenstående metode. Vi analyserer derfor først ulighederne og bemærker, at ulighederne i første og anden begrænsning kan omskrives til lighedstegn ved at tilføje underskudsvariable. Omvendt gælder det for tredje begrænsning, at vi kan tilføje en overskudsvariabel for at ophæve uligheden. Ydermere kan vi eliminere den ubegrænsede variabel x_3 ved at omskrive den til $x_3 = x_3^+ - x_3^-$ og underlægge x_3^+ og x_3^- den ikke-negative begrænsning. Vi får da:

$$
\begin{array}{ll}
\text{Minimér} & Z = 2x_1 + 4x_2 - 4x_3^+ + 4x_3^- \\
\text{Underlagt} & 2x_1 + 3x_2 + s_1 \ = \ 8 \\
 & 5x_1 + s_2 \ = \ 11 \\
 & x_1 - 2x_2 - s_3 \ = \ -3 \\
 & x_1, x_2, x_3^+, x_3^-, s_1, s_2, s_3 \geq 0
\end{array}
$$

◀

Selvom et lineært programmeringsproblem omskrives til standardform, vil vi opnå samme løsning som for det oprindelige lineære programmeringsproblem. Der gives et geometrisk argument for dette i kapitel 4.

4 Geometrien i lineære programmeringsproblemer

I dette kapitel introduceres en række geometriske principper, som vi anvender til at give en geometrisk beskrivelse af lineære programmeringsproblemer og deres løsninger. Kapitlet vil bemærke, hvornår der findes optimale løsninger og i disse tilfælde, hvor de findes. Dette giver os muligheden for at anskue specifikke steder i problemets geometriske repræsentation i jagten på en optimal løsning.

4.1 Polyedre og konvekse mængder

Lad os indlede med en genereliseret definition af det geometriske objekt kaldet et *polyeder*:

Definition 4.1 Polyedre

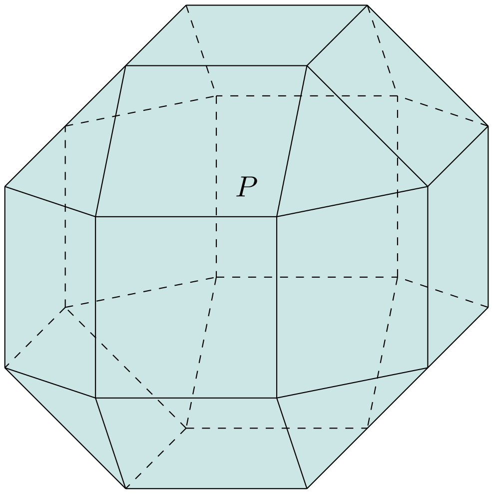
Lad A være en $m \times n$-matrix og $\vec{b}$ en vektor i $\mathbb{R}^m$. Mængden

$$P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} \geq \vec{b}\}$$

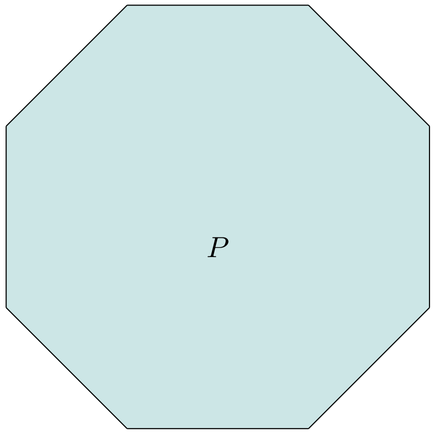
kaldes da et *polyeder* [2, s. 42].

Bemærk at vi kan betragte polyedre som geometriske n-dimensionelle objekter konstrueret af planer. Dette er illustreret i figur 4.1 og 4.2. Derudover kan vi også betragte polyedre som mulighedsmængder til vilkårlige lineære programmeringsproblemer, som er givet på formen af et standard lineært minimeringsproblem - se definition 3.3.

Bemærk at vi i dette kapitel fortsat kun anskuer lineære minimeringsproblemer.



Figur 4.1: Et polyeder i tre dimensioner.



Figur 4.2: Et polyeder i to dimensioner.

Vi har brug for at definere yderligere to geometriske objekter, for at kunne betragte et lineært programmeringsproblems lineære begrænsninger geometrisk. Vi definerer først *hyperplaner*:

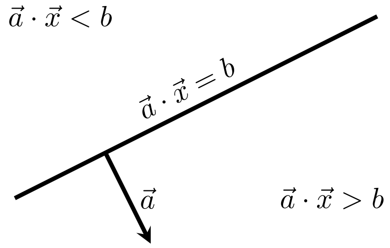
Definition 4.2 Hyperplaner

Lad $\vec{a}$ være en vektor i $\mathbb{R}^n$, hvor $\vec{a} \neq \vec{0}$ og lad b være en reel skalar. Mængden

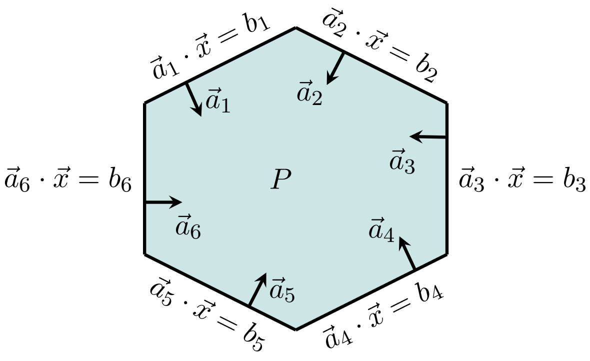
$$H = \{\vec{x} \in \mathbb{R}^n \mid \vec{a} \cdot \vec{x} = b\}$$

kaldes da en *hyperplan* [2, s. 43].

Vi ser her, at hyperplaner beskriver løsningsmængden til en lineær begrænsning bestående af en lighed. Et eksempel på den geometriske repræsentation af disse former for begrænsninger kan betragtes i figur 4.3.



Figur 4.3: Hyperplan i $\mathbb{R}^2$.



Figur 4.4: Flere halvrum i $\mathbb{R}^2$, hvor fællesmængden af begrænsningerne danner P.

Bemærk at et hyperplan deler vektorrummet op i to lige store *halvrum*. Disse to halvrum defineres som følgende:

Definition 4.3 Halvrum

Lad $\vec{a}$ være en vektor i $\mathbb{R}^n$, hvor $\vec{a} \neq \vec{0}$ og b er en reel skalar. Følgende mængder

(a) $H_l = \{\vec{x} \in \mathbb{R}^n \mid \vec{a} \cdot \vec{x} \leq b\}$

(b) $H_u = \{\vec{x} \in \mathbb{R}^n \mid \vec{a} \cdot \vec{x} \geq b\}$

kaldes da henholdsvis det *nedre* og *øvre halvrum* [2, s. 43].

Det nedre halvrum beskriver altså løsningenmængden til en lineær begrænsning på formen $\vec{a} \cdot \vec{x} \leq b$ og modsat gælder det, at det øvre halvrum er løsningsmængden til en lineær begrænsning på formen $\vec{a} \cdot \vec{x} \geq b$.

Givet disse definitioner kan vi nu fastslå, at et polyeder består af fællesmængden af et endeligt antal af halvrum - se figur 4.4. Tilsvarende betyder dette, at mulighedsmængden til et lineært programmeringsproblem er den mængde af vektorer, der opfylder alle de givne begrænsninger. Dette svarer til vores definition 3.2 af mulige løsninger i det forrige kapitel.

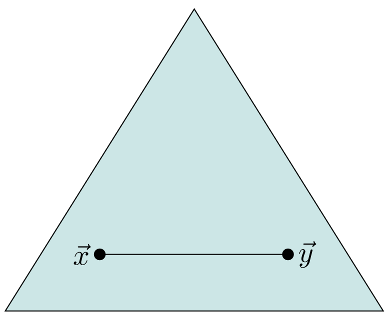
Lineære programmeringsproblemer er altså en speciel type af optimeringsproblemer, hvor vi ønsker at finde et ekstremum af en lineær funktion defineret på et polyeder. Vi vil afslutte dette afsnit med at vise, at lineære programmeringsproblemer hører under en mere generel type af optimeringsproblemer - samt en naturlig konsekvens af dette.

Et polyeder er en speciel type mængde, som kaldes en *konveks mængde*. Det at en mængde er konveks, defineres herunder:

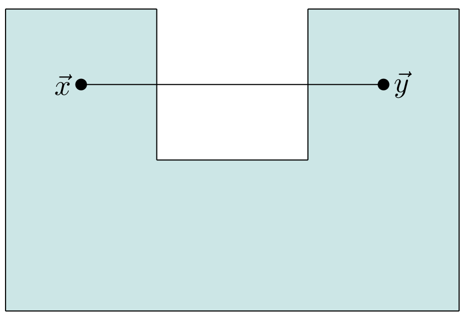
Definition 4.4 Konvekse mængder

En mængde $S \subseteq \mathbb{R}^n$ kaldes *konveks*, hvis og kun hvis det for ethvert par $\vec{x}$, $\vec{y} \in S$ gælder at $\lambda\vec{x} + (1 - \lambda)\vec{y} \in S$ for alle $\lambda \in [0, 1]$ [2, s. 43].

Bemærk at $\lambda\vec{x}+(1-\lambda)\vec{y}$ er et linjestykke, der forbinder begge vektorer i rummet, idet $\lambda \in [0, 1]$. Vi kan derfor nemt danne os en idé om, hvilke mængder der er konvekse - se figur 4.5 og 4.6.



Figur 4.5: En konveks mængde i $\mathbb{R}^2$.



Figur 4.6: En ikke-konveks mængde i $\mathbb{R}^2$.

Bemærk at vektorerne på det forbindende linjestykke kan beskrives som et specielt vægtet gennemsnit af linjestykkets endepunkter. Et sådan vægtet gennemsnit kaldes en *konveks kombination*:

Definition 4.5 Konvekse kombinationer

Lad $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k$ være vektorer i $\mathbb{R}^n$ og lad $\lambda_1, \lambda_2, \ldots, \lambda_k$ være ikke-negative skalarer, hvor $\sum_{i=1}^{k} \lambda_i = 1$.
Vektoren

$$\sum_{i=1}^{k} \lambda_i \vec{x}_i$$

kaldes en *konveks kombination* af vektorerne $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_k$. Hvis det i en konveks kombination gælder $\lambda_i < 1$ for alle $i = 1, 2, \ldots, k$, så kaldes det en *skarp konveks kombination* [2, s. 44].

Bemærk at - modsat et generelt vægtet gennemsnit - kræver en konveks kombination at vægtenes sum er lig én. En konveks mængde kræver derfor, at alle konvekse kombinationer af to vektorer i mængden også er i mængden. Vi kan nu bruge dette til at bevise, at ethvert polyeder er konveks.

Sætning 4.6

(a) Fællesmængden af endeligt mange konvekse mængder er konveks.

(b) Ethvert polyeder er konvekst [2, s. 44].

Bevis:

(a) Lad S_i være konvekse mængder, hvor $i \in I$, og hvor I er en endelig indeksmængde. Antag da, at $\vec{x}$ og $\vec{y}$ tilhører fællesmængden af et antal konvekse mængder, $\cap_{i \in I} S_i$.
Lad $\lambda \in [0, 1]$. Da alle mængderne S_i er konvekse, og indeholder $\vec{x}$ og $\vec{y}$, får vi

$$\lambda \vec{x} + (1 - \lambda)\vec{y} \in S_i.$$

Dette viser dermed også, at $\lambda \vec{x} + (1 - \lambda)\vec{y}$ tilhører fællesmængden af mængderne S_i. Derfor er $\cap_{i \in I} S_i$ også konveks.

(b) Lad $\vec{a} \neq \vec{0}$ og b en reel skalar. Antag at $\vec{x}$ og $\vec{y}$ opfylder $\vec{a} \cdot \vec{x} \geq b$ og $\vec{a} \cdot \vec{y} \geq b$. Da ved vi fra definition 4.3, at $\vec{x}$ og $\vec{y}$ tilhører det samme halvrum. Lad $\lambda \in [0, 1]$. Da $\vec{x}$ og $\vec{y}$ tilhører det samme halvrum, kan den konvekse kombination af $\vec{a} \cdot \vec{x}$ og $\vec{a} \cdot \vec{y}$ udtrykkes:

$$\lambda \vec{a} \cdot \vec{x} + (1 - \lambda)\vec{a} \cdot \vec{y} = \vec{a} \cdot (\lambda \vec{x} + (1 - \lambda)\vec{y}) \geq \lambda b + (1 - \lambda)b = b.$$

Altså kan vi se, at $\lambda \vec{x} + (1 - \lambda)\vec{y}$ også tilhører det samme halvrum, hvorfor halvrummet må være konvekst. Idet et polyeder er en fællesmængde af halvrum, er ethvert polyeder konvekst som konsekvens af (a) [2, s. 45].

■

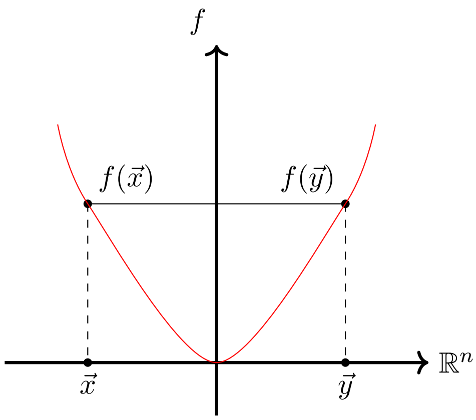
Vi har nu vist, at polyederet vi ønsker at optimere over er en konveks mængde. Konveksitet er et større begreb, der ikke kun dækker mængder og kombinationer. Vi vil nu vise, at den lineære objektfunktion vil høre under en type af funktioner, der kaldes *konvekse funktioner*:

Definition 4.7 Konveks funktion

En funktion $f : \mathbb{R}^n \to \mathbb{R}$ kaldes *konveks*, hvis den opfylder, at

$$f(\lambda\vec{x} + (1-\lambda)\vec{y}) \leq \lambda f(\vec{x}) + (1-\lambda)f(\vec{y})$$

for alle $\vec{x}, \vec{y} \in \mathbb{R}^n$ og for alle $\lambda \in [0,1]$ [2, s. 15].



Figur 4.7: En konveks funktion.

Den overstående definition er illustreret i figur 4.7. Bemærk at objektfunktionen i et lineært programmeringsproblem nødvendigvis må være en konveks funktion, da det for alle par af vektorer i rummet, og for alle reelle λ gælder, at

$$\vec{c} \cdot (\lambda\vec{x} + (1-\lambda)\vec{y}) = \lambda\vec{c} \cdot \vec{x} + (1-\lambda)\vec{c} \cdot \vec{y}.$$

Dette er tilfældet, da objektfunktionen er en lineær funktion, og derfor opfylder linearitetsbetingelserne.

Lineære programmeringsproblemer er altså stillet ved at optimere en konveks funktion, defineret på en konveks mængde. Lineære programmeringsproblemer indgår derfor i en klasse af problemer kaldet de konvekse programmeringsproblemer [7, s. 147]. Et fundamentalt resultat inden for konveks programmering er, at ethvert *lokalt minimum* nødvendigvis må være et *globalt minimum*. Disse begreber defineres herunder:

Definition 4.8 Globalt minimum

Vektoren $\vec{x} \in C$ kaldes et *globalt minimum* af funktionen $f : C \to \mathbb{R}$, hvis:

$$\forall \vec{y} \in C \ : \ f(\vec{x}) \leq f(\vec{y}) \text{ [2, s. 15].}$$

Hvor et globalt minimum kræver en betingelse opfyldt for hele definitionsmængden, gælder et lokalt minimum kun for en *omegn* af $\vec{x}$ i definitionsmængden. Vi vælger at definere denne omegn, som værende fællesmængden mellem definitionsmængden og den *åbne kugle*:

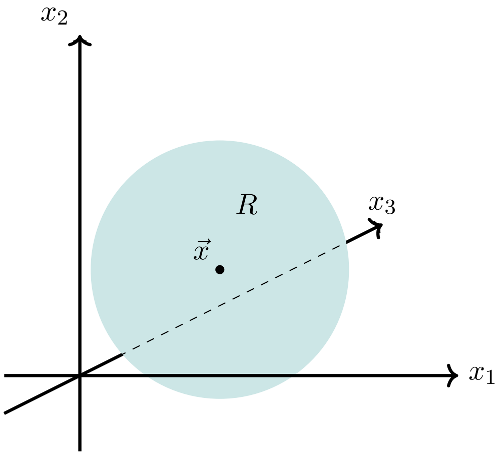
Definition 4.9 Åbne kugler

Lad $\vec{x}$ være en vektor i $\mathbb{R}^n$ og ε en positiv reel skalar. Mængden

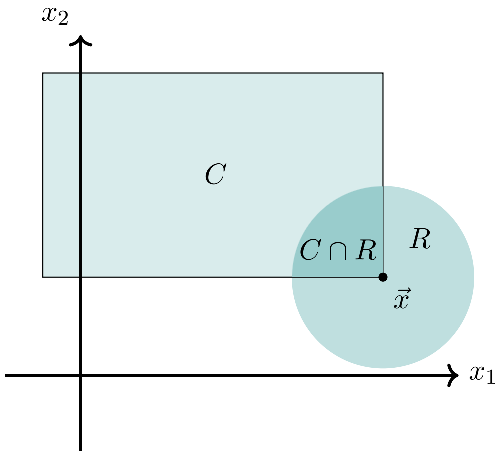
$$R(\vec{x}, \varepsilon) = \{\vec{y} \in \mathbb{R}^n \mid ||\vec{x} - \vec{y}|| < \varepsilon\}$$

kaldes en *åben kugle* om $\vec{x}$ i $\mathbb{R}^n$ [8, s. 90].

Den overstående definition ses illustreret i figur 4.8. Figur 4.9 viser da vores definition af en omegn af en vektor i definitionsmængden.



Figur 4.8: Mængden R er en åben kugle om $\vec{x}$ i $\mathbb{R}^3$.



Figur 4.9: Fællesmængden af den åbne kugle R om $\vec{x} \in \mathbb{R}^2$ og definitionsmængden C. Alle vektorer i denne fællesmængde, befinder sig i en omegn af $\vec{x}$.

Definition 4.10 Lokalt minimum

Vektoren $\vec{x} \in C$ kaldes et *lokalt minimum* af funktionen $f : C \to \mathbb{R}$, hvis

$$\exists \varepsilon > 0 \, \forall \vec{y} \in C \cap R(\vec{x}, \varepsilon) \ : \ f(\vec{x}) \leq f(\vec{y}) \text{ [2, s. 15].}$$

Vi vil nu formulere det førnævnte resultat og bevise det:

Sætning 4.11

Lad $f : C \to \mathbb{R}$ være en konveks funktion defineret på en konveks mængde C. Lad derudover $\vec{x} \in C$ være et lokalt minimum for f på C. Da er $\vec{x}$ også et globalt minimum for f på C [7, s. 147].

Bevis:
Da $\vec{x}$ er et lokalt minimum for f på mængden C, vil der eksistere et $\varepsilon > 0$ sådan at $\forall \vec{y} \in C \cap R(\vec{x}, \varepsilon) : f(\vec{x}) \leq f(\vec{y})$. Lad nu $\vec{z} \in C$ opfylde $\vec{z} \neq \vec{x}$. Vi vil da gerne vise, at $f(\vec{z}) \geq f(\vec{x})$. Ved et tilpas lille $\lambda \in \,]0, 1]$ vil der gælde, at $\lambda \vec{z} + (1 - \lambda)\vec{x} \in R(\vec{x}, \varepsilon) \cap C$. Der gælder da, at $f(\vec{x}) \leq f(\lambda \vec{z} + (1 - \lambda)\vec{x})$. Da funktionen er konveks kan vi ifølge definition 4.7 udlede, at

$$f(\vec{x}) \leq f(\lambda \vec{z} + (1 - \lambda)\vec{x}) \leq \lambda f(\vec{z}) + (1 - \lambda) f(\vec{x}).$$

Dette kan omskrives til:

$$\begin{gathered} f(\vec{x}) - (1 - \lambda) f(\vec{x}) \leq \lambda f(\vec{z}) \Leftrightarrow \\ f(\vec{x}) - (f(\vec{x}) - \lambda f(\vec{x})) \leq \lambda f(\vec{z}) \Leftrightarrow \\ \lambda f(\vec{x}) \leq \lambda f(\vec{z}) \Leftrightarrow \\ f(\vec{x}) \leq f(\vec{z}). \end{gathered}$$

For ethvert $\vec{z}$ gælder altså, at $f(\vec{x}) \leq f(\vec{z})$. Dermed er $\vec{x}$ også et globalt minimum for f på mængden C [7, s. 147]. ∎

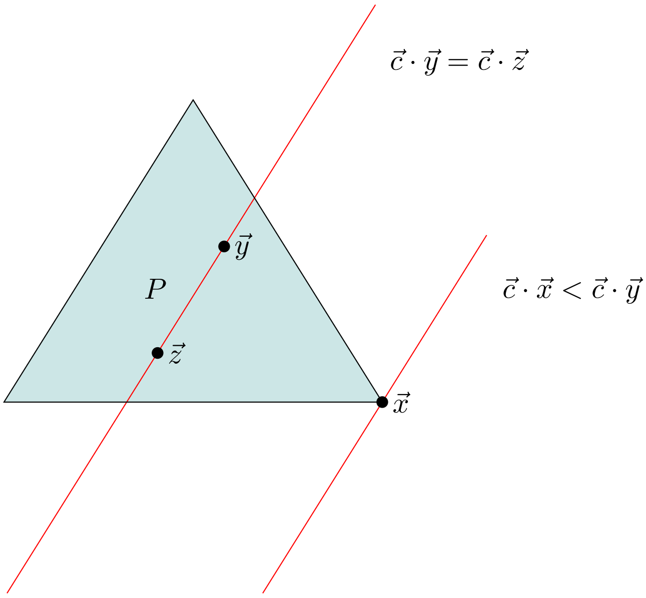
Som konsekvens af dette resultat, har vi, at hvis vi finder en løsning i et lineært minimeringsproblem, der tilsyneladende er et minima i dens omegn, så kan vi stoppe vores søgen efter en bedre løsning. Det skyldes, at denne løsning også vil være et globalt minimum og derfor den optimale løsning. Det er altså betydeligt nemmere at finde den optimale løsning for konvekse funktioner end for ikke-konvekse funktioner, hvor ovenstående ikke nødvendigvis er gældende.

4.2 Kritiske punkter, hjørnepunkter og basale mulige løsninger

Det vil vise sig, at den optimale løsning til et lineært programmeringsproblem, der har en endelig optimal værdi, vil befinde sig i polyederets *hjørnepunkter*. Vi giver derfor nu en definition af et hjørnepunkt, som værende den entydige optimale løsning til et specifikt lineært programmeringsproblem, med mulighedsmængden afgrænset af polyederet, se figur 4.10.

Definition 4.12 Hjørnepunkter

Lad P være et polyeder. En vektor $\vec{x} \in P$ kaldes et *hjørnepunkt* i P, hvis der eksisterer en vektor $\vec{c}$, således at $\vec{c} \cdot \vec{x} < \vec{c} \cdot \vec{y}$ for alle $\vec{y} \in P$, hvor $\vec{x} \neq \vec{y}$ [2, s. 47].



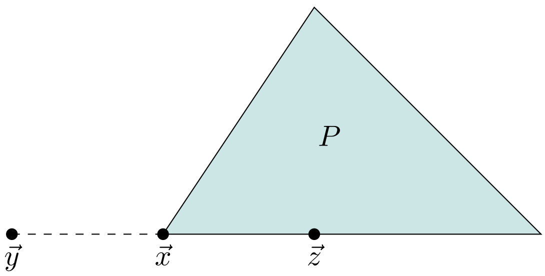
Figur 4.10: Vektor $\vec{x}$ er et hjørnepunkt

Vi vil afslutte dette afsnit med at vise, at et polyeders hjørnepunkter er ækvivalente med dets *kritiske punkter*. At en vektor i et polyeder kaldes et kritisk punkt defineres herunder:

Definition 4.13 Kritiske punkter

Lad P være et polyeder og $\vec{x} \in P$. Vektoren $\vec{x}$ kaldes et *kritisk punkt*, hvis der *ikke* findes en reel skalar $\lambda \in [0, 1]$ og to vektorer $\vec{y}, \vec{z} \in P$, hvor $\vec{y}, \vec{z} \neq \vec{x}$, således $\vec{x} = \lambda\vec{y} + (1 - \lambda)\vec{z}$ opfyldes [2, s. 46].

En vektor i polyederet er altså et kritisk punkt, hvis den ikke kan udtrykkes som en konveks kombination af to distinkte vektorer i polyederet. Et eksempel på dette kan betragtes i figur 4.11. Omvendt betyder dette selvfølgelig, at et ikke-kritisk punkt kan gives som en konveks kombination af to distinkte vektorer i polyederet. Det vil senere vise sig at være smart at sammenligne hjørnepunkter med kritiske punkter, da det vil gøre os i stand til at bevise optimaliteten af et hjørnepunkt ved at bevise optimaliteten af et kritiske punkt.



Figur 4.11: Vi kan ikke udtrykke $\vec{x}$ som en konveks kombination af to vektorer - begge indeholdt i P. Derimod kan $\vec{x}$ udtrykkes som en konveks kombination af $\vec{y}$ og $\vec{z}$, hvor $\vec{y} \notin P$. Vektoren $\vec{x}$ er derfor et kritisk punkt.

Vi har nu givet to geometriske definitioner af et polyeders hjørnepunkter. I praksis kan vi gennemskue, at det ville være svært at finde alle de vektorer, der opfylder disse definitioner. Vi vælger derfor at opbygge en algebraisk forståelse af, hvad det vil betyde, at en vektor er et hjørnepunkt i et polyeder.

Lad et polyeder P være en egentlig delmængde af vektorrummet $\mathbb{R}^n$, og lad det være opbygget af lineære begrænsninger bestående af både ligheder og uligheder. I det forrige kapitel lavede vi en inddeling af disse begrænsninger i tre disjunkte indeksmængder; M_1, M_2 og M_3. Ved at anskue figur 4.11 endnu engang, ser vi, at et hjørnepunkt er placeret der, hvor randen af flere forskellige halvrum mødes. En vektor, der er et hjørnepunkt, vil altså opfylde en lighed i nogle af de givne begrænsninger. Dette fænomen navngiver vi i følgende definition:

Definition 4.14 Aktive begrænsninger

Lad $\vec{x}$ være en vektor i $\mathbb{R}^n$ og $P \subset \mathbb{R}^n$ være et polyeder opbygget af lineære begrænsninger. Hvis $\vec{a}_i \cdot \vec{x} = b_i$, for et i tilhørende en af indeksmængderne M_1, M_2 eller M_3, er den tilhørende begrænsning *aktiv* i $\vec{x}$ [2, s. 48].

Vi bemærker her, at en vektor der opfylder, at n begrænsninger er aktive, også vil være en løsning til et ligningsystem af n ligninger med n ubekendte. Vi kan derfor formulere og udlede følgende resultat om en sådan vektor:

Sætning 4.15

Lad $\vec{x} \in \mathbb{R}^n$ og lad $I = \{i \mid \vec{a}_i \cdot \vec{x} = b_i\}$ være mængden af indeks af de begrænsninger, som er aktive i $\vec{x}$. Da er følgende ækvivalente:

(a) Der er n vektorer i mængden $\{\vec{a}_i \mid i \in I\}$ der er lineært uafhængige;

(b) Vektorerne $\vec{a}_i, i \in I$ udspænder hele $\mathbb{R}^n$;

(c) Ligningssystemet $\vec{a}_i \cdot \vec{x} = b_i, i \in I$, har en entydig løsning [2, s. 48].

Bevis:
Vi vil nu vise, at (a) $\Leftrightarrow$ (b).
Antag at vektorerne $\vec{a}_i, i \in I$ udspænder hele $\mathbb{R}^n$. Vi ved, at n af disse vektorer er en basis for $\mathbb{R}^n$, hvilket betyder at disse er lineært uafhængige. Omvendt antages at n af vektorerne $\vec{a}_i$ er lineært uafhængige. Da danner de en basis for et n-dimensionalt underrum, hvilket er hele $\mathbb{R}^n$. Dette betyder at (a) $\Leftrightarrow$ (b).

Vi vil nu vise, at (b) $\Leftrightarrow$ (c).
Antag at ligningssystemet $\vec{a}_i \cdot \vec{x} = b_i, i \in I$, har flere løsninger. Lad $\vec{y}$ og $\vec{z}$ være to af disse. Vektoren $\vec{d} = \vec{y} - \vec{z}$ opfylder da, at $\vec{a}_i \cdot \vec{d} = 0$ for alle $i \in I$ og $\vec{d} \neq \vec{0}$, fordi

$$\begin{aligned}\vec{a}_i \cdot \vec{d} &= \vec{a}_i \cdot (\vec{y} - \vec{z}) \\ &= \vec{a}_i \cdot \vec{y} - \vec{a}_i \cdot \vec{z} \\ &= b_i - b_i = 0.\end{aligned}$$

Vi ser, at $\vec{d}$ befinder sig i et ikke-trivielt nulrum for en matrix med rækkerne $\vec{a}_i, i \in I$. Rækkerne i matricen er da lineært afhængige, hvilket betyder, at $\{\vec{a}_i \mid i \in I\}$ ikke udspænder hele $\mathbb{R}^n$. Det er nu vist ved kontraposition at (b) $\Rightarrow$ (c).

Antag nu at vektorerne $\vec{a}_i, i \in I$ ikke udspænder hele $\mathbb{R}^n$. Da ved vi, at matricen A, med $\vec{a}_i, i \in I$ som rækker, opfylder at $\text{rank}(A) \leq n - 1$. Derfor ved vi også, at matricen vil have et ikke-trivielt nulrum ifølge sætning 2.22.

Der findes da en vektor $\vec{0} \neq \vec{d} \in \text{Null}(A)$. Hvis $\vec{x}$ er en løsning til $\vec{a}_i \cdot \vec{x} = b_i, i \in I$, er vektoren $\vec{x} + \vec{d}$ også en løsning

$$\begin{aligned}\vec{a}_i \cdot (\vec{x} + \vec{d}) &= \vec{a}_i \cdot \vec{x} + \vec{a}_i \cdot \vec{d} \\ &= b_i + 0 = b_i.\end{aligned}$$

Det betyder, at der eksisterer flere løsninger til ligningsystemet. Det er nu vist ved kontraposition at (c) $\Rightarrow$ (b) og vi kan derfor konkludere at at (b) $\Leftrightarrow$ (c) [2, s. 48-49]. ■

Givet dette resultat ved vi nu, at hvis en løsning $\vec{x} \in \mathbb{R}^n$ har n lineært uafhængige aktive begrænsninger, så vil den være entydig. Vi giver derfor nu en specifik definition af disse entydige vektorer:

Definition 4.16 Basale løsninger og basale mulige løsninger

Lad $P \subset \mathbb{R}^n$ være et polyeder opbygget af lineære begrænsninger og lad $\vec{x}$ være en vektor i $\mathbb{R}^n$.

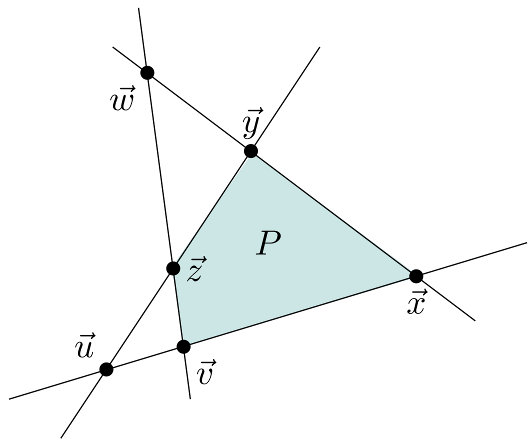
(a) Vektoren $\vec{x}$ er en *basal løsning*, hvis den opfylder at;

 (i) Alle begrænsninger med ligheder er aktive.

 (ii) Ud af de aktive begrænsninger ved $\vec{x}$ er n af dem lineært uafhængige.

(b) Hvis $\vec{x}$ er en basal løsning og overholder alle begrænsningerne, kaldes denne for en *basal mulig løsning* [2, s. 50].

Man kunne forestille sig, at basale løsninger ville kunne repræsentere et hjørnepunkt i et polyeder, da de er givet entydigt i et punkt, hvor randen af flere halvrum mødes. Vi kan dog betragte figur 4.12 og indse, at en sådan vektor ikke nødvendigvis behøver at være indenfor polyederet. Vi bliver derfor nød til at indskrænke os til basale løsninger, der befinder sig i polyederet - altså basale mulige løsninger.



Figur 4.12: Overstående figur viser et polyeder i $\mathbb{R}^2$. Det ses, at der er to lineært uafhængige aktive begrænsninger i alle punkterne. Vektorerne $\vec{w}$ og $\vec{u}$ er derfor basale løsninger, men da de ikke overholder alle begrænsningerne, men blot lighederne, ligger de uden for polyederet P og er derfor ikke basale mulig løsninger.

Indtil videre har vi blot antaget, at basale mulige løsninger er en god algebraisk definition af et hjørnepunkt, og ligeledes for de kritiske punkter. Vi vil derfor nu vise, at dette er tilfældet, og at de tre begreber er ækvivalente:

Sætning 4.17

Lad P være et ikke-tomt polyeder i $\mathbb{R}^n$, og lad $\vec{x}$ være en vektor i P. Følgende er da ækvivalent:

(a) $\vec{x}$ er et hjørnepunkt;

(b) $\vec{x}$ er et kritisk punkt;

(c) $\vec{x}$ er en basal mulig løsning [2, s. 50].

Bevis:
Antag at P er et polyeder, som er opbygget af begrænsninger på formen $\vec{a}_i \cdot \vec{x} \geq b_i$ og $\vec{a}_i \cdot \vec{x} = b_i$.

Vi vil nu bevise, at (a) $\Rightarrow$ (b). Antag at $\vec{x} \in P$ er et hjørnepunkt. For hjørnepunkter findes der en vektor $\vec{c} \in \mathbb{R}^n$, som opfylder at $\vec{c} \cdot \vec{x} < \vec{c} \cdot \vec{y}$ for alle $\vec{y} \in P$, hvor $\vec{y} \neq \vec{x}$.
Hvis der findes en vektor $\vec{y}$ og en vektor $\vec{z}$, der yderligere opfylder, at $\vec{y}, \vec{z} \in P$, hvor $\vec{y}, \vec{z} \neq \vec{x}$. Det vil da gælde, at

$$\vec{c} \cdot \vec{x} < \vec{c} \cdot \vec{y} \quad \text{og} \quad \vec{c} \cdot \vec{x} < \vec{c} \cdot \vec{z}.$$

Da begge vektorer tilhører P, kan vi for alle $\lambda \in [0,1]$ opskrive uligheden

$$\vec{c} \cdot \vec{x} < \vec{c} \cdot (\lambda \vec{y} + (1 - \lambda)\vec{z}).$$

Denne ulighed viser altså, at $\vec{x} \neq \lambda \vec{y} + (1 - \lambda)\vec{z}$. Vi kan derfor konkludere, at $\vec{x}$ altså ikke kan udtrykkes som en konveks kombination og er derfor et kritisk punkt ifølge definition 4.13. Dette medfører, at (a) $\Rightarrow$ (b).

Vi vil nu vise, at (b) $\Rightarrow$ (c). Antag at $\vec{x} \in P$ ikke er en basal mulig løsning. Lad I være indeksmængden af aktive begrænsninger i $\vec{x}$, altså $I = \{i \mid \vec{a}_i \cdot \vec{x} = b_i\}$.

Da $\vec{x}$ ikke er en basal mulig løsning, eksisterer der ikke n lineært uafhængige vektorer i mængden $\{\vec{a}_i \mid i \in I\}$. Derfor vil vektorerne $\vec{a}_i, i \in I$ udspænde et ikke-trivielt underrum af $\mathbb{R}^n$, hvilket betyder, at nulrummmet er ikke-trivielt. Derfor eksisterer der en vektor $\vec{d} \neq \vec{0}$, som opfylder $\vec{a}_i \cdot \vec{d} = 0$ for $i \in I$.

Lad $\varepsilon > 0$ være lille nok, så følgende vektorer $\vec{y} = \vec{x} + \varepsilon\vec{d}$ og $\vec{z} = \vec{x} - \varepsilon\vec{d}$ er indeholdt i polyederet. Det gælder da, når $i \in I$, at

$$\vec{a}_i \cdot \vec{y} = \vec{a}_i \cdot (\vec{x} + \varepsilon\vec{d}) = \vec{a}_i \cdot \vec{x} + \vec{a}_i \cdot \varepsilon\vec{d} = \vec{a}_i \cdot \vec{x} = b_i.$$

Lignende er gældende for $\vec{z}$.
Modsat gælder det, når $i \notin I$, så er $\vec{a}_i \cdot \vec{x} > b_i$ og

$$\vec{a}_i \cdot \vec{y} = \vec{a}_i \cdot \vec{x} + \varepsilon\vec{a}_i \cdot \vec{d} > b_i.$$

Lignende gælder for $\vec{z}$. Vektorerne $\vec{y}$ og $\vec{z}$ er da mulige løsninger. Bemærk da, at $\vec{x}$ kan udtrykkes som en konveks kombination af $\vec{y} = \vec{x} + \varepsilon\vec{d}$ og $\vec{z} = \vec{x} - \varepsilon\vec{d}$, da

$$\begin{aligned}
&\vec{y} + \vec{z} = \vec{x} + \varepsilon\vec{d} + \vec{x} - \varepsilon\vec{d} \Leftrightarrow \\
&\vec{y} + \vec{z} = 2\vec{x} \Leftrightarrow \\
&\frac{\vec{y} + \vec{z}}{2} = \vec{x}.
\end{aligned}$$

Vektoren $\vec{x}$ er altså *ikke* et kritisk punkt og dermed er det bevist, at et kritisk punkt er en basal mulig løsning. Dette medfører, at (b) $\Rightarrow$ (c).

Vi vil nu vise, at (c) $\Rightarrow$ (a). Lad $\vec{x}$ være en basal mulig løsning og $I = \{i \mid \vec{a}_i \cdot \vec{x} = b_i\}$. Lad nu $\vec{c} = \sum_{i \in I} \vec{a}_i$. Vi ser da, at

$$\vec{c} \cdot \vec{x} = \sum_{i \in I} \vec{a}_i \cdot \vec{x} = \sum_{i \in I} b_i.$$

For alle andre $\vec{y} \in P$ gælder det, at

$$\vec{c} \cdot \vec{y} = \sum_{i \in I} \vec{a}_i \cdot \vec{y} \geq \sum_{i \in I} b_i.$$

Ud fra dette kan vi se, at $\vec{c} \cdot \vec{x} \leq \vec{c} \cdot \vec{y}$, hvilket viser, at $\vec{x}$ er den optimale løsning. Da $\vec{x}$ antages at være en basal mulig løsning, vil der være n lineært uafhængige begrænsninger, hvilket medfører, at $\vec{x}$ også er en entydig løsning. Uligheden $\vec{c} \cdot \vec{x} < \vec{c} \cdot \vec{y}$ er dermed opfyldt og dette betyder, at $\vec{x}$ også er et hjørnepunkt. Dette medfører, at $(c) \Rightarrow (a)$ [2, s. 51-52]. ■

Vi har nu bevist, at vores algebraiske definition af et hjørnepunkt er brugbar og derved også bevist, at de kritiske punkter er entydige optimale løsninger til specifikke lineære programmeringsproblemer. Vi observerer dog, som konsekvens af definition 4.16 af basale løsninger, at et lineært programmeringsproblem stillet ved m lineære begrænsninger for n variable, ikke har nogle basale mulige løsninger, hvis $m < n$, idet n af begrænsningerne skal være lineært uafhængige.

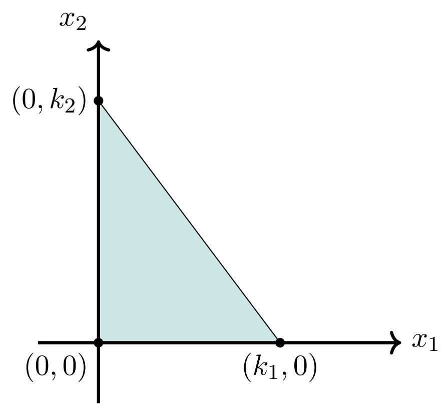
En anden detalje vedrørende de basale mulige løsninger, herunder hjørnepunkterne og de kritiske punkter, er, at der er et endeligt antal af dem i et lineært programmeringsproblem stillet ved et endeligt antal af lineære begrænsninger:

Korollar 4.18

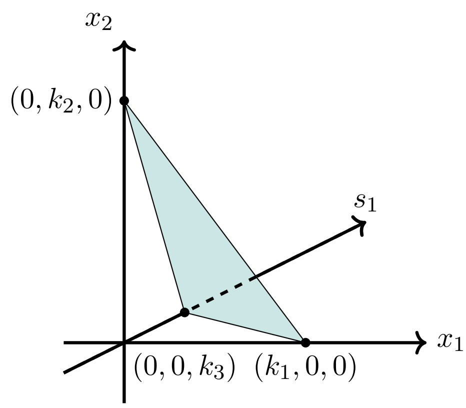
Givet et endeligt antal lineære begrænsninger af uligheder, eksisterer der kun et endeligt antal basale løsninger og basale mulige løsninger [2, s. 52].

Bevis:
Givet et polyeder i $\mathbb{R}^n$ begrænset af m lineære uligheder, vil der til enhver basal løsning være n lineært uafhængige aktive begrænsninger - se definition 4.16. Forskellige delmængder, med n lineært uafhængige aktive begrænsninger, opfyldes af forskellige entydige basale løsninger. Det medfører, at antallet af basale løsninger afgrænses opadtil af antallet af delmængder med n lineært uafhængige uligheder. Da der er et endeligt antal uligheder, må antallet af basale- og basale mulige løsninger være endeligt [2, s. 52]. ■

Vi vil nu afslutte dette afsnit med at anskueliggøre, at et lineært programmeringsproblem omskrevet til standardform har samme hjørnepunkter som det originale problem. Betragt mulighedsmængden til et lineært programmeringsproblem med én hovedbegrænsning bestående af to variable underlagt den ikke-negative begrænsning, som vist i figur 4.13. Her ser vi, at de basale mulige løsninger er hjørnepunkterne i mængden $E_0 = \{(k_1, 0), (0, k_2), (0, 0)\}$.
Vi kan se på figuren, at i dette lineære programmeringsproblem er begrænsningens indeks enten indeholdt i M_1 eller M_2. Derfor skal vi enten tilføje en overskuds- eller underskudsvariabel s_1, for at kunne omskrive problemet til standardform. Dette tilføjer altså en ekstra variabel og øger dimensionen i problemet. Det samme problem omskrevet til standardform kan dermed ses i figur 4.14. Her ser vi, at de basale mulige løsninger er hjørnepunkterne i mængden $E_1 = \{(k_1, 0, 0), (0, k_2, 0), (0, 0, k_3)\}$.



Figur 4.13: Optimeringsproblem med to variable og en begrænsning.



Figur 4.14: Optimeringsproblem med to variable og en begrænsning på standardform.

Hvis vi betragter de to ovenstående figurer, ser vi, at vi kan lave en bijektiv funktion $f : E_1 \to E_0$, $f((x_1, x_2, s_1)) = (x_1, x_2)$, som afbilder hjørnepunkterne fra problemet på standardforms mulighedsmængde over i hjørnepunkterne i det originale problems mulighedsmængde. Vi ser, at objektfunktionen for problemet på standardform Z_1 har samme værdi i et hjørnepunkt $\vec{x}$, som objektfunktionen for det originale problem Z_0 i hjørnepunktet $f(\vec{x})$. Dette er tilfældet, da overskuds- og underskudsvariable ikke bidrager til objektfunktionens værdi.

4.3 Polyeder på standardform

Indtil nu har vi betragtet polyedre på formen som et standard lineært minimeringsproblem, altså $P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} \geq \vec{b}\}$. Da vi i det forrige kapitel fandt ud af, at alle lineære programmeringsproblemer kan skrives på standardform, ønsker vi nu at generaliserer til polyedre på standardform. At et polyeder siges at være på standardform, defineres herunder:

Definition 4.19 Polyedre på standardform

Lad A være en $m \times n$-matrix og $\vec{b}$ en vektor i $\mathbb{R}^m$. Mængden

$$P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{b}, \vec{x} \geq \vec{0}\}$$

kaldes da et *polyeder på standardform* [2, s. 53].

Bemærk at hvis P er et ikke-tomt polyeder på standardform, så opfylder enhver $\vec{x} \in P$ et ligningssytem af m ligninger med n variable, givet hovedbegrænsningerne i matrix A. Da vi er interesserede i at finde polyederets basale mulige løsninger, husker vi, at hvis $\vec{x}$ skal være en basal løsning, så skal den have n lineært uafhængige aktive begrænsninger og opfylde alle ligheder.

Ved at antage at rækkerne i A er lineært uafhængige, har enhver løsning m lineært uafhængige aktive begrænsninger givet ved hovedbegrænsningerne $A\vec{x} = \vec{b}$. Vi vil senere vise, at denne antagelse altid vil gælde for et ikke-tomt polyeder på standardform. Se sætning 4.21.
Denne antagelse vil dog også betyde, at $m \leq n$. I tilfælde, hvor $m < n$, vil hovedbegrænsningerne ikke give os tilstrækkeligt mange lineært uafhængige aktive begrænsninger. For da at få n lineært uafhængige aktive begrænsninger, skal vi vælge at gøre $n - m$ af de ikke-negative begrænsninger aktive. Dette betyder, at disse bliver ligheder og dermed lig nul. Valget af de $n - m$ ikke-negative begrænsninger viser sig dog at være bestemt:

Sætning 4.20

Betragt begrænsningerne $A\vec{x} = \vec{b}$ og $\vec{x} \geq \vec{0}$ og antag at $m \times n$-matricen A har lineært uafhængige rækker. En vektor $\vec{x}$ i $\mathbb{R}^n$ er en basal løsning, hvis og kun hvis $A\vec{x} = \vec{b}$, og der eksisterer indeks $B(1), \ldots, B(m)$ sådan at:

(a) Søjlevektorerne $\vec{A}_{B(1)}, \ldots, \vec{A}_{B(m)}$ er lineært uafhængige.

(b) $x_i = 0$ for $i \neq B(1), \ldots, B(m)$ [2, s. 53].

Bevis:
Vi vil første vise, at hvis $A\vec{x} = \vec{b}$ og at (a) og (b) er gældende, da er $\vec{x}$ en basal løsning.
Betragt en vektor $\vec{x} \in \mathbb{R}^n$ og antag at indeksene $B(1), \dots, B(m)$ opfylder at søjlerne $\vec{A}_{B(1)}, \dots, \vec{A}_{B(m)}$ er lineært uafhængige og at $x_i = 0$ for $i \neq B(1), \dots, B(m)$. Ligningsystemet $A\vec{x} = \vec{b}$ kan da omskrives så

$$\sum_{i=1}^{n} \vec{A}_i x_i = \sum_{i=1}^{m} \vec{A}_{B(i)} x_{B(i)} = \vec{b}.$$

Da søjlerne $\vec{A}_i, i = B(1), \dots, B(m)$ er lineært uafhængige, vil variablene $x_{B(1)}, \dots, x_{B(m)}$ være entydige. Ligningssystemet, bestående af $A\vec{x} = \vec{b}$ og de ikke negative begrænsninger $x_i = 0, i \neq B(1), \dots, B(m)$, vil derfor have en entydig løsning. Ud fra sætning 4.15 ved vi da, at der er n lineært uafhængige aktive begrænsninger, og at $\vec{x}$ derfor er en basal løsning.

Vi vil nu vise, at hvis $\vec{x}$ er en basal løsning, da er $A\vec{x} = \vec{b}$ og kriterierne opfyldt.
Lad $\vec{x}$ være en basal løsning, og lad komponenterne $x_{B(1)}, \dots, x_{B(k)} > 0$. Da $\vec{x}$ er en basal løsning vil ligningssystemet bestående af de aktive begrænsninger $A\vec{x} = \vec{b}$ og $x_i = 0, i \neq B(1), \dots, B(k)$ have en entydig løsning. Da er søjlerne $\vec{A}_{B(1)}, \dots, \vec{A}_{B(k)}$ lineært uafhængige. Da matricen A har m rækker, ved vi, at $k \leq m$. Da rækkerne i A er lineært uafhængige, ved vi at A har m lineært uafhængige søjler, som udspænder $\mathbb{R}^m$. Vi kan derfor finde $m - k$ søjler $\vec{A}_{B(k+1)}, \dots, \vec{A}_{B(m)}$, sådan at søjlerne $\vec{A}_i, i = B(1), \dots, B(m)$ er lineært uafhængige. Da $k \leq m$ har vi, at når $i \neq B(1), \dots, B(m) \Rightarrow i \neq B(1), \dots, B(k)$ og at $x_i = 0$ når $i \neq B(1), \dots, B(m)$. Vi ser, at (a) og (b) er opfyldt [2, s. 53]. ■

Hvor vi før, for generelle polyedre, skulle tjekke om en vektor var en basal løsning, kan vi nu, ved at generelisere til polyedre på standardform, konstruere basale løsninger ud fra overstående resultat. Metoden til dette er som følgende:

1. Vælg m lineært uafhængige søjler; $\vec{A}_{B(1)}, \dots, \vec{A}_{B(m)}$.
2. Lad $x_i = 0$ for alle $i \neq B(1), \dots, B(m)$.
3. Løs ligningssystemet $A\vec{x} = \vec{b}$ for de resterende variable; $x_{B(1)}, \dots, x_{B(m)}$ [2, s. 54].

Vi husker, at enhver basal mulig løsning også er en basal løsning. Da enhver basal løsning kan konstrueres ud fra ovenstående metode, givet resultatet i sætning 4.20, kan enhver basal mulig løsning også konstrueres fra ovenstående metode. Da vi betragter polyedre på standarform, gør den ikke-negative begrænsning sig gældende. Vi skal derfor blot tjekke, om en given basal løsning, konstrueret ud fra ovenstående metode, er ikke-negativ, da den da vil være en basal mulig løsning.

Denne metode til at finde basale mulige løsninger, gør sig gældende til konstruktionen af Simplex-metoden i kapitel 5. Vi introducerer derfor nu nogle nyttige begreber. De valgte lineært uafhængige søjler $\vec{A}_{B(1)}, \dots, \vec{A}_{B(m)}$, kaldes de *basale søjler*, hvorimod de resterende søjler i A kaldes de *ikke-basale søjler*. Lignende kalder vi $x_{B(1)}, \dots, x_{B(m)}$ de *basale variable*, og de resterende, de *ikke-basale variable*. Indeksmængden $\{B(1), \dots, B(m)\}$ kalder vi derfor løsningens *base*, hvis elementer er de *basale indeks*.

Da de basale søjler er lineært uafhængige, kan vi opstille en invertibel $m \times m$-matrix $B = [\vec{A}_{B(1)} \cdots \vec{A}_{B(m)}]$, som vi kalder en *basismatrix*. Tilhørende denne definerer vi *basisvektoren*

$\vec{x}_B = (x_{B(1)}, \ldots, x_{B(m)})^T$. Vi kan forsimple beregningerne i overstående metode, ved at betragte ligningssystemet på følgende måde:

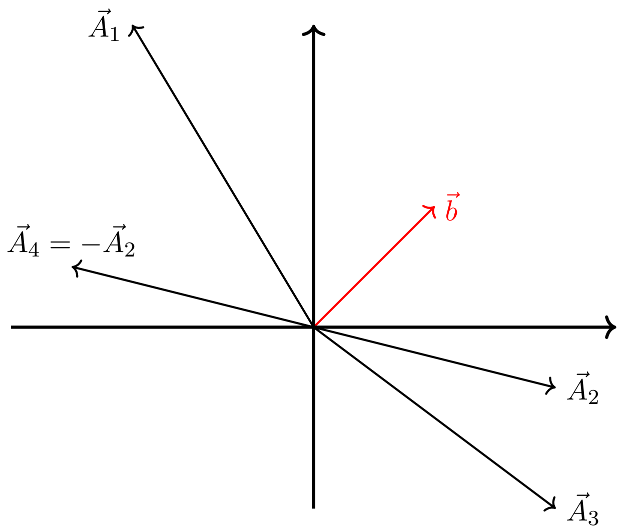
$$\sum_{i=1}^{n} \vec{A}_i x_i = \sum_{i=1}^{m} \vec{A}_{B(i)} x_{B(i)} = B\vec{x}_B = \vec{b}.$$

Da B er invertibel, kan vi da beregne $\vec{x}_B$ ved at gange den inverse basismatrix på fra venstre:

$$\vec{x}_B = B^{-1}\vec{b}.$$

Det tredje trin i overstående metode finder en linearkombination af de basale søjler, som er lig med vektoren $\vec{b}$. Hvis denne linearkombination kun har positive koefficienter betyder det, at vektoren $\vec{x}_B \geq \vec{0}$, hvilket betyder at den konstruerede løsning er en basal mulig løsning. Modsat betyder dette, at hvis en eller flere af koefficienterne i linearkombinationen er negativ, da vil en eller flere af variablene i $\vec{x}_B$ være negative, hvilket bryder den ikke-negative begrænsning. En sådan løsning vil derfor være en basal løsning udenfor mulighedsmængden. Dette princip er illustreret i figur 4.15.

Bemærk at hvis to indeksmængder, af basale indeks, indeholder de samme elementer men i en forskellig rækkefølge, vil de beskrive den samme løsning. Enhver basal løsning, vil derfor have en entydig indeksmængde af basale indeks, som er løsningens base.



Figur 4.15: Betragt et lineært programmeringsproblem på standardform, givet ved $A\vec{x} = \vec{b}, \vec{x} \geq \vec{0}$, hvor A er en 2×4-matrix med søjlerne $\vec{A}_1, \ldots, \vec{A}_4$. En basal løsning vil da være en linearkombination af søjlerne tilhørende dens base, som konstruerer $\vec{b}$. Basen $\{1, 2\}$ vil svare til en basal mulig løsning, da vektoren $\vec{b}$ kan konstrueres ved en positiv linearkombination af de basale søjler $\vec{A}_1$ og $\vec{A}_2$. Derimod vil basen $\{2, 3\}$ konstruere en basal løsning uden for mulighedsmængden, da konstruktionen af $\vec{b}$ vil kræve en negativ koefficient i linearkombinationen af de basale søjler. Indeksmængden $\{4, 2\}$ er ikke en base, da $\vec{A}_4$ og $\vec{A}_2$ er lineært afhængige.

4.3.1 Antagelse om lineært uafhængige rækker

Vi afslutter nu afsnittet med at bevise, at vores tidligere antagelse - at rækkerne i A er lineært uafhængige - altid er gældende i problemer på standardform:

Sætning 4.21

Lad $P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{b}, \vec{x} \geq \vec{0}\}$ være et ikke-tomt polyeder på standardform, hvor A er en $m \times n$-matrix, med rækkerne $\vec{a}_i, i = 1, \dots, m$. Antag $\text{rank}(A) = k < m$ og at rækkerne $\vec{a}_{i_j}, j = 1, \dots, k$ er lineært uafhængige. Da vil polyederet

$$F = \{\vec{x} \in \mathbb{R}^n \mid \vec{a}_{i_1} \cdot \vec{x} = \vec{b}_{i_1}, \dots, \vec{a}_{i_k} \cdot \vec{x} = \vec{b}_{i_k}, \vec{x} \geq \vec{0}\}$$

være det samme polyeder, som P [2, s. 57].

Bevis:
Vi beviser resultatet i tilfældet, hvor de første k rækker er lineært uafhængige. Dette tilfælde kan altid konstrueres ved rækkeombytning i A og ved at rekonstruere vektoren $\vec{b}$. Vi har da at

$$F = \{\vec{x} \in \mathbb{R}^n \mid \vec{a}_1 \cdot \vec{x} = \vec{b}_1, \dots, \vec{a}_k \cdot \vec{x} = \vec{b}_k, \vec{x} \geq \vec{0}\}.$$

Bemærk at $P \subseteq F$, da ethvert element i P også opfylder begrænsningerne i F. Vi vil nu vise, at det omvendte også gælder.

Vektorerne $\vec{a}_j, j = 1, \dots, k$ udgør en basis for rækkerummet af A. Enhver række i A kan derfor konstrueres af en linearkombination af disse. Det gælder derfor for ethvert element $\vec{x} \in P$, at

$$b_i = \vec{a}_i \cdot \vec{x} = \sum_{j=1}^{k} \lambda_{ij} \vec{a}_j \cdot \vec{x} = \sum_{j=1}^{k} \lambda_{ij} b_j, \; i = 1, \dots, m.$$

Betragt nu et element $\vec{y} \in F$. Bemærk at ethvert element i F opfylder ligningssystemet $\vec{a}_j \cdot \vec{y} = b_j, j = 1, \dots, k$. Vi har da, at

$$\vec{a}_i \cdot \vec{y} = \sum_{j=1}^{k} \lambda_{ij} \vec{a}_j \cdot \vec{y} = \sum_{j=1}^{k} \lambda_{ij} b_j, \; i = 1, \dots, m.$$

Hvilket betyder at ethvert $\vec{y} \in F$ også tilhører P. Vi har derfor at $F \subseteq P$. Derfor må $P = F$ [2, s. 57]. ■

Hvis vi har et lineært programmeringsproblem, hvor vi har lineært afhængige rækker, kan vi, for at fjerne disse rækker, udføre elementære rækkeoperationer på den transponerede matrix A^T, for at nå frem til en matrix på trappeform. Herefter fjerner vi ikke-pivotsøjlerne og transponere den nye matrix, som vi også kalder A. På denne måde laver vi en ny matrix A, hvis *rækker* er lineært uafhængige, men stadig beskriver polyederet til det oprindelige lineære programmeringsproblem, grundet overstående sætning.

Derudover er det væsentligt at undersøge, om det oprindelige problem har en løsning. Dette gøres ved at udføre elementære rækkeoperationer på totalmatricen $[A \mid b\,]$ for at nå frem til, om ligningssystemet er konsistent - se afsnit 2.5.1.

4.4 Degeneration

Basale løsninger i et lineært programmeringsproblem har hver mindst n lineært uafhængige aktive begrænsninger. Dette betyder selvfølgelig, at der kan opstå tilfælde, hvor en basal løsning har flere end n aktive begrænsninger. Det vil senere vise sig, at disse tilfælde vil være problematiske for Simplex-metoden. Vi kalder derfor disse *degenererede* basale løsninger:

Definition 4.22 Degenererede basale løsninger

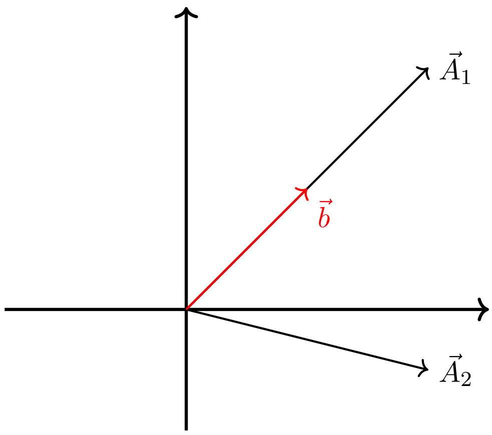
En basal løsning $\vec{x} \in \mathbb{R}^n$ er *degenereret*, hvis mere end n af begrænsningerne er aktive i $\vec{x}$ [2, s. 58].

Vi husker, at enhver basal løsning i et lineært programmeringsproblem på standardform opfylder alle hovedbegrænsninger, og har mindst $n-m$ ikke-negative begrænsninger aktive. For da at få flere end n lineært uafhængige aktive begrænsninger, skal en sådan løsning have flere end $n-m$ ikke-negative begrænsninger aktive. Vi kan derfor give en specifik definition af degenererede basale løsninger i lineære programmeringsproblemer på standardform:

Definition 4.23 Degenererede basale løsninger på standardform

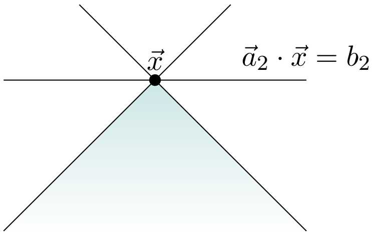
Betragt et polyeder på standardform $P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{b},\ \vec{x} \geq \vec{0}\}$ og lad $\vec{x}$ være en basal løsning. Lad m være antallet af rækker i A. Vektoren $\vec{x}$ er en degenereret basal løsning, hvis mere end $n-m$ af komponenterne i $\vec{x}$ er lig 0 [2, s. 59].

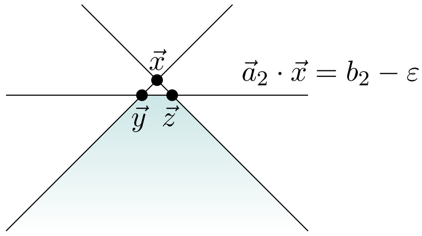
En degenereret basal løsning på standardform vil derfor altid have en eller flere variable lig nul i dens tilhørende vektor $\vec{x}_B$. Dette betyder at en eller flere af de basale søjler ikke anvendes til konstruktionen af vektoren $\vec{b}$. Dette kan betragtes grafisk i figur 4.16.



Figur 4.16: Basen $\{1, 2\}$ vil give en basal løsning, der opfylder, at $A\vec{x} = \vec{b}$. Vektoren $\vec{b}$ kan i dette tilfælde konstrueres ved et multiplum af søjlen $\vec{A}_1$. Vi får da, at $x_2 = 0$, hvilket gør denne basale løsning degenereret.

Det er midlertidigt muligt at undgå degenererede basale løsninger ved at ændre på de overflødige aktive begrænsninger, se figur 4.17. Da vil den basale løsning ikke længere gøre de overflødige begrænsninger aktive, hvilket betyder, at den ikke længere er degenereret.





Figur 4.17: I det lineære programmeringsproblem i (a), ser vi at vektoren $\vec{x}$ er en degenereret basal løsning. Bemærk at begrænsningen $\vec{a}_2 \cdot \vec{x} = b_2$ er en overflødig aktiv begrænsning. Vi kan betragte det samme problem i (b), hvor vi har trukket en meget lille skalar $\varepsilon > 0$ fra den overflødige begrænsnings konstant. Bemærk nu at vektoren $\vec{x}$ er en ikke-degenereret basal løsning. Vi har nu fået to nye basale mulige løsninger $\vec{y}$ og $\vec{z}$, som begge er ikke-degenererede.

Begrænsningerne i et lineært programmeringsproblem er dog ofte givet specifikt, og vi vil derfor ikke være i stand til at justere på dem. Af den grund kan vi forvente at møde degenererede løsninger i reelle problemer.

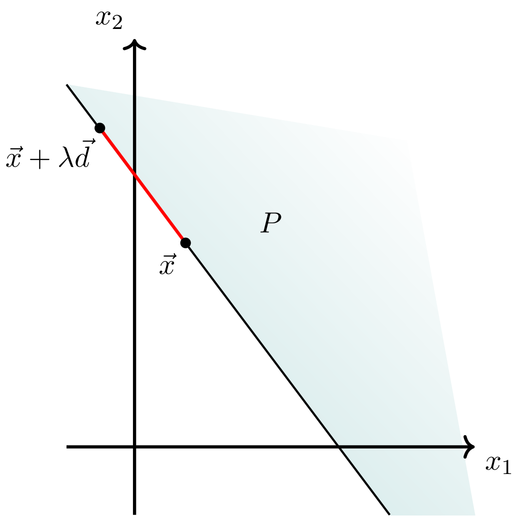
4.5 Eksistens af kritiske punkter

Vi har indtil nu taget udgangspunkt i lineære programmeringsproblemer, der har haft basale mulige løsninger, altså kritiske punkter. Som vi tidligere nævnte i afsnit 4.2, vil der i et lineært programmeringsproblem ikke eksistere nogle basale løsninger, hvis antallet af begrænsninger er mindre end antallet af variable. Som konsekvens af dette, vil der ikke findes nogle basale mulige løsninger - altså kritiske punkter. Vi vil i dette afsnit derfor karakterisere de lineære programmeringsproblemer, hvor dette er tilfældet.

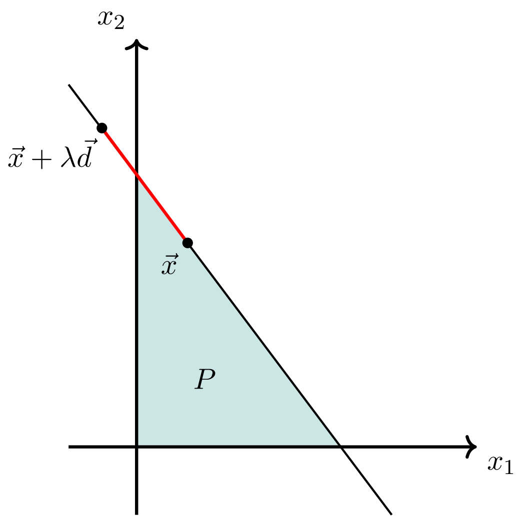
Vi kan beskrive disse tilfælde geometrisk, da det viser sig, at polyederet vil indeholde en *linje*, når der ingen kritiske punkter findes. Dette definerer vi herunder:

Definition 4.24 Linjer

Et polyeder $P \subset \mathbb{R}^n$ indeholder en *linje*, hvis der findes en vektor $\vec{x} \in P$ og en vektor $\vec{d} \in \mathbb{R}^n$ hvor $\vec{d} \neq \vec{0}$, som opfylder at $\vec{x} + \lambda\vec{d} \in P$ for alle reelle λ [2, s. 63].



Figur 4.18: Her tilhører $\vec{x}+\lambda\vec{d}$ polyederet P for alle λ. Polyederet indeholder derfor en linje.



Figur 4.19: Her tilhører $\vec{x}+\lambda\vec{d}$ ikke polyederet P for alle λ. Polyedret indeholder derfor ikke en linje.

Vi kan, ud fra overstående definition, nu bevise vores tidligere påstand om eksistensen af kritiske punkter.

Sætning 4.25

Lad $P = \{\vec{x} \in \mathbb{R}^n \mid \vec{a}_i \cdot \vec{x} \geq b_i, i = 1, \dots, m\}$ være et ikke-tomt polyeder. Følgende er da ækvivalent:

(a) Polyederet P indeholder ikke en linje;

(b) Polyederet P har mindst et kritisk punkt;

(c) Der findes n vektorer $\vec{a}_i$ for $i = 1, \dots, m$, som er lineært uafhængige [2, s. 63].

Bevis:
Først vil vi bevise, at (a) $\Rightarrow$ (b).
Antag at P ikke indeholder en linje. Lad da $\vec{x} \in P$ og lad indeksmængden $I = \{i \,|\, \vec{a}_i \cdot \vec{x} = b_i\}$. Hvis n af vektorerne $\vec{a}_i, i \in I$ er lineært uafhængige, så har vi per definition en basal mulig løsning og dermed eksisterer der et kritisk punkt, givet sætning 4.17.

Hvis dette *ikke* er tilfældet, da ligger vektorerne $\vec{a}_i, i \in I$ i et ikke-trivielt underrum af $\mathbb{R}^n$. Da eksisterer der en vektor $\vec{0} \neq \vec{d} \in \mathbb{R}^n$, som opfylder at $\vec{a}_i \cdot \vec{d} = 0$ for $i \in I$.
Betragt da linjen bestående af alle punkter på formen $\vec{y} = \vec{x} + \lambda\vec{d}$, hvor $\lambda \in \mathbb{R}$. Tydeligvis opfylder alle punkter på denne linje, at $\vec{a}_i \cdot \vec{y} = b_i$ for $i \in I$. De begrænsninger der derfor er aktive i $\vec{x}$, vil da også være aktive i $\vec{y}$. Da P ikke indeholder en linje, vil vi på et tidspunkt bryde en begrænsning ved et tilpas stort λ, se figur 4.19. Da eksisterer der en reel skalar $\hat{\lambda}$, hvor en ny begrænsning bliver aktiv: $\vec{a}_j \cdot (\vec{x} + \hat{\lambda}\vec{d}) = b_j$ for et $j \notin I$. Da gælder at $\vec{a}_j \cdot \vec{x} \neq b_j$, da $j \notin I$. Derfor må $\vec{a}_j \cdot \vec{d} \neq 0$, hvilket betyder at $\vec{a}_j$ ikke kan være en linearkombination af $\vec{a}_i$, hvor $i \in I$.

Vi har nu beskrevet en procedure hvorved antallet af lineært uafhængige aktive begrænsninger stiger med mindst en, ved at vi bevæger os fra $\vec{x}$ til $\vec{x} + \hat{\lambda}\vec{d}$. Fortsættelse af proceduren vil resultere i, at mindst n lineært uafhængige begrænsninger bliver aktive i samme punkt. Dette punkt vil per definition 4.16 være en basal løsning og en basal mulig løsning, altså et kritisk punkt, da vi kun har bevæget os inden for P.

Dernæst vil vi bevise, at (b) $\Rightarrow$ (c). Antag at P har et kritisk punkt $\vec{x}$. Grundet sætning 4.17 er $\vec{x}$ også en basal mulig løsning. Per definition eksisterer der da n lineært uafhængige begrænsninger, som er aktive i $\vec{x}$.

For at bevise at (c) $\Rightarrow$ (a) antager vi, at n af vektorerne $\vec{a}_i$ er lineært uafhængige. Vi kan da bytte om på rækkerne, sådan at $\vec{a}_1, \ldots, \vec{a}_n$ er linært uafhængige. Antag nu, at P indeholder en linje på formen $\vec{x} + \lambda\vec{d}$, hvor $\vec{d} \neq \vec{0}$ og $\lambda \in \mathbb{R}$. Da skal følgende gælde: $\forall\lambda : \vec{a}_i \cdot (\vec{x} + \lambda\vec{d}) \geq b_i$, for alle $i = 1, \ldots, m$. Hvis $\vec{a}_i \cdot \vec{d} < 0$, kan vi vælge et tilpas stort λ, hvor $\vec{a}_i \cdot (\vec{x} + \lambda\vec{d}) < b_i$ for $i = 1, \ldots, m$. Tilsvarende gælder det samme, hvis $\vec{a}_i \cdot \vec{d} > 0$. Derfor må $\vec{d} = \vec{0}$, hvilket er en modstrid, som betyder, at P ikke indeholder en linje [2, s. 63-64]. ■

Vi kan betragte illustrationen i figur 4.18, som mulighedsmængden tilhørende et lineært programmeringsproblem afgrænset udelukkende af et enkelt halvrum. Vi ser i dette tilfælde, at antallet af begrænsninger garanterer eksistensen af en linje, og derfor vil et sådan problem ikke indeholde nogle kritiske punkter. På standardform er et sådan tilfælde selvfølgelig ikke muligt, da enhver variabel vil have en tilhørende ikke-negativ begrænsning. Lineære programmeringsproblemer på standardform vil derfor mindst have ét kritisk punkt.

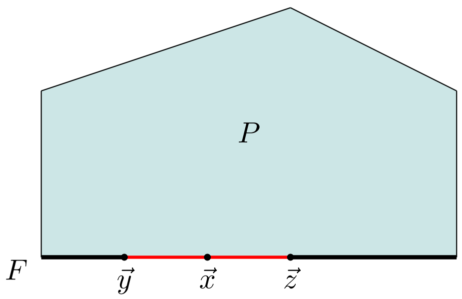
4.6 Optimalitet af kritiske punkter

Tidligere i dette kapitel antydede vi, at den optimale løsning til lineære programmeringsproblemer vil findes i et kritisk punkt, når den optimale værdi af objektfunktionen er endelig. Vi vil derfor indlede dette afsnit med at bevise, at hvis den optimale værdi er endelig, da kan vi finde en optimal løsning i mindst ét af polyederets kritiske punkter. Vi vil derefter afslutte kapitlet med et bevis for vores tidligere udsagn; at mulige lineære programmeringsproblemer enten har en uendelig optimal værdi eller en optimal løsning i ét kritisk punkt.

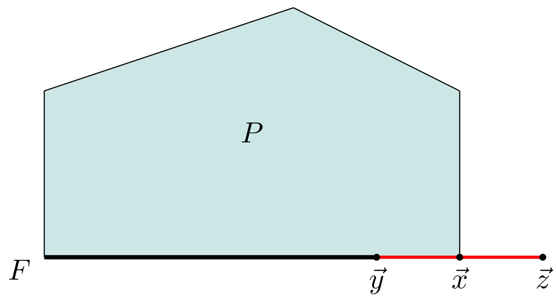
Sætning 4.26

Betragt et lineært programmeringsproblem, hvor objektfunktionen $\vec{c} \cdot \vec{x}$ ønskes minimeret, for et polyeder P. Antag at P har mindst ét kritisk punkt, og at der findes en optimal løsning. Så eksisterer der en optimal løsning, som er et kritisk punkt i P [2, s. 65].

Bevis:
Lad F være en ikke-tom mængde af alle optimale løsninger.
Lad P være et polyeder givet ved $P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} \geq \vec{b}\}$ og lad v være den optimale værdi af objektfunktionen $\vec{c} \cdot \vec{x}$. Så er $F = \{\vec{x} \in \mathbb{R}^n | A\vec{x} \geq \vec{b}, \vec{c} \cdot \vec{x} = v\}$ ligeledes et polyeder.
Idet P har et kritisk punkt, indeholder den ikke en linje, ifølge sætning 4.25. Da F er en delmængde af P, så indeholder F heller ikke en linje. Dermed må F have et kritisk punkt.



Figur 4.20: Vektor $\vec{x}$ er ikke et kritisk punkt i F.



Figur 4.21: Vektor $\vec{x}$ er et kritisk punkt i F.

Lad så $\vec{x}$ være et kritisk punkt i F. Vi ønsker da at vise, at $\vec{x}$ også er et kritisk punkt i P. For at gøre dette, viser vi det kontrapositive udsagn. Antag da, at $\vec{x}$ *ikke* er et kritisk punkt i P. Dette betyder ifølge definition 4.13, at der eksisterer to vektorer, som opfylder at $\vec{y}, \vec{z} \neq \vec{x}$ og $\vec{y}, \vec{z} \in P$. Derudover findes der en reel skalar $\lambda \in [0, 1]$, sådan at vi kan udtrykke $\vec{x}$ som en konveks kombination af $\vec{y}$ og $\vec{z}$:

$$\vec{x} = \lambda\vec{y} + (1-\lambda)\vec{z}.$$

Dette er illustreret i figur 4.20. Dette medfører, at

$$v = \vec{c} \cdot \vec{x} = \lambda\vec{c} \cdot \vec{y} + (1-\lambda)\vec{c} \cdot \vec{z}.$$

Idet v er den optimale værdi af objektfunktionen, vil det betyde, at hverken $\vec{c} \cdot \vec{y}$ eller $\vec{c} \cdot \vec{z}$ kan være strengt mindre end $\vec{c} \cdot \vec{x}$. Deraf får vi, at

$$\vec{c} \cdot \vec{y} \geq \vec{c} \cdot \vec{x} = v \qquad \text{og} \qquad \vec{c} \cdot \vec{z} \geq \vec{c} \cdot \vec{x} = v.$$

Idet vi ved, at $\vec{x}$ kan udtrykkes som en konveks kombination af $\vec{y}$ og $\vec{z}$ og ligger i F, må det betyde, at

$$v = \vec{c} \cdot \vec{x} = \vec{c} \cdot \vec{y} = \vec{c} \cdot \vec{z}.$$

Vi ser, at $\vec{y}$ og $\vec{z}$ begge er lig den optimale værdi v, når de prikkes med $\vec{c}$, hvilket betyder at $\vec{y}, \vec{z} \in F$. Da dette er tilfældet, betyder det, at $\vec{x}$ heller ikke kan være et kritisk punkt i F.

Det kontrapositive af dette udsagn vil derfor medføre, at hvis $\vec{x}$ er et kritisk punkt i F, så er $\vec{x}$ et kritisk punkt i P - se figur 4.21. Og idet $\vec{x}$ tilhører F, betyder det også, at $\vec{x}$ er en optimal løsning [2, s. 65-66]. ■

Vi vil nu bevise vores antagelse om hjørnepunkters optimalitet:

Sætning 4.27

Betragt et lineært programmeringsproblem, hvor objektfunktionen $\vec{c} \cdot \vec{x}$ ønskes minimeret, for et polyeder P. Antag at P har mindst ét kritisk punkt. Da er den optimale værdi af objektfunktionen enten lig med $-\infty$, eller der eksisterer et kritisk punkt, som er optimalt [2, s. 66].

Bevis:
I dette bevis definerer vi rank($\vec{x}$), som antallet af lineært uafhængige aktive begrænsninger i en vektor $\vec{x}$ i P.

Vi antager, at den optimale værdi af objektfunktionen er endelig, og at P er et polyeder på formen $P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} \geq \vec{b}\}$. Vi betragter nogle $\vec{x} \in \mathbb{R}^n$ med rank$(\vec{x}) = k < n$. Vi vil vise, at der eksisterer nogle $\vec{y} \in P$, som har en større rang end $\vec{x}$ og opfylder at $\vec{c} \cdot \vec{y} \leq \vec{c} \cdot \vec{x}$. Lad $I = \{i \mid \vec{a_i} \cdot \vec{x} = b_i\}$. Da $k < n$ ligger vektorene $\vec{a}_i, i \in I$ i et ikke-trivielt underrum af $\mathbb{R}^n$. Derfor kan vi vælge en vektor $\vec{d} \neq \vec{0} \in \mathbb{R}^n$, som opfylder at $\vec{a}_i \cdot \vec{d} = 0$, for alle $i \in I$.

Antag at $\vec{c} \cdot \vec{d} < 0$ og betragt *halvlinjen* bestående af vektorerne $\vec{y} = \vec{x} + \lambda\vec{d}$, for alle $\lambda > 0$. Som det blev bevist i sætning 4.25, gælder det, at $\vec{a}_i \cdot \vec{y} = b_i$ for $i \in I$. Hvis hele halvlinjen er indeholdt i P, vil den optimale værdi af objektfunktionen være lig med $-\infty$. Dette er i modstrid med antagelsen om, at objektfunktionen har en endelig værdi. Derfor må halvlinjen gå ud af P. Når dette sker, har vi nogle $\hat{\lambda} > 0$ og et nyt indeks $j \notin I$, sådan at $\vec{a_j} \cdot (\vec{x} + \hat{\lambda}\vec{d}) = b_j$. Vi lader $\vec{y} = \vec{x} + \hat{\lambda}\vec{d}$ og vi bemærker, at $\vec{c} \cdot \vec{y} < \vec{c} \cdot \vec{x}$. Som vist i sætning 4.25, er $\vec{a_j}$ lineært uafhængig af $\vec{a_i}, i \in I$, og $\vec{y}$ har rang mindst $k + 1$.

Antag nu, at $\vec{c} \cdot \vec{d} = 0$ og betragt linjen $\vec{y} = \vec{x} + \lambda\vec{d}$, hvor λ er en vilkårlig skalar. Da P ikke indeholder en linje, må linjen derfor gå ud af P. Igen når dette sker, får vi at rank($\vec{y}$)>rank($\vec{x}$). Deraf har vi, at $\vec{c} \cdot \vec{y} = \vec{c} \cdot \vec{x}$.

I begge tilfælde er der fundet en ny vektor $\vec{y}$, som opfylder at $\vec{c} \cdot \vec{y} \leq \vec{c} \cdot \vec{x}$, og som har større rang end $\vec{x}$. Denne procedure gentages så mange gange, som det er nødvendigt, og vi vil til sidst få en vektor $\vec{w}$, som har rang n. Dette vil gøre den til en basal mulig løsning og betyde, at $\vec{c} \cdot \vec{w} \leq \vec{c} \cdot \vec{x}$.

Lad $\vec{w}_1, \ldots, \vec{w}_r$ være de basale mulige løsninger i P, og lad $\vec{w}$ være en basal mulig løsning, sådan at $\vec{c} \cdot \vec{w} \leq \vec{c} \cdot \vec{w}_i$ for alle $i = 1, \ldots, r$. Vi har vist, at der for enhver $\vec{x}$ eksisterer nogle i, sådan at $\vec{c} \cdot \vec{w}_i \leq \vec{c} \cdot \vec{x}$. Dette medfører at $\vec{c} \cdot \vec{w} \leq \vec{c} \cdot \vec{x}$ for alle $\vec{x} \in P$ og $\vec{w}$ er derfor den optimale løsning [2, s. 66-67]. ■

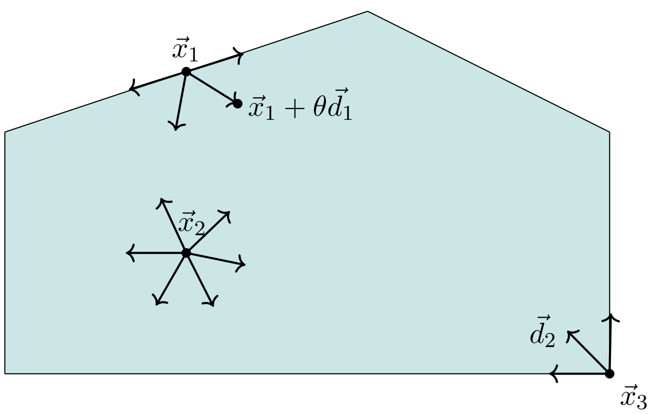
En strategi til at finde den optimale løsning i et lineært programmeringsproblem, vil da være at undersøge alle de kritiske punkter i det tilhørende polyeder. Vi kan dog forestille os at finde og sammenligne alle basale mulige løsninger i et stort lineært programmeringsproblem kan være besværligt rent beregningsmæssigt, f.eks i polyederet set i figur 4.1. Vi vil derfor udlede Simplex-metoden i det næste kapitel, da den giver en systematisk måde til at gennemgå de basale mulige løsninger.

5 | Simplex-metoden

I dette kapitel vil vi nu udlede en metode til at bevæge os imellem hjørnepunkter i et polyeder og sammenligne deres værdi i objektfunktionen, i håb om at finde den optimale løsning. Denne metode kaldes Simplex-metoden, og giver os muligheden for at løse lineære programmeringsproblemer algoritmisk. Derudover vil vi betragte forskellige implementationer af Simplex-metoden, samt forklare hvordan vi undgår problemet med degenererede løsninger. Vi vil fortsætte med at betragte lineære minimeringsproblemer på standardform givet ved polyederet $P = \{\vec{x} \in \mathbb{R}^n \mid A\vec{x} = \vec{b}, \vec{x} \geq \vec{0}\}$, hvor A er en $m \times n$-matrix med lineært uafhængige rækker.

5.1 Konstruktion af Simplex-metoden

En måde, hvorpå vi i et optimeringsproblem kan lede efter en optimal løsning, er ved at betragte en allerede kendt mulig løsning til problemet og lede efter en bedre løsning i nærheden af denne. Hvis en bedre løsning ikke eksisterer, vil den kendte da være et lokalt ekstrema. I lineære minimeringsproblemer ønsker vi at minimere en konveks funktion defineret på en konveks mængde. Vi husker, at vi tidligere i sætning 4.11 har vist, at et lokalt minimum i dette tilfælde også vil være et globalt minimum. Vi kan altså starte i en kendt basal mulig løsning og sammenligne den med de nærliggende basale mulige løsninger, for at finde den optimale løsning. Vi vil i dette afsnit derfor udlede en metode til at bevæge os systematisk imellem problemets basale mulige løsninger og samtidigt betragte ændringen i funktionsværdien.



Figur 5.1: Vektorerne $\vec{x}_1, \vec{x}_2$ og $\vec{x}_3$ har alle forskellige mulige retninger afhængigt af deres placering i polyederet.

Antag at vi står i en basal mulig løsning $\vec{x}$ i et polyeder P på standard form og ønsker at finde en nærliggende basal mulig løsning. For at bevæge os væk fra vores nuværende løsning, skal vi vælge en retningsvektor $\vec{d}$, som *ikke* bringer os uden for polyederet, som i figur 5.1. En sådan vektor definerer vi herunder:

Definition 5.1 Mulige retninger

Lad $\vec{x}$ være en vektor i P. En vektor $\vec{d} \in \mathbb{R}^n$ kaldes en *mulig retning* fra $\vec{x}$, hvis der eksisterer en positiv skalar θ, sådan at $\vec{x} + \theta\vec{d} \in P$ [2, s. 83].

Som direkte konsekvens af et polyeders konveksitet har vi, at hvis $\vec{x} \in P$ og $\vec{d}$ er en mulig retning, da vil $\vec{x} + \theta^{'}\vec{d} \in P$ for alle $\theta^{'} \in]0, \theta]$.

Lad os nu finde en mulig retning $\vec{d}$, som bringer os fra den basale mulige løsning $\vec{x}$ til en ny løsning $\vec{x} + \theta\vec{d}$. Vi kan vælge at bevæge os i en retning, der øger værdien af én ikke-basal variabel x_j, imens vi fastholder værdien af alle de andre ikke-basale variable. Vektoren $\vec{d}$ vil derfor have komponenten $d_j = 1$ tilhørende den valgte ikke-basale variabel og $d_i = 0$ for de resterende indeks tilhørende ikke-basale variable. For at finde værdien af de resterende komponenter i retningsvektoren, konstruerer vi vektoren $\vec{d}_B = (d_{B(1)}, \ldots, d_{B(m)})^T$, som består af komponenterne tilhørende de basale indeks [2, s. 83].

For at garantere, at $\vec{x}+\theta\vec{d}$ er en mulig løsning på standardform, kræves det, at $A(\vec{x}+\theta\vec{d}) = \vec{b}$. Da $\vec{x}$ er en basal mulig løsning, ved vi, at $A\vec{x} = \vec{b}$. Vi får da, at

$$\begin{aligned} A(\vec{x} + \theta\vec{d}) = A\vec{x} + \theta A\vec{d} = \vec{b} \Leftrightarrow \\ \vec{b} + \theta A\vec{d} = \vec{b} \Leftrightarrow \\ \theta A\vec{d} = \vec{0}. \end{aligned}$$

Da $\theta > 0$, skal den mulige retning $\vec{d}$ ligge i nulrummet tilhørende matricen A. Da $d_j = 1$ og $d_i = 0$ for alle ikke-basale variable med indeks $i \neq j$, kan vi betragte $A\vec{d} = \vec{0}$ på følgende måde:

$$\begin{aligned} \vec{0} = A\vec{d} &= \sum_{i=1}^{n} \vec{A}_i d_i \\ &= \vec{A}_j + \sum_{i=1}^{m} \vec{A}_{B(i)} d_{B(i)} \\ &= \vec{A}_j + B\vec{d}_B. \end{aligned}$$

Vi husker, at basismatricen, tilhørende den basale mulige løsning, er en invertibel matrix B. Vektoren $\vec{d}_B$ kan da isoleres, ved at gange B^{-1} på fra venstre:

$$\begin{aligned} \vec{A}_j + B\vec{d}_B = \vec{0} \Leftrightarrow \\ B\vec{d}_B = -\vec{A}_j \Leftrightarrow \\ \vec{d}_B = -B^{-1}\vec{A}_j \quad [2, s. 83]. \end{aligned}$$

Bemærk at valget af et andet ikke-basalt indeks j ville have udledt en anden retningsvektor $\vec{d}$. Vi vælger derfor at kalde retningen, tilhørende et valgt indeks j, for den *j'te basale retning*:

Definition 5.2 Basale retninger

Lad $\vec{x}$ være en basal mulig løsning til et lineært programmeringsproblem på standardform. Den *j'te basale retning* $\vec{d}$ fra $\vec{x}$ har komponenterne:

$$\begin{aligned} d_j &= 1 \\ d_i &= 0 \ \text{ for } \ i \neq B(1), \ldots, B(m) \text{ og } i \neq j \\ \vec{d}_B &= -B^{-1}\vec{A}_j \end{aligned}$$

for et ikke-basalt indeks j [2, s. 83].

Vi husker, at vi ønsker at finde en mulig retning fra vores kendte basale mulige løsning. Vi vil nu vise at den j'te basale retning er en mulig retning, hvis vores basale mulige løsning er ikke-degenereret:

Sætning 5.3

Lad $\vec{x}$ være en ikke-degenereret basal mulig løsning til et lineært programmeringsproblem på standardform. Da er den j'te basale retning fra $\vec{x}$ en mulig retning i det tilsvarende polyeder [2, s. 84].

Bevis:
Vi ved at vektoren $\vec{d}$ ligger i nulrummet tilhørende A. Det betyder, at $A(\vec{x}+\theta\vec{d}) = \vec{b}$. Deraf ses det, at alle ligheder er opfyldt. Vi mangler nu blot at vise, at de ikke-negative begrænsninger er overholdt. Værdien af den ikke-basale variabel x_j øges strengt, da $d_j = 1$ og $\theta > 0$. Alle andre ikke-basale variable beholder værdien 0. Vi skal derfor vise, at komponenterne i $\vec{x}_B + \theta\vec{d}_B$ overholder den ikke-negative begrænsning.
Da $\vec{x}$ er en ikke-degenereret basal mulig løsning, ved vi, at $\vec{x}_B > \vec{0}$. Når θ derfor er lille nok, vil $\vec{x}_B + \theta\vec{d}_B \geq \vec{0}$, hvilket betyder, at ethvert komponent i $\vec{x}_B + \theta\vec{d}_B$ overholder den tilhørende ikke-negative begrænsning [2, s. 84]. ■

Vi kan, som konsekvens heraf, altid bruge en basal retning fra en basal mulig løsning som en mulig retning, hvis den basale mulige løsning er ikke-degenereret.

Vi støder her på det første problem med degenererede løsninger. Betragt en degenereret basal mulig løsning $\vec{x}$ og en tilhørende basal retning $\vec{d}$. Da $\vec{x}$ er degenereret, vil det gælde at $x_{B(i)} = 0$ for mindst et $B(i)$ tilhørende løsningens base. I tilfældet hvor den tilsvarende komponent i $\vec{d}$ er negativ, vil den tilsvarende ikke-negative begrænsning brydes af $\vec{x} + \theta\vec{d}$ for alle $\theta > 0$. Af den grund skal vi være opmærksomme på degenererede basale mulige løsninger.

Målet ved at bevæge os væk fra vores nuværende basale mulige løsning, er at finde en ny løsning med en mindre værdi af objektfunktionen. Lad os derfor undersøge ændringen i omkostningen, ved at gå i en basal retning $\vec{d}$:

$$\begin{aligned}\vec{c}\cdot\vec{d} &= \sum_{i=1}^{n} c_i d_i \\ &= \vec{c}_B\cdot\vec{d}_B + c_j \\ &= \vec{c}_B\cdot(-B^{-1}\vec{A}_j) + c_j \\ &= c_j - \vec{c}_B\cdot(B^{-1}\vec{A}_j).\end{aligned}$$

Vi kan nu vurdere, hvorvidt bevægelsen i den j'te basale retning reducerer omkostningen. Dette er selvfølgelig tilfældet, hvis $c_j - \vec{c}_B\cdot(B^{-1}\vec{A}_j) < 0$, da vi for alle positive θ da har

$$\vec{c}\cdot(\vec{x}+\theta\vec{d}) = \vec{c}\cdot\vec{x} + \theta\vec{c}\cdot\vec{d} = \vec{c}\cdot\vec{x} + \theta(c_j - \vec{c}_B\cdot(B^{-1}\vec{A}_j)) < \vec{c}\cdot\vec{x}.$$

Vi vælger derfor at definere størrelsen på reduktionen for hvert θ, som den *reducerede omkostning*:

Definition 5.4 Reducerede omkostninger

Lad $\vec{x}$ være en basal løsning med basismatrix B og dertilhørende omkostningsvektor $\vec{c}_B = (c_{B(1)}, c_{B(2)}, \ldots, c_{B(m)})^T$. Værdien

$$\tilde{c}_j = c_j - \vec{c_B}\cdot(B^{-1}\vec{A}_j)$$

kaldes den *reducerede omkostning* for den j'te basale retning [2, s. 84].

For at sammenligne alle de reducerede omkostninger på én gang, kan vi konstruere en *reduceret omkostningsvektor* - hvis k'te komponent betegner den reducerede omkostning - ved at vælge den k'te basale retning fra en basal mulig løsning:

$$\tilde{\vec{c}} = (\vec{c}^T - \vec{c}_B^T B^{-1}A)^T = \begin{bmatrix} c_1 - \vec{c}_B\cdot(B^{-1}\vec{A}_1) \\ \vdots \\ c_k - \vec{c}_B\cdot(B^{-1}\vec{A}_k) \\ \vdots \\ c_n - \vec{c}_B\cdot(B^{-1}\vec{A}_n)\end{bmatrix}.$$

Bemærk at den overstående vektor har en indgang tilhørende alle problemets variable. Vektoren vil da tydeligvis have nul-indgange i de indgange tilhørende en basal variabel, da deres basale retninger er nulvektoren.

Betragt konstruktionen af en basal retning ved et valg af et basalt indeks $B(j)$. Vi skal som før sørge for, at retningen ikke bryder vores hovedbegrænsninger, altså at $A(\vec{x}+\theta\vec{d}) = \vec{b}$. Vi får da, at retningsvektoren nødvendigvis må tilhøre nulrummet til A, så $A\vec{d} = \vec{0}$. Da vi har valgt, at $\vec{d}_{B(j)} = 1$, må alle komponenter i retningsvektoren med ikke-basale indeks være lig nul. Vi får da, at den basale retning skal opfylde, at $B\vec{d}_B = \vec{0}$. Da B er invertibel, ses det, at den eneste mulige retning er nul-vektoren. En basal retning tilhørende en basal variabel vil derfor altid være nul-vektoren.

Dette afspejler sig i de basale indgange i den reducerede omkostningsvektor:

$$\begin{aligned}\tilde{c}_{B(j)} &= c_{B(j)} - \vec{c}_B \cdot (B^{-1}\vec{A}_{B(j)}) \\ &= c_{B(j)} - \vec{c}_B \cdot \vec{e}_j \\ &= c_{B(j)} - c_{B(j)} = 0 \text{ [2, s. 84].}\end{aligned}$$

Det vil derfor aldrig betale sig at forsøge at øge værdien af en basal variabel.

Hvis den reducerede omkostningsvektor $\tilde{\vec{c}} \geq \vec{0}$, kan det ikke betale sig at bevæge os væk fra den nuværende basale mulige løsning mod en nærliggende løsning. Vores nuværende basale mulige løsning vil derfor være et lokalt minimum af objektfunktionen, hvilket medfører, at det også er et globalt minimum og derfor optimal. Dette beviser vi er tilfældet herunder:

Sætning 5.5

Betragt en basal mulig løsning $\vec{x}$ associeret med en basismatrix B og lad $\tilde{\vec{c}}$ være den tilhørende reducerede omkostningsvektor.

(a) Hvis $\tilde{\vec{c}} \geq \vec{0}$, så er $\vec{x}$ optimal.

(b) Hvis $\vec{x}$ er optimal og ikke-degenereret, så er $\tilde{\vec{c}} \geq \vec{0}$ [2, s. 86].

Bevis:

(a) Vi vil først vise, at hvis $\tilde{\vec{c}} \geq \vec{0}$, så er $\vec{x}$ en optimal løsning.
Lad $\tilde{\vec{c}} \geq \vec{0}$ og lad $\vec{y}$ være en vilkårlig mulig løsning. Vi definerer da, at $\vec{d} = \vec{y} - \vec{x}$. Da $\vec{x}$ og $\vec{y}$ er mulige løsninger, ved vi, at $A\vec{x} = A\vec{y} = \vec{b} \Rightarrow A\vec{d} = 0$. Lad N være indeksmængden for de ikke-basale variable og B være basismatricen. Da kan vi omskrive ligeheden $A\vec{d} = \vec{0}$ til

$$B\vec{d}_B + \sum_{i \in N} \vec{A}_i d_i = \vec{0}.$$

Da B er en invertibel matrix, kan vi isolere $\vec{d}_B$ ved at gange B^{-1} på fra venstre på begge sider:

$$\begin{aligned}B\vec{d}_B &= -\sum_{i \in N} \vec{A}_i d_i \Leftrightarrow \\ B^{-1}B\vec{d}_B &= -B^{-1}\sum_{i \in N} \vec{A}_i d_i \Leftrightarrow \\ I\vec{d}_B &= -\sum_{i \in N} B^{-1}\vec{A}_i d_i.\end{aligned}$$

Vi ser da, at

$$\vec{d}_B = -\sum_{i \in N} B^{-1}\vec{A}_i d_i.$$

Vi kan nu betragte forskellen i objektfunktionen $\vec{c} \cdot \vec{d}$ på følgende måde:

$$\begin{aligned}
\vec{c} \cdot \vec{d} &= \vec{c}_B \cdot \vec{d}_B + \sum_{i \in N} c_i d_i = \vec{c}_B \cdot \left(-\sum_{i \in N} (B^{-1} \vec{A}_i) d_i \right) + \sum_{i \in N} c_i d_i \\
&= -\vec{c}_B \cdot \left(\sum_{i \in N} (B^{-1} \vec{A}_i) d_i \right) + \sum_{i \in N} c_i d_i \\
&= \left(\sum_{i \in N} -\vec{c}_B \cdot (B^{-1} \vec{A}_i) d_i \right) + \sum_{i \in N} c_i d_i \\
&= \sum_{i \in N} \left(-\vec{c}_B \cdot (B^{-1} \vec{A}_i) \right) d_i + c_i d_i \\
&= \sum_{i \in N} \left(c_i - \vec{c}_B \cdot (B^{-1} \vec{A}_i) \right) d_i \Leftrightarrow \\
\vec{c} \cdot \vec{d} &= \sum_{i \in N} \tilde{c}_i d_i.
\end{aligned}$$

Vi husker, at $x_i = 0, i \in N$. Idet $\vec{y}$ er en mulig løsning har vi, at $y_i \geq 0, i \in N$. Da ved vi, at $d_i \geq 0, i \in N$. Heraf: $\vec{c} \cdot \vec{d} = \vec{c} \cdot \vec{y} - \vec{c} \cdot \vec{x} \geq 0$. Dette medfører, at $\vec{c} \cdot \vec{y} \geq \vec{c} \cdot \vec{x}$. Da $\vec{y}$ er en vilkårlig mulig løsning, er $\vec{x}$ en optimal løsning.

(b) Vi vil nu vise (b) ved kontraposition. Lad $\vec{x}$ være en ikke-degenereret basal mulig løsning og $\tilde{c}_j < 0$. Vi ved, at den reducerede omkostning $\tilde{c}_i = 0$ for alle basale variable x_i, for $i = B(1), \ldots, B(m)$. Komponenten x_j må derfor være en ikke-basal variabel. Da $\vec{x}$ er en ikke-degenereret basal mulig løsning, ved vi, at den j'te basale retning $\vec{d}$, er en mulig retning. Vi kan derfor finde en vektor $\vec{y} = \vec{x} + \theta \vec{d}$, så $\vec{c} \cdot \vec{y} < \vec{c} \cdot \vec{x}$, og $\vec{x}$ er derfor *ikke* en optimal løsning.

Det kontrapositive udsagn af dette er dermed at, når $\vec{x}$ er en ikke-degenereret optimal løsning, så er $\tilde{\vec{c}} \geq \vec{0}$ [2, s. 86].

■

Vi kan nu med overstående resultat være sikre på, at vi har en optimal basal mulig løsning, hvis dens tilhørende reducerede omkostningsvektor er ikke-negativ, og hvis den opfylder den ikke-negative begrænsning. Bemærk dog at vi, som konsekvens af resultatet, ikke kan tjekke optimaliteten af degenererede basale mulige løsninger med denne metode. For at kunne anvende overstående resultat til at bestemme en optimal løsning til et problem, vælger vi at definere kravene for en *optimal basismatrix* herunder:

Definition 5.6 Optimale basismatricer

En basismatrix B siges at være *optimal*, hvis:

(a) $B^{-1}\vec{b} \geq \vec{0}$, og

(b) $\tilde{\vec{c}} \geq \vec{0}$ [2, s. 87].

Vi bemærker, at det første krav i overstående definition er til for at opfylde den ikke-negative begrænsning, da $B^{-1}\vec{b} = \vec{x}_B$.

Vi har nu udledt en metode til at finde basale retninger, der kan reducere omkostningen i problemets objektfunktion. Lad os nu undersøge, hvorvidt vi skal bevæge os i en valgt basal retning. Vi starter med at antage, at vi betragter et problem uden degenererede løsninger, hvilket sikrer, at alle basale retninger er mulige retninger, som bevist i sætning 5.3.

Antag nu at vi starter i en basal mulig løsning $\vec{x}$. Vi starter med at beregne dens tilhørende reducerede omkostningsvektor $\tilde{\vec{c}}$. Hvis vi her ser, at ingen basale retninger vil reducere omkostningen, ved vi da, at vores løsning er optimal, og vi er færdige. Vi kan forestille os, at dette sjældent er tilfældet. Dette betyder, at der findes en basal retning $\vec{d}$, hvis tilhørende reducerede omkostning $\tilde{c}_j$ er negativ. Ved at bevæge os langs denne retning vil værdien af den tilhørende ikke-basale variabel x_j stige, imens de resterende ikke-basale variable forbliver nul. Vi vil fremover her sige, at variablen x_j og dens tilhørende ikke-basale søjle *træder* ind i basen.

Da omkostningsværdien vil falde i den j'te basale retning $\vec{d}$, ønsker vi at bevæge os så langt væk som muligt fra løsningen $\vec{x}$. Vi ønsker altså at finde den største skalar θ^*, som opfylder, at den nye vektor er en mulig løsning:

$$\theta^* = \max\{\theta \geq 0 \mid \vec{x} + \theta\vec{d} \in P\}. \tag{5.7}$$

Vi vil nu udlede en formel for θ^*. Givet $A\vec{d} = \vec{0}$, har vi, at $A(\vec{x} + \theta\vec{d}) = A\vec{x} = \vec{b}$ for alle $\theta \geq 0$. Problemets hovedbegrænsninger $A\vec{x} = \vec{b}$ vil da aldrig blive brudt. Vi skal altså kun bekymre os om problemets ikke-negative begrænsning, når vi vil finde θ^*. Der vil her opstå to tilfælde, alt efter hvordan koefficienterne i den basale retning fremtræder:

1. Hvis $\vec{d} \geq \vec{0}$, så er $\vec{x} + \theta\vec{d} \geq \vec{0}$ for alle $\theta \geq 0$. Enhver ny løsning langs den basale retning vil da være mulig, og vi lader $\theta^* = \infty$.

2. Hvis $d_i < 0$ for mindst ét indeks i tilhørende et af problemets variable, skal vi tage højde for den tilhørende ikke-negativ begrænsning $x_i + \theta d_i \geq 0$. Vi kan omformulere dette til en begrænsning af θ ved at omskrive til $\theta \leq -x_i/d_i$. Denne begrænsning skal være opfyldt for ethvert indeks i, hvorom det gælder, at $d_i < 0$. Bemærk at $\theta = -x_i/d_i$ også vil opfylde begrænsningen for θ. Den største mulige skalar θ^*, som ikke bringer os uden for polyederet, er da:

$$\theta^* = \min_{\{i \mid d_i < 0\}} \left(-\frac{x_i}{d_i} \right). \tag{5.8}$$

   Vi husker, at de ikke-basale indgange i den basale retning er ikke-negative. Vi behøver derfor kun at tjekke de basale indeks:

$$\theta^* = \min_{\{i=1,\dots,m \mid d_{B(i)} < 0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right) \text{ [2, s. 88-89]}. \tag{5.9}$$

Antag at vi i et problem finder en basal retning, der både opfylder at dens indgange er ikke-negative, og at den dertilhørende reducerede omkostning er negativ. I dette tilfælde vil $\theta^* = \infty$. Vi vil her være færdige, da værdien af objektfunktionen kan blive vilkårlig lille. Problemet er altså ubegrænset nedadtil og den optimale værdi vil være $-\infty$ - som vi så i afsnit 3.2. Dette vil ikke være tilfældet, hvis θ^* er begrænset.

Vi antager derfor i stedet, at θ^* er endelig. Efter bestemmelse af θ^* bevæger vi os til den nye mulig løsning $\vec{y} = \vec{x} + \theta^* \vec{d}$. Den valgte j'te basale retning har en dertilhørende ikke-basal variabel og indgang, som før bevægelsen opfyldte; $x_j = 0$ og $d_j = 1$. Værdien af den j'te variabel i $\vec{y}$ er da $y_j = \theta^* > 0$. Den ikke-basale variabel x_j, som trådte ind i basen, bliver derfor positiv og en ny basal-variabel. Dette betyder dog, at vi skal skifte den ud med en eksisterende basal variabel. For at en basal variabel i løsningen $\vec{x}$ kan skiftes ud, skal den være en ikke-basal variabel i løsningen $\vec{y}$, hvilket betyder, at den nødvendigvis skal være lig nul [2, s. 89]. Vi vil nu vise, at dette også er tilfældet for mindst én basal variabel tilhørende den basale mulige løsning $\vec{x}$.

Lad indekset l være et indeks, som opfylder, at

$$\theta^* = -\frac{x_{B(l)}}{d_{B(l)}} = \min_{\{i=1,\dots,m \mid d_{B(i)}<0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}} \right),$$

hvor $d_{B(l)} < 0$. Betragt nu den $B(l)$'te indgang i den nye løsning $\vec{y}$:

$$y_{B(l)} = x_{B(l)} + \theta^* d_{B(l)} = x_{B(l)} - \frac{x_{B(l)}}{d_{B(l)}} d_{B(l)} = 0.$$

Den basale variabel $x_{B(l)}$ er blevet nul, hvorimod den før valgte ikke-basale variabel x_j er blevet positiv. Vi kan nu opstille en ny base, hvor vi udskifter $B(l)$ med j. Dette betyder selvfølgelig også, at vi skal opstille en ny basismatrix $\tilde{B}$, hvor vi erstatter søjlen $\vec{A}_{B(l)}$ med $\vec{A}_j$ i den gamle basismatrix:

$$\tilde{B} = \begin{bmatrix} \vec{A}_{B(1)} & \cdots & \vec{A}_{B(l-1)} & \vec{A}_j & \vec{A}_{B(l+1)} & \cdots & \vec{A}_{B(m)} \end{bmatrix}.$$

Den gamle base $\{B(1), \dots, B(m)\}$ udskiftes altså med en ny indeksmængde $\{\tilde{B}(1), \dots, \tilde{B}(m)\}$, givet ved:

1. $\tilde{B}(i) = B(i)$, hvis $i \neq l$
2. $\tilde{B}(i) = j$, hvis $i = l$ [2, s. 89].

Det, at vi skifter ud i basen, kaldes også en *pivot-operation*.

Vi vil nu vise, at den nye løsning $\vec{y} = \vec{x} + \theta^* \vec{d}$ er en basal mulig løsning, der er tilknyttet den nye base, og at basens tilhørende matrix er en basismatrix:

Sætning 5.10

(a) Søjlerne $\vec{A}_i$, for alle $i \in \{\tilde{B}(1), \dots, \tilde{B}(m)\}$ er lineært uafhængige.

(b) Vektoren $\vec{y} = \vec{x} + \theta^* \vec{d}$ er en basal mulig løsning tilknyttet basismatricen $\tilde{B}$ [2, s. 89].

Bevis:

(a) Antag at søjlerne $\vec{A}_{\tilde{B}(i)}$, $i = 1, \ldots, m$, er lineært afhængige. Da findes der tilhørende skalarer $\lambda_1, \ldots, \lambda_m$, hvor mindst én er forskellig fra 0, sådan at

$$\sum_{i=1}^{m} \lambda_i \vec{A}_{\tilde{B}(i)} = \vec{0}. \tag{5.11}$$

Da basismatricen B er invertibel, kan vi omskrive udtrykket til

$$\sum_{i=1}^{m} \lambda_i B^{-1} \vec{A}_{\tilde{B}(i)} = \vec{0},$$

hvilket viser, at vektorerne $B^{-1}\vec{A}_{\tilde{B}(i)}$ også er lineært afhængige, idet $\vec{0}$ kan udtrykkes som en linearkombination af søjlerne, hvor mindst én koefficient er forskellig fra 0.

Vi vil nu vise, at dette ikke kan lade sig gøre, ved at bevise at vektorerne $B^{-1}\vec{A}_{\tilde{B}(i)}$ er lineært uafhængige. Vi ved, at $B^{-1}B = I_m$. Da $\vec{A}_{\tilde{B}(i)}, i \neq l$, er søjler i B, vil vektorene $B^{-1}\vec{A}_{\tilde{B}(i)} = \vec{e}_i$, for $i \neq l$. Derimod vil vektoren $B^{-1}\vec{A}_{\tilde{B}(l)} = -\vec{d}_B$ - jævnfør definition 5.2.
Komponenten $-d_{B(l)}$ er forskellig fra 0, givet valget af l, som opfylder ligning 5.9. Vektoren $B^{-1}\vec{A}_{\tilde{B}(l)}$ kan derfor ikke udtrykkes ved standardvektorerne $\vec{e}_i$, hvor $i \neq l$. Søjlerne $B^{-1}\vec{A}_{\tilde{B}(i)}$ for $i = 1, \ldots, m$ er da lineært uafhængige.
Vi har nu, at ligning 5.11 kun er opfyldt, hvis $\lambda_i = 0$, for $i = 1, \ldots, m$. Matricen $\tilde{B}$ er derfor en basismatrix.

(b) Valget af retningsvektoren $\vec{d}$ og skalaren θ^* medfører, at vektoren $\vec{y} \geq \vec{0}$, $A\vec{y} = \vec{b}$ og at $y_i = 0$, for $i \neq \tilde{B}(1), \ldots, \tilde{B}(m)$. Da søjlerne i basismatricen $\tilde{B}$ er uafhængige, må $\vec{y}$ være en basal mulig løsning tilknyttet basismatricen [2, s. 89-90].

■

Vi kan nu udlede en metode, hvorpå vi kan bevæge os fra en basal mulig løsning, til en ny basal mulig løsning med lavere omkostning, ved at følge en basal retning i polyederet. Dette er, hvad vi kalder Simplex-metoden.

5.2 Den naive Simplex-metode

Da hver iteration i Simplex-metoden slutter i en basal mulig løsning, kan vi nu starte en ny iteration med denne løsning i jagten på en bedre basal mulig løsning. Dette kan gentages indtil en optimal løsning er fundet.

Vi vil nu, ud fra ovenstående viden, opskrive en iteration af Simplex-metoden. Det viser sig at være smart at definere vektoren $\vec{u}$ ved

$$\vec{u} = -\vec{d}_B = B^{-1}\vec{A}_j,$$

for at simplificere beregningerne.

Vi starter med at udlede en naiv implementation af Simplex-metoden.

En iteration i den naive Simplex-metode

1. Start med en basismatrix B, med søjlerne $\vec{A}_{B(1)}, \dots, \vec{A}_{B(m)}$ og en dertilhørende basal mulig løsning $\vec{x}$.
2. Udregn den reducerede omkostning $\tilde{c}_j = c_j - \vec{c}_B \cdot (B^{-1}\vec{A}_j)$ for alle ikke-basale variable j. Hvis alle de reducerede omkostninger er ikke-negative, så er $\vec{x}$ den optimale løsning og metoden stopper. Vælg ellers ét j, hvor $\tilde{c}_j < 0$.
3. Udregn vektoren $\vec{u} = B^{-1}\vec{A}_j$. Hvis der ikke er en positiv komponent i $\vec{u}$, så er $\theta^* = \infty$ og den optimale omkostning bliver derfor $-\infty$, hvorefter metoden stopper.
4. Hvis der derimod er en positiv komponent i $\vec{u}$, udregnes θ^* ud fra følgende:
$$\theta^* = \min_{\{i=1,\dots,m \mid u_i>0\}} \left(\frac{x_{B(i)}}{u_i} \right)$$
5. Lad l opfylde, at $\theta^* = \frac{x_{B(l)}}{u_l}$. Konstruer en ny basismatrix $\tilde{B}$ ved at udskifte $\vec{A}_{B(l)}$ med $\vec{A}_j$. Den nye basismatrix har en tilhørende basal mulig løsning $\vec{y}$, hvorom det gælder, at $y_j = \theta^*$ og $y_{B(i)} = x_{B(i)} - \theta^* u_i$ for $i \neq l$.

En følge af disse iterationer kalder vi den naive Simplex-metode. Når vi bestemmer θ^*, antager vi, at enhver basal mulig løsning i problemet er ikke-degenereret. Vi vil nu vise, at hvis dette er tilfældet, så vil Simplex-metoden stoppe efter et endeligt antal iterationer:

Sætning 5.12

Antag at mulighedsmængden er ikke-tom og enhver basal mulig løsning er ikke-degenereret. Da stopper Simplex-metoden efter et endeligt antal iterationer [2, s. 91].

Bevis:
Der er to mulige grunde til at Simplex-metoden stopper:

(a) Hvis metoden stopper i 2. trin, skyldes det, at vi har fundet en basismatrix B, som er optimal. Da er $\vec{\tilde{c}} \geq \vec{0}$. Dette medfører, at den nuværende basale mulige løsning $\vec{x}$ er optimal jævnfør sætning 5.5.

(b) Hvis metoden stopper i 3. trin, har vi en basal mulig løsning $\vec{x}$, hvis tilhørende reducerede omkostningsvektor har en negativ indgang, altså $\tilde{c}_j < 0$. Den j'te basale retning $\vec{d}$ opfylder $A\vec{d} = \vec{0}$, og at $\vec{d} \geq \vec{0}$. Vi har da, at $\vec{x} + \theta\vec{d} \in P$ for alle $\theta > 0$. Da $\vec{c} \cdot \vec{d} = \tilde{c}_j < 0$, kan vi ved at vælge et vilkårligt stort θ, lave omkostningen vilkårligt mere negativ. Den optimale løsning er derfor $-\infty$.

Ved enhver iteration flytter vi løsningen med en positiv værdi af θ^* langs retningen $\vec{d}$, som opfylder $\vec{c} \cdot \vec{d} < 0$. Derfor bliver omkostningen skarpt mindre, for hver gang metoden undersøger et punkt. Ingen af de basale mulige løsninger undersøges derfor mere end én gang. Vi har tidligere vist, at der i problemer opstillet med et endeligt antal af begrænsninger kun findes et endeligt antal af basale mulige løsninger, se korollar 4.18. Antallet af metodens iterationer er derfor også endeligt, og vi vil nødvendigvis finde en basal mulig løsning, som opfylder (a) eller en basal retning, der opfylder (b) [2, s. 91]. ■

Bemærk at det overstående resultat svarer til det geometriske resultat i sætning 4.27, som vi beviste i kapitel 4.

5.3 Pivot-regler og degenererede problemer

Vi har indtil nu antaget, at vi anvender Simplex-metoden på problemer, hvis basale mulige løsninger er ikke-degenererede. Lad os nu undersøge, hvad der sker, hvis Simplex-metoden støder på en degenereret basal mulig løsning. Antag derfor at vi udfører metoden på et problem med degenererede basale mulige løsninger. Algoritmen kan da støde på følgende:

1. Hvis vi starter en iteration med en degenereret basal mulig løsning $\vec{x}$, kan der opstå en situation, hvor $\theta^* = 0$. Dette er tilfældet, hvis en basal variabel $x_{B(l)} = 0$ og den tilhørende indgang i den j'te basale retning $d_{B(l)} < 0$. En pivot-operation kan stadig gennemføres, men den nye basale mulige løsning vil være identisk med den gamle. Omkostningen vil derfor ikke ændre sig. Den nye basis $\tilde{B}$ konstrueres stadig ved at bytte søjlen $\vec{A}_{B(l)}$ med $\vec{A}_j$.
2. Hvis vi under en iteration støder på en degenereret basal mulig løsning, vil det betyde at flere af de basale variable er blevet lig nul i den nye løsning. Dette er tilfældet, hvis der er flere basale indeks $B(l)$, som opfylder
$$-\frac{x_{B(l)}}{d_{B(l)}} = \min_{\{i=1,\ldots,m \mid d_{B(i)}<0\}} \left(-\frac{x_{B(i)}}{d_{B(i)}}\right).$$
Den nye basale mulige løsning vil da være degenereret, da kun én af disse basale variable træder ud af basen.

Simplex-metoden, som beskrevet indtil videre, giver en vis grad af frihed i valget af, hvilken ikke-basal variabel x_j, vi vil lade træde ind i den nye base. Vi kræver blot, at den tilhørende indgang i den reducerede omkostningsvektor er negativ. På samme måde kan der være flere indeks l, som opfylder beregningen af θ^*, hvor vi kun skal vælge ét til at træde ud af basen. Et frit valg af de indgående indeks i pivot-operationer i problemer med degenererede basale mulige løsninger kan lede til problemer.

Vi kan forestille os situationer, hvor et frit valg af basisskift - hvor ét indeks træder ind, og et andet træder ud af basen - kan lede os tilbage til den base, vi startede med. Hvis vi fortsætter med at vælge de samme indeks, vil vi blive ved med at starte og slutte i den samme basale mulige løsning, og vi vil gå i ring. Et sådan eksempel kan ses i afsnit 6.4. Dette er selvfølgelig problematisk, da det vil betyde, at Simplex-metoden aldrig vil finde en optimal løsning og derfor ikke stopper. Af den grund er det smart at pålægge retningslinjer for valg af de tiltrædende og fratrædende indeks. En sådan retningslinje kaldes en *pivot-regel*.

For valget af hvilket indeks, der træder ind i basen, findes der to klassiske pivot-regler:

1. **Mest negative reducerede omkostning**: Her vælges indekset j, hvorom det gælder, at den tilhørende reducerede omkostning $\tilde{c}_j$ er den mindste. Vi vælger altså den mulige retning, hvor værdien af objektfunktionen aftager hurtigst.

Den samlede ændring af objektfunktionens værdi afhænger dog både af $\tilde{c}_j$ og θ^*. Dette leder frem til den anden pivot-regel.

2. **Største reduktion**: Her vælges det indeks j, hvor vi kan reducere objektfunktionens værdi mest. Dette vil selvfølgelig være ved det valg af indeks j, hvor værdien $\theta^* \tilde{c}_j$ er mest negativ [2, s. 93].

Begge pivot-regler har fordele og ulemper, dog kan ingen af dem garantere, at vi ikke går i ring. Der er derfor en tredje pivot-regel, *Blands regel*, der viser sig, at kunne løse nogle af de problemer, som de to klassiske pivot-regler ikke kan løse.

Blands regel

Blands regel kendes også som *mindste-indeks-reglen*, da vi ved valg af tiltrædende indeks vælger den første mulige retning, vi finder, hvor den reducerede omkostning er negativ. Dette vil selvfølgelig være ved det mindste indeks j, hvor $\tilde{c}_j < 0$ [2, s. 94, 111]. Ligeledes skal vi under Blands regel vælge det indeks l med mindste værdi til at tiltræde basen. Man kan bevise at, hvis man følger Blands regel, vil Simplex-metoden undgå at gå i ring. Dette bevises dog ikke i indeværende projekt, men der henvises derimod til beviset i henvisninger i [2, s. 137].

5.4 Revideret Simplex-metode

Ved en naiv implementation af Simplex-metoden findes den inverse til basismatricen B på traditionel vis i starten af hver iteration. I en alternativ implementation af Simplex-metoden, kendt som den *Reviderede* Simplex-metode, udregnes den inverse til den nye basismatrix $\tilde{B}$ i slutningen af hver iteration ved hjælp af elementære rækkeoperationer på den forrige basismatrix B. I det følgende afsnit vil vi udlede en metode til at gøre dette.

Lad matricen $B = [\vec{A}_{B(1)} \cdots \vec{A}_{B(m)}]$ være basismatricen ved begyndelsen af en iteration, og lad matricen $\tilde{B} = [\vec{A}_{B(1)} \cdots \vec{A}_{B(l-1)} \;\; \vec{A}_j \;\; \vec{A}_{B(l+1)} \cdots \vec{A}_{B(m)}]$ være den nye basismatrix fundet til sidst i iterationen. B og $\tilde{B}$ deler de samme søjler undtagen den $B(l)$'te søjle, som er blevet skiftet ud i den nye basismatrix. Vi kan derfor forestille os, at B^{-1} kan bruges til at udregne $\tilde{B}^{-1}$.

Vi ved, at $B^{-1}B = I_m$. Dette betyder, at $B^{-1}\vec{A}_{B(i)} = \vec{e}_i$, hvor $\vec{e}_i$ er den i'te standardvektor. Ud fra dette ser vi, at matrix-produktet $B^{-1}\tilde{B}$ er givet ved

$$B^{-1}\tilde{B} = [\vec{e}_1 \cdots \vec{e}_{l-1} \;\; \vec{u} \;\; \vec{e}_{l+1} \cdots \vec{e}_m] \quad [2, \text{s. } 95].$$

Bemærk at $\vec{u} = B^{-1}\vec{A}_j$. Vi husker nu, at en følge af elementære rækkeoperationer på en matrix B, er det samme som at udregne matrix-produktet QB, hvor matricen Q er et produkt af elementær-matricer, som vi har set i afsnit 2.5.2. Vi kan derfor finde en matrix Q, sådan at

$QB^{-1}\tilde{B} = I_m$. Det at gange Q på $B^{-1}\tilde{B}$ fra venstre svarer altså til den følge af elementære rækkeoperationer, som omskriver $B^{-1}\tilde{B}$ til identitetsmatricen I_m.
Følgende følge af rækkeoperationer gør dette:

1. Lav den l'te søjle om til $u_l\vec{e}_l$ ved rækkeoperationen $\frac{-u_i}{u_l}\vec{r}_l + \vec{r}_i \to \vec{r}_i$, for alle $i \neq l$.
2. Skalér den l'te række med rækkeoperationen $\frac{1}{u_l}\vec{r}_l \to \vec{r}_l$ [2, s. 97].

Betragt igen ligningen $QB^{-1}\tilde{B} = I_m$. Ved at gange matricen $\tilde{B}^{-1}$ på fra højre på begge sider af lighedstegnet får vi, at $QB^{-1} = \tilde{B}^{-1}$.

Vi gennemfører nu et eksempel på konstruktionen af $\tilde{B}^{-1}$ ved brug af den ovenstående følge af rækkeoperationer.

Eksempel 5.13.
Lad matricen $[B^{-1} \mid \vec{u}\,]$ være givet ved

$$[B^{-1} \mid \vec{u}\,] = \left[\begin{array}{ccc|c} 1 & 2 & 4 & 3 \\ 0 & 5 & 1 & 6 \\ 3 & 0 & 1 & 12 \end{array}\right].$$

Vi vil da, ved hjælp af ovenstående procedure, komme frem til matricen $[\tilde{B}^{-1} \mid \vec{e}_2]$. Bemærk at $l = 2$.

$$[B^{-1} \mid \vec{u}\,] = \left[\begin{array}{ccc|c} 1 & 2 & 4 & 3 \\ 0 & 5 & 1 & 6 \\ 3 & 0 & 1 & 12 \end{array}\right] \sim \begin{matrix} \vec{r}_1 - (3/6)\vec{r}_2 \to \vec{r}_1 \\ \vec{r}_3 - (12/6)\vec{r}_2 \to \vec{r}_3 \end{matrix} \sim \left[\begin{array}{ccc|c} 1 & -1/2 & 7/2 & 0 \\ 0 & 5 & 1 & 6 \\ 3 & -10 & -1 & 0 \end{array}\right]$$

$$\sim (1/6)\vec{r}_2 \to \vec{r}_2 \sim \left[\begin{array}{ccc|c} 1 & -1/2 & 7/2 & 0 \\ 0 & 5/6 & 1/6 & 1 \\ 3 & -10 & -1 & 0 \end{array}\right] = [\tilde{B}^{-1} \mid \vec{e}_2].$$

Matricen $\tilde{B}^{-1}$ er altså givet ved

$$\tilde{B}^{-1} = \begin{bmatrix} 1 & -1/2 & 7/2 \\ 0 & 5/6 & 1/6 \\ 3 & -10 & -1 \end{bmatrix}.$$

Udfører vi samme følge af rækkeoperationer på identitetsmatricen I_3, får vi matricen

$$Q = \begin{bmatrix} 1 & -1/2 & 0 \\ 0 & 1/6 & 0 \\ 0 & -2 & 1 \end{bmatrix}.$$

◀

Da vi kender følgen af elementære rækkeoperationer, kan vi finde et mere generelt udtryk for Q. Vi husker, at Q kan betragtes som summen $I_m + D$, og at indgangene i D fortæller noget om, hvilke rækkeoperationer vi skal lave, som set i afsnit 2.5.2. Da I_m er kendt, betragter vi derfor kun følgen af rækkeoperationer, vi skal lave for at bestemme D. Vi skal udføre rækkeoperationen $\frac{-u_i}{u_l}\vec{r}_l + \vec{r}_i \to \vec{r}_i$ for alle $i \neq l$ og $\frac{1}{u_l}\vec{r}_l \to \vec{r}_l$. Den l'te søjle i D må da være

$$\vec{D}_l^T = \left(\frac{-u_1}{u_l}, \cdots, \frac{-u_{l-1}}{u_l}, \frac{1}{u_l} - 1, \frac{-u_{l+1}}{u_l}, \cdots, \frac{-u_m}{u_l} \right).$$

Resten af indgangene i D er, som beskrevet, lig 0, og vi har nu en generel metode til at udtrykke Q. Bemærk at den l'te indgang i den l'te søjle i D består af udtrykket $\frac{1}{u_l} - 1$. Den l'te indgang i den l'te søjle i Q er da givet

$$Q_{ll} = I_{ll} + D_{ll} = 1 + \frac{1}{u_l} - 1 = \frac{1}{u_l}.$$

Dette sikrer 2. trin i følgen af elementære rækkeoperationer; skaleringen af den l'te række.

Ved at udregne den inverse til basismatricen $\tilde{B}$, som beskrevet ovenfor, kan vi udlede den Reviderede Simplex-metode [2, s. 95]. Bemærk at vi introducerer $1 \times m$-matricen K, når vi skal udregne den reducerede omkostning for alle ikke-basale variable. På den måde undgår vi at udregne $\vec{c}_B^T B^{-1}$, hver gang vi udregner $\tilde{c}_j$ for et ikke-basalt indeks j.

En iteration i den Reviderede Simplex-metode

1. Start med en basismatrix B, med søjlerne $\vec{A}_{B(1)}, \ldots, \vec{A}_{B(m)}$, dens inverse B^{-1} og den tilhørende basale mulige løsning $\vec{x}$.

2. Udregn $1 \times m$-matricen $K = \vec{c}_B^T B^{-1}$. Udregn herefter den reducerede omkostning $\tilde{c}_j = c_j - K\vec{A}_j$ for alle ikke-basale indeks j. Hvis alle $\tilde{c}_j$ er ikke-negative, så er $\vec{x}$ den optimale løsning og metoden stopper, ellers vælg et j hvor $\tilde{c}_j < 0$.

3. Udregn vektoren $\vec{u} = B^{-1}\vec{A}_j$. Hvis der ikke er en positiv komponent i $\vec{u}$, så er $\theta^* = \infty$ og den optimale omkostning bliver derfor $-\infty$, hvorefter metoden stopper.

4. Hvis der derimod er en positiv komponent i $\vec{u}$, udregn da θ^* ud fra følgende:

$$\theta^* = \min_{\{i=1,\ldots,m \mid u_i > 0\}} \left(\frac{x_{B(i)}}{u_i} \right).$$

5. Lad l opfylde, at $\theta^* = \frac{x_{B(l)}}{u_l}$. Lav en ny basismatrix ved at udskifte $\vec{A}_{B(l)}$ med $\vec{A}_j$. Denne nye basismatrix har en tilhørende basal mulig løsning $\vec{y}$, hvorom det gælder, at $y_j = \theta^*$ og $y_{B(i)} = x_{B(i)} - \theta^* u_i$ for $i \neq l$.

6. Opstil matricen $[B^{-1} \mid \vec{u}\,]$. Omskriv den sidste søjle $\vec{u}$ til standardvektoren $\vec{e}_l$ ved brug af Q. De første m søjler i denne nye matrix er den inverse til den nye basismatrix $\tilde{B}$ - altså $\tilde{B}^{-1}$ [2, s. 97-98].

5.5 Simplex på Tabelform

En anden klassisk implementation af Simplex-metoden er *Simplex på Tabelform*. Hvor den Reviderede Simplex-metode skiller sig ud ved effektivt at opdatere B^{-1}, opdaterer Simplex på Tabelform en $m \times (n+1)$-matrix $[B^{-1}A \mid B^{-1}\vec{b}\,]$, som kaldes *Simplex-tabellen*, ved hver iteration. Den sidste søjle $B^{-1}\vec{b}$ i Simplex-tabellen kaldes *den* 0*'te søjle* og indeholder værdien af de basale variable. Med andre ord er den 0'te søjle den vektor, som vi tidligere har kaldt $\vec{x}_B$. Søjlen $B^{-1}\vec{A}_i$ kaldes den i'te søjle. Søjlen $\vec{u} = B^{-1}\vec{A}_j$ for det j, hvor den tilhørende ikke-basale variabel indtræder i basen, kaldes *pivot-søjlen*. For det valgte indeks l, der opfylder ligning 5.9, kalder vi den l'te række for *pivot-rækken*. Den skalar u_l i $\vec{u}$, som ligger i pivot-rækken kaldes *pivot-elementet*. Dette anskueliggøres i nedenstående matrix:

$$\begin{array}{cc} & \vec{u} \\ \vec{r}_l & \begin{bmatrix} & | & \\ — & u_l & — \\ & | & \end{bmatrix} \end{array}$$

. Bemærk at pivot-elementet u_l altid vil være positivt, idet metoden istedet ville have fundet en mulig retning, hvor θ^* er ubegrænset, hvis $\vec{u} \leq \vec{0}$, og metoden da ville stoppe [2, s. 98]. Vi kan betragte Simplex-tabellen som en totalmatrix, der repræsenterer ligningsystemet

$$B^{-1}A\vec{x} = B^{-1}\vec{b} \Leftrightarrow A\vec{x} = \vec{b}.$$

Her anvender Simplex på Tabelform den samme følge af rækkeoperationer som den Reviderede Simplex-metode til at opdatere B^{-1} sådan, at skalarerne i pivot-søjlen ændres til 0 pånær pivot-elementet, som får værdien 1. Ofte vælger vi at udvide Simplex-tabellen ved at tilføje *den* 0*'te række* til tabellen så den står på formen

$$S = \begin{array}{|c|c|} \hline \vec{c}^T - \vec{c}_B^T B^{-1} A & -\vec{c}_B^T B^{-1}\vec{b} \\ \hline B^{-1}A & B^{-1}\vec{b} \\ \hline \end{array}$$

hvilket også kan skrives

$$S = \begin{array}{|ccc|c|} \hline \tilde{c}_1 & \dots & \tilde{c}_n & -\vec{c}_B^T B^{-1}\vec{b} \\ \hline | & & | & \vec{x}_{B(1)} \\ B^{-1}\vec{A}_1 & \dots & B^{-1}\vec{A}_n & \vdots \\ | & & | & \vec{x}_{B(m)} \\ \hline \end{array} \quad [2, s. 99].$$

Bemærk at den sidste indgang i den 0'te række angiver den negative værdi af objektfunktionen $-\vec{c}_B \cdot \vec{x}_B$. Det vil vise sig, at det er smart at angive den negative værdi, når vi skal opdatere Simplex-tabellen, da det giver mulighed for at omskrive den 0'te række på en særlig form. De resterende skalarer i den 0'te række svarer til den reducerede omkostningsvektor $\tilde{\vec{c}}^T$. Den 0'te række opdateres på samme måde som de andre rækker ved at lave de elementære rækkeoperationer, der omdanner $\vec{u}$ til den l'te standardvektor $\vec{e}_l$. Det vil nu blive vist, at når vi opdaterer den 0'te række med denne metode, vil vi få de rigtige værdier. Vi opdaterer altså objektfunktionens værdi og den reducerede omkostningsvektor.

Betragter vi igen følgen af elementære rækkeoperationer på Simplex-tabellen som produktet QS, hvor $Q = I + D$, bemærker vi, at ved tilføjelsen af den 0'te række til Simplex-tabellen, må Q være en $(m+1) \times (m+1)$-matrix. Den første indgang i den l'te søjle i D må have værdien $\frac{-\tilde{c}_j}{u_l}$. De resterende indgange i den l'te søjle vil være som tidligere beskrevet:

$$\vec{D}_l{}^T = \left(\frac{-\tilde{c}_j}{u_l}, \frac{-u_1}{u_l}, \ldots, \frac{-u_{l-1}}{u_l}, \frac{1}{u_l} - 1, \frac{-u_{l+1}}{u_l}, \ldots, \frac{-u_m}{u_l} \right).$$

De resterende indgange i D er, ligesom før, lig 0. Vi kan nu bruge Q til at opdatere den udvidede Simplex-tabel S.

Da vi vælger at lade variablen x_j træde ind i basen, skal dens reducerede omkostning blive lig 0, altså $\tilde{c}_j = \tilde{c}_{B(l)} = 0$. Vi vil nu vise, at dette er tilfældet ved brug af Q. Betragt indgangen hørende til $\tilde{c}_j$ i den opdaterede udvidede Simplex-tabel $(QS)_{1j}$:

$$(QS)_{1j} = \vec{q}_1 \cdot \vec{S}_j = \vec{q}_1 \cdot \begin{bmatrix} \tilde{c}_j \\ \vec{u} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \frac{-\tilde{c}_j}{u_l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \begin{bmatrix} \tilde{c}_j \\ u_1 \\ \vdots \\ u_{l-1} \\ u_l \\ u_{l+1} \\ \vdots \\ u_m \end{bmatrix} = \tilde{c}_j - \frac{\tilde{c}_j}{u_l} u_l = 0.$$

Ligeledes skal det gælde, at de reducerede omkostninger for de resterende basale variable skal forblive nul. Vi vil altså vise, at $\tilde{c}_{B(i)} = 0$, for $i \neq l$ i den opdaterede udvidede Simplex-tabel. Betragt en indgang i den 0'te række tilhørende en basal variabel $B(i)$, for $i \neq l$ i den opdaterede udvidede Simplex-tabel $(QS)_{1B(i)}$:

$$(QS)_{1B(i)} = \vec{q}_1 \cdot \vec{S}_{B(i)} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ \frac{-\tilde{c}_j}{u_l} \\ 0 \\ \vdots \\ 0 \end{bmatrix} \cdot \vec{e}_{B(i)} = 0.$$

Bemærk at da $i \neq l$, vil det l'te led i skalarproduktet være $\frac{-\tilde{c}_j}{u_l} \cdot 0 = 0$. På samme måde vil alle andre led i linearkombinationen opfylde nulreglen.

Vi har nu vist, at de reducerede omkostninger tilhørende de basale variable opdateres korrekt. Vi vil nu vise, at dette er tilfældet for alle indgange i den 0'te række.

I starten af en iteration vil den 0'te række stå på formen

$$\begin{aligned}[\vec{c}^{\,T} - \vec{c}_B^{\,T} B^{-1} A \mid \, -\vec{c}_B^{\,T} B^{-1} \vec{b}\,] &= [\vec{c}^{\,T} \mid 0\,] - [\,\vec{c}_B^{\,T} B^{-1} A \mid \vec{c}_B^{\,T} B^{-1} \vec{b}\,] \\ &= [\vec{c}^{\,T} \mid 0\,] - \vec{c}_B^{\,T} B^{-1} [A \mid \vec{b}\,].\end{aligned}$$

Vi husker, at $\vec{c}_B^{\,T} B^{-1} = K$, se afsnit 5.4. Da får vi, at

$$[\vec{c}^{\,T} \mid 0\,] - \vec{c}_B^{\,T} B^{-1} [A \mid \vec{b}] = [\vec{c}^{\,T} \mid 0\,] - K[A \mid \vec{b}\,].$$

Bemærk da at den 0'te række består af en linearkombination af rækkerne i $[A \mid \vec{b}\,]$, trukket fra $[\vec{c}^{\,T} \mid 0\,]$. Her indeholder indgangene i $1 \times m$-matricen K linearkombinationens koefficienter. Vi har vist, at de basale indgange i den opdaterede Simplex-tabels 0'te række er blevet lig nul ved brug af Q. I dette tilfælde er en reduceret omkostning tilhørende en basal variabel opdateret ved at tilføje en linearkombination af indgangene i den tilhørende basale søjle. Vi har altså vist, at det for det nye indeks i basen gælder, at

$$c_j - G\vec{A}_j = c_{\tilde{B}(l)} - G\vec{A}_{\tilde{B}(l)} = 0,$$

og, at det for de gamle indeks gælder, at

$$c_{B(i)} - G\vec{A}_{B(i)} = c_{\tilde{B}(i)} - G\vec{A}_{\tilde{B}(i)} = 0, \text{for } i \neq l,$$

hvor $1 \times m$-matricen G indeholder linearkombinationens koefficienter. Vi ser, at dette medfører, at de basale indgange i den reducerede omkostningsvektor består delvist af en linearkombination af rækkerne i den opdaterede basismatrix $\tilde{B}$:

$$\vec{c}_{\tilde{B}}^{\,T} - G\tilde{B} = \vec{0}^{\,T}.$$

Vi kan nu udlede G:

$$\begin{aligned}\vec{c}_{\tilde{B}}^{\,T} - G\tilde{B} &= \vec{0}^{\,T} \Leftrightarrow \\ -G\tilde{B} &= -\vec{c}_{\tilde{B}}^{\,T} \Leftrightarrow \\ G &= \vec{c}_{\tilde{B}}^{\,T} \tilde{B}^{-1}.\end{aligned}$$

Vi ser da, at den 0'te række i den opdaterede Simplex-tabel er givet ved

$$[\,\vec{c}^{\,T} \mid 0\,] - \vec{c}_{\tilde{B}}^{\,T} \tilde{B}^{-1} [\,A \mid \vec{b}\,].$$

De reducerede omkostninger, tilhørende de ikke-basale variable, opdateres da også korrekt. Ligeledes gælder det for værdien af objektfunktionen [2, s. 100].

Af ovenstående kan vi hermed udlede en iteration i Simplex på Tabelform.

En iteration i Simplex på Tabelform

1. Start med en Simplex-tabel, en basismatrix B, den 0'te række og en basal mulig løsning $\vec{x}$.
2. Betragt de reducerede omkostninger i den 0'te række. Hvis $\tilde{\vec{c}} \geq \vec{0}$ er $\vec{x}$ en optimal løsning og metoden stopper. Vælg ellers et j, hvor $\tilde{c}_j < 0$.

3. Betragt pivot-søjlen $\vec{u} = B^{-1}\vec{A}_j$. Hvis $\vec{u} \geq \vec{0}$ er den optimale værdi $-\infty$ og metoden stopper.
4. Udregn $\frac{x_{B(i)}}{u_i}$ for alle i, hvor $u_i > 0$. Lad indekset l opfylde, at $\frac{x_{B(l)}}{u_l} \leq \frac{x_{B(i)}}{u_i}$ for alle i, hvor $u_i > 0$. Søjlen $\vec{A}_{B(l)}$ træder ud af basen og søjlen $\vec{A}_j$ træder ind i basen.
5. Rekonstruér søjlen $\vec{u}$ til den l'te standardvektor $\vec{e}_l$ ved brug af Q. [2, s. 100].

5.6 To-Fase Simplex-metoden

I de tre ovenstående implementationer af Simplex-metoden begynder algoritmerne med en basal mulig løsning. Det er derfor essentielt at have en metode til at finde en sådan.

Bemærk, at hvis vi omskriver et lineært programmeringsproblem underlagt de lineært uafhængige begrænsninger $A\vec{x} \leq \vec{b}$ og den ikke-negative begrænsning $\vec{x} \geq \vec{0}$, vil vi allerede have en kendt basal mulig løsning, hvis $\vec{b} \geq \vec{0}$. Dette er tilfældet, da vi har indført underskudsvariablene s_i, for $i = 1, \ldots, m$, så hovedbegrænsningerne er på formen $A\vec{x}+\vec{s} = \vec{b}$. Løsningen $\vec{s} = \vec{b}$ og $\vec{x} = \vec{0}$ er da en basal mulig løsning, hvor den tilhørende basismatrix er identitetsmatricen.

Da vi i dette projekt arbejder med vilkårlige lineære programmeringsproblemer omskrevet til standardform, er dette *ikke* altid tilfældet. Vi kan derfor bruge To-Fase Simplex-metoden til at finde en basal mulig løsning, vi kan begynde med.
Vi betragter først det lineære programmeringsproblem på standardform:

$$\begin{array}{ll} \text{Minimér} & Z = \vec{c} \cdot \vec{x} \\ \text{Underlagt} & A\vec{x} = \vec{b} \\ & \vec{x} \geq \vec{0} \end{array}$$

Vi antager, at $\vec{b} \geq \vec{0}$, da vi ved at gange skalaren -1 på den tilsvarende begrænsning kan skifte fortegn, som vi vil. I første fase betragter vi vektoren $\vec{y} \in \mathbb{R}^m$, som repræsenterer nogle *støtte-variable*. Vi kan da, ved hjælp af Simplex-metoden, løse følgende problem, kaldet *støtte-problemet*.

$$\begin{array}{ll} \text{Minimér} & Z_y = y_1 + \cdots + y_m \\ \text{Underlagt} & A\vec{x} + \vec{y} = \vec{b} \\ & \vec{x} \geq \vec{0} \\ & \vec{y} \geq \vec{0} \end{array}$$

Her kan vi starte med den basale mulige løsning $\vec{x} = \vec{0}$ og $\vec{y} = \vec{b}$, hvor den tilhørende basismatrix er identitetsmatricen.
Hvis vi da finder en optimal løsning til støtte-problemet $(\vec{x}, \vec{y})$, hvor $\vec{x}$ er en mulig løsning til det oprindelige problem og $\vec{y} = \vec{0}$, har vi fundet en mulig løsning til det oprindelige problem.

Hvis vi derimod finder en optimal løsning $(\vec{x}, \vec{y})$, hvor $\vec{y}$ indeholder nogle indgange med positive værdier, vil den optimale værdi af objektfunktionen $Z_y = y_1 + \cdots + y_m$ være positiv. Det vil sige, at vi ikke kan finde en vektor $\vec{x}$, som opfylder, at $A\vec{x} = \vec{b}$, og det oprindelige problem har derfor ikke en løsning [2, s. 112]. Hvis den tilhørende basismatrix B til den mulige løsning $\vec{x}$, består af søjler fra A, er $\vec{x}$ en basal mulig løsning, og vi kan fortsætte til anden fase, som finder den optimale løsning til det oprindelige problem ved at fjerne de kunstige variable og de tilhørende søjler.

Hvis én eller flere af de basale variable er kunstige variable, skal vi først have skiftet dem ud. Lad $c < m$ være antallet af søjler i A, som ligger i basismatricen B, og lad indeksene $B(i)$, for $i = 1, \ldots, c$ betegne disse. Vi ved, at $\vec{y} = \vec{0}$ og at alle ikke-basale variable må have værdien 0. Variablene $x_{B(1)}, \ldots, x_{B(c)}$ må da være de eneste variable forskellige fra 0. Da vi antager, at A har lineært uafhængige rækker, kan vi finde $m - c$ søjler $\vec{A}_{B(c+1)}, \ldots, \vec{A}_{B(m)}$, sådan at søjlerne $\vec{A}_{B(1)}, \ldots, \vec{A}_{B(m)}$ er lineært uafhængige og danner en basismatrix udelukkende bestående af søjler fra A [2, s. 113]. Vi kan da fjerne de kunstige variable og deres tilhørende søjler. Denne operation udføres på samme måde, som når vi i Simplex på Tabelform bytter underskudsvariable ud af basismatricen.

En iteration i To-Fase Simplex-metoden

Første fase:

1. Omskriv begrænsningerne så $\vec{b} \geq \vec{0}$ ved at skalere de rækker, hvor $\vec{b}$'s indgange er negative, med -1.
2. Tilføj de kunstige variable $y_1, \ldots, y_m$ hvis nødvendigt, og minimér objektfunktionen $Z_y = y_1 + \cdots + y_m$ underlagt de nye begrænsninger.
3. Hvis den optimale værdi af Z_y er positiv, har det oprindelige problem ingen løsning og algoritmen stopper.
4. Hvis den optimale værdi af Z_y er 0 er $\vec{x}$ en mulig løsning til det oprindelige problem. Hvis basismatricen B ikke indeholder søjler, hvis tilsvarende variable er kunstige, fjernes de kunstige variable og deres søjler, og B er en basismatrix til det oprindelige problem.
5. Hvis den l'te basale variabel er kunstig, undersøg da den l'te række i $B^{-1}A$. Hvis den j'te indgang i rækken er forskellig fra 0, lad da denne indgang være pivot-element, og udskift basis ved at skifte den l'te basale variabel ud med x_j. Gentag denne procedure indtil ingen basale variable er kunstige variable.

Anden fase:

1. Løs det oprindelige problem med udgangspunkt i den basale mulige løsning og basismatrix fundet i første fase ved hjælp af Simplex-metoden [2, s. 116].

Denne implementation af Simplex-metoden har, i modsætning til de implementationer tidligere beskrevet, den fordel, at den undersøger, om problemet er løseligt. Bemærk dog, at To-Fase Simplex-metoden ikke er en selvstændig metode, der finder en optimal løsning til et lineært programmeringsproblem, men derimod er den en tilføjelse til en af de tre overstående Simplex-metoder.

6 Eksempler på løsning af lineære programmeringsproblemer

I dette kapitel gennemføres fire eksempler: ét eksempel, hvor Simplex på Tabelform anvendes, ét eksempel på anvendelsen af To-Fase Simplex-metoden, ét eksempel på degenererede løsninger til lineære programmeringsproblemer og slutteligt ét eksempel på Simplex der vil gå i ring.

6.1 Eksempel på Simplex på Tabelform

Nedenfor har vi et lineært programmeringsproblem, som vi ønsker at minimere.

$$\begin{aligned} \text{Minimér} \quad Z = & -16x_1 + 2x_2 + x_3 - 2x_4 - 3x_5 \\ \text{Underlagt} \quad & 3x_1 + x_2 + 3x_3 - 3x_4 + 9x_5 \leq 12 \\ & 2x_1 + 8x_2 + 4x_3 + 2x_4 - 4x_5 \leq 10 \\ & x_1, \ldots, x_5 \geq 0 \end{aligned}$$

Vi ønsker at minimere objektfunktionen ved hjælp af Simplex på Tabelform med pivot-reglen *mest negative reducerede omkostning*. For at vi kan benytte denne metode, skal vores lineære programmeringsproblem være på standardform. Vi benytter værktøjerne fra afsnit 3.3, hvorfor vi får følgende:

$$\begin{aligned} \text{Minimér} \quad Z = & -16x_1 + 2x_2 + x_3 - 2x_4 - 3x_5 \\ \text{Underlagt} \quad & 3x_1 + x_2 + 3x_3 - 3x_4 + 9x_5 + s_1 = 12 \\ & 2x_1 + 8x_2 + 4x_3 + 2x_4 - 4x_5 + s_2 = 10 \\ & x_1, \ldots, x_5, s_1, s_2 \geq 0 \end{aligned}$$

Vi kan nu opstille Simplex-tabellen og den 0'te række:

	x_1	x_2	x_3	x_4	x_5	s_1	s_2	
	-16	2	1	-2	-3	0	0	0
s_1	3	1	3	-3	9	1	0	12
s_2	2	8	4	2	-4	0	1	10

Den er opstillet sådan, at vi har variablene til at stå i øverste række. Under variablene har vi den 0'te række med værdier fra den reducerede omkostningsvektor $\tilde{\vec{c}}$ under den tilhørende variabel, og den negative værdi af objektfunktionen. Yderst til højre har vi den 0'te søjle, hvor den negative værdi af objektfunktionens, samt værdien af de basale variable kan aflæses. I søjlen til venstre betegnes de basale variable, som i starten er vores underskudsvariable. Disse er lig med begrænsningernes værdier. Vi husker, at de ikke-basale variable sætter vi lig med nul. Vi kan derved aflæse den basale mulige løsning til $(0, 0, 0, 0, 0, 12, 10)^T$. Objektfunktionens værdi, i dette punkt, kan vi aflæse i tabellen til at være lig med 0.

Vores formål er nu at undersøge, om vi kan bevæge os til et andet hjørnepunkt, som vil medføre en endnu lavere værdi af vores objektfunktion. Dette gør vi ved at betragte den 0'te række, hvor vi ser, om der er nogle negative indgange i $\tilde{c}^T$, da disse retninger vil reducere værdien af objektfunktionen. Vi ser, at der er tre negative indgange i $\tilde{c}^T$; $-16, -2$ og -3. Da der er flere end én, vælger vi vores pivot-søjle ud fra den mest negative. Det vil sige -16, der befinder sig i søjlen for x_1.

	x_1	x_2	x_3	x_4	x_5	s_1	s_2	
	-16	2	1	-2	-3	0	0	0
s_1	3	1	3	-3	9	1	0	12
s_2	2	8	4	2	-4	0	1	10

Vi betragter derfor søjlen under x_1, hvor vi skal vælge et pivot-element. Dette gør vi ved at udregne ratioen $\frac{x_{B(i)}}{u_i}$, hvor $x_{B(i)}$ er en basal variabel og u_i er en indgang i pivot-søjlen for alle i, hvor $u_i > 0$, som opfylder, at $\frac{x_{B(l)}}{u_l} \leq \frac{x_{B(i)}}{u_i}$. Da begge indgange i pivot-søjlen er positive, kan vi derved beregne $\frac{x_{B(i)}}{u_i}$ for begge elementer. Betragter vi nedenstående tabel har vi, at den basale variabel $x_{B(1)}$ er givet ved 12 og dens tilhørende u_1 indgang, som er 3. Dens ratio vil derved være givet ved $12/3 = 4$. Betragter vi den anden basale variabel $x_{B(2)}$, givet ved 10 og dens tilhørende u_2 indgang, som er lig med 2, ser vi, at dens ratio er lig med $10/2 = 5$. Det vil sige at rækken med den mindste ratio er 4. Vi vælger derfor den tilsvarende indgang i pivot-søjlen som vores pivot-element og laver de nødvendige rækkeoperationer for at opdatere vores tabel.

	x_1	x_2	x_3	x_4	x_5	s_1	s_2	
	-16	2	1	-2	-3	0	0	0
s_1	3	1	3	-3	9	1	0	12
s_2	2	8	4	2	-4	0	1	10

Vi noterer, at da indgangen, som vi lavede til pivot, befandt sig i søjlen under x_1 og rækken ud for s_1, sætter vi x_1 ind på s_1's plads. Det vil sige, at x_1 er vores nye basale variabel. Vi får derved følgende tabel:

	x_1	x_2	x_3	x_4	x_5	s_1	s_2	
	0	22/3	17	-18	45	16/3	0	64
x_1	1	1/3	1	-1	3	1/3	0	4
s_2	0	22/3	2	4	-10	-2/3	1	2

Vi bemærker, at værdien af objektfunktionen Z nu er reduceret til -64. Men da der stadig er en negativ koefficient i $\tilde{\tilde{c}}^T$, kan vi minimerer den yderligere, ved at gennemgå endnu en iteration. I denne iteration er den eneste negative koefficient i $\tilde{\tilde{c}}^T$ lig -18, som befinder sig i søjlen under x_4. Ser vi på de nedenstående elementer i søjlen, ved vi, at vi skal lave pivot ved skalaren 4, da det andet element er negativt.

	x_1	x_2	x_3	x_4	x_5	s_1	s_2	
	0	22/3	17	-18	45	16/3	0	64
x_1	1	1/3	1	-1	3	1/3	0	4
s_2	0	22/3	2	4	-10	-2/3	1	2

Vi laver derved igen de nødvendige rækkeoperationer, sådan at vi får pivot i den bestemte indgang. Vi husker igen, at sætte x_4 ind på s_2's plads, da der bliver lavet pivot i søjlen under x_4 og rækken ud for s_2. Vi får derved følgende tabel:

	x_1	x_2	x_3	x_4	x_5	s_1	s_2	
	0	121/3	26	0	0	7/3	9/2	73
x_1	1	13/6	3/2	0	1/2	1/6	1/4	9/2
x_4	0	11/6	1/2	1	-5/2	-1/6	1/4	1/2

Da der ikke er flere negative koefficienter, ved vi da, at objektfunktionen ikke kan minimeres mere. Vi stopper derfor algoritmen. Vi kan aflæse den optimale værdi til at være $-Z = 73$, ved den basale mulige løsning, hvis variable vil være:

$$
\begin{aligned}
x_1 &= 9/2 \\
x_2 &= 0 \\
x_3 &= 0 \\
x_4 &= 1/4 \\
x_5 &= 0 \\
s_1 &= 0 \\
s_2 &= 0
\end{aligned}
$$

6.2 Eksempel på To-Fase Simplex-metoden

Givet er et lineært programmeringsproblem:

$$
\begin{array}{ll}
\text{Minimér} & Z = 10x_1 + 6x_2 + 2x_3 \\
\text{Underlagt} & x_1 - x_2 - x_3 \leq -1 \\
 & -3x_1 - 2x_2 + x_3 \leq -2 \\
 & x_1, x_2, x_3 \geq 0
\end{array}
$$

Vi ønsker at finde en basal mulig løsning. Dette vil vi gøre ved hjælp af To-Fase Simplex-metoden. Vi starter derfor med at omskrive det lineære programmeringsproblem til standardform, hvor vi vil få følgende resultat:

$$\begin{aligned} \text{Minimér} \quad & Z = 10x_1 + 6x_2 + 2x_3 \\ \text{Underlagt} \quad & x_1 - x_2 - x_3 + s_1 = -1 \\ & -3x_1 - 2x_2 + x_3 + s_2 = -2 \\ & x_1, x_2, x_3, s_1, s_2 \geq 0 \end{aligned}$$

Da $b_i < 0$ ved begge begrænsninger, ganger vi dem igennem med -1. Dette medfører:

$$\begin{aligned} \text{Minimér} \quad & Z = 10x_1 + 6x_2 + 2x_3 \\ \text{Underlagt} \quad & -x_1 + x_2 + x_3 - s_1 = 1 \\ & 3x_1 + 2x_2 - x_3 - s_2 = 2 \\ & x_1, x_2, x_3, s_1, s_2 \geq 0 \end{aligned}$$

Vi vil nu opstille et støtte-problem ved, at vi tilføjer nogle støtte-variable $y_1, \ldots, y_m$ til vores to begrænsninger, som skal minimere objektfunktionen $Z_y = y_1 + \cdots + y_m$. Det vil sige:

$$\begin{aligned} \text{Minimér} \quad & Z_y = y_1 + y_2 \\ \text{Underlagt} \quad & -x_1 + x_2 + x_3 - s_1 + y_1 = 1 \\ & 3x_1 + 2x_2 - x_3 - s_2 + y_2 = 2 \\ & x_1, x_2, x_3, s_1, s_2, y_1, y_2 \geq 0 \end{aligned}$$

Vi har derved de basale variable til at være støtte-variable, hvor beslutnings- samt undeskudsvariablene er de ikke-basale. Vi udregner den reducerede omkostningsvektor ved $-\vec{c}_B^{\,T} A$. Vi kan nu opstille tabellen for vores støtte-problem, som ses nedenunder:

	x_1	x_2	x_3	s_1	s_2	y_1	y_2	
	-2	-2	0	1	1	0	0	-3
y_1	-1	1	1	-1	0	1	0	1
y_2	3	1	-1	0	-1	0	1	2

Vi bemærker, at værdien af Z_y er givet ved $y_1 + y_2 = 3$. Vi ønsker nu at lade støtte-variablene træde ud af basen og finde en basal mulig løsning. Metoden vi vil gøre dette på, er det samme som i det ovenstående eksempel. Vi får derved tabellen til sidst:

	x_1	x_2	x_3	s_1	s_2	y_1	y_2	
	0	0	0	0	0	1	5/3	0
x_2	0	1	1/2	-3/4	-1/4	3/4	1/4	5/4
x_1	1	0	-1/2	1/4	-1/4	-1/4	1/4	1/4

Vi kan se ud fra vores sidste tabel, at vi ikke længere har støtte-variable i vores basale mulige løsning. Den optimale værdi for objektfunktionen for støtte-problemet er da 0. Det vil sige, at vi kan fjerne støtte-variablene samt deres søjler og $\vec{x}$ er en basal mulig løsning til det oprindelige problem. Med denne kan vi nu påbegynde anden fase og dermed løse det oprindelige programmeringsproblem.

6.3 Eksempel på Simplex på Tabelform med degenererede løsninger

Givet er et lineært programmeringsproblem:

$$\begin{aligned} \text{Minimér} \quad & Z = 2x_1 + 4x_2 - 4x_3 + 7x_4 \\ \text{Underlagt} \quad & 8x_1 - 2x_2 + x_3 - x_4 \ \leq \ 50 \\ & 3x_1 + 5x_2 + 2x_3 \ \leq \ 150 \\ & x_1 - x_2 + 2x_3 - 4x_4 \ \leq \ 100 \\ & x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Vi omskriver problemet til standardform ved at tilføje underskudsvariable s_1, s_2 og s_3:

$$\begin{aligned} \text{Minimér} \quad & Z = 2x_1 + 4x_2 - 4x_3 + 7x_4 \\ \text{Underlagt} \quad & 8x_1 - 2x_2 + x_3 - x_4 + s_1 \ = \ 50 \\ & 3x_1 + 5x_2 + 2x_3 + s_2 \ = \ 150 \\ & x_1 - x_2 + 2x_3 - 4x_4 + s_3 \ = \ 100 \\ & x_1, x_2, x_3, x_4, s_1, s_2, s_3 \geq 0 \end{aligned}$$

Vi opstiller det nu på tabelform, og gennemgår samme udregningsmetode som før, hvor vi vil få følgende resultater:

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	2	4	-4	7	0	0	0	0
s_1	8	-2	1	-1	1	0	0	50
s_2	3	5	2	0	0	1	0	150
s_3	1	-1	2	-4	0	0	1	100

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	34	-4	0	3	4	0	0	200
x_3	8	-2	1	-1	1	0	0	50
s_2	3	5	0	2	0	1	0	150
s_3	-15	3	0	-2	-2	0	1	0

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	14	0	0	1/3	4/3	0	4/3	200
x_3	-2	0	1	-7/3	-1/3	0	2/3	50
s_2	28	0	0	16/3	10/3	1	-5/3	150
x_2	-5	1	0	-2/3	-2/3	0	1/3	0

Da der ikke er flere negative indgange i den 0'te række, kan vi nu aflæse den optimale værdi værende -200, som tilhører den basale mulige løsning $(0,0,50,0,0,150,0)^T$. Vi bemærker dog, at vi har opnået den samme optimale værdi to gange, som markeret i tabelen med en cirkel om objektfunktionen.

Grunden til dette er, at vi opnår den samme degenererede basale mulige løsning efter den anden pivot-operation. Dette kan ses i den anden tabel, hvor vi får nul nederst i den 0'te søjle. Vi har derfor udført én pivot-operation for meget, for at finde den optimale løsning, da den optimale løsning er degenereret.

6.4 Eksempel på at gå i ring

Givet er et lineært programmeringsproblem:

$$\begin{aligned}
\text{Minimér} \quad Z = & -10x_1 + 57x_2 + 9x_3 + 24x_4 \\
\text{Underlagt} \quad & \frac{1}{2}x_1 - \frac{11}{2}x_2 - \frac{5}{2}x_3 + 9x_4 \ \leq \ 0 \\
& \frac{1}{2}x_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3 + x_4 \ \leq \ 0 \\
& x_1 + x_2 + x_3 + x_4 \ \leq \ 1 \\
& x_1, x_2, x_3, x_4 \geq 0
\end{aligned}$$

Vi omskriver problemet på standardform ved at tilføje underskudsvariable s_1, s_2 og s_3:

$$\begin{aligned}
\text{Minimér} \quad Z = & -10x_1 + 57x_2 + 9x_3 + 24x_4 \\
\text{Underlagt} \quad & \frac{1}{2}x_1 - \frac{11}{2}x_2 - \frac{5}{2}x_3 + 9x_4 + s_1 \ = \ 0 \\
& \frac{1}{2}x_1 - \frac{3}{2}x_2 - \frac{1}{2}x_3 + x_4 + s_2 \ = \ 0 \\
& x_1 + x_2 + x_3 + x_4 + s_3 \ = \ 1 \\
& x_1, x_2, x_3, x_4, s_1, s_2, s_3 \geq 0
\end{aligned}$$

Vi opstiller nu vores problem på tabelform, hvorfor vi vil få følgende:

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	-10	57	9	24	0	0	0	0
s_1	1/2	-11/2	-5/2	9	1	0	0	0
s_2	1/2	-3/2	-1/2	1	0	1	0	0
s_3	1	1	1	1	0	0	1	1

Udfra denne tabel har vi, at den optimale værdi er 0 ved den basale mulige løsning $(0, 0, 0, 0, 0, 0, 1)^T$. Vi vil nu igen gennemgå de samme udregninger, som de forrige eksempler. Inden vi begynder med udregningerne, minder vi om, at vi benytter pivot-reglen, hvor vi betragter de mest negative reducerede omkostning. Vi vil derved få nedenstående tabeller.

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	0	-53	-41	204	20	0	0	0
x_1	1	-11	-5	18	2	0	0	0
s_2	0	4	2	-8	-1	1	0	0
s_3	0	12	6	-17	-2	0	1	1

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	0	0	-29/2	98	27/4	53/4	0	0
x_1	1	0	1/2	-4	-3/4	11/4	0	0
x_2	0	1	1/2	-2	-1/4	1/4	0	0
s_3	0	0	0	7	1	-3	1	1

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	29	0	0	-18	-15	93	0	0
x_3	2	0	1	-8	-3/2	11/2	0	0
x_2	-1	1	0	2	1/2	-5/2	0	0
s_3	0	0	0	7	1	-3	1	1

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	20	9	0	0	-21/2	141/2	0	0
x_3	-2	4	1	0	1/2	-9/2	0	0
x_4	-1/2	1/2	0	1	1/4	-5/4	0	0
s_3	7/2	-7/2	0	0	-3/4	23/4	1	1

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	-22	93	21	0	0	-24	0	0
s_1	4	8	2	0	1	-9	0	0
x_4	1/2	-3/2	-1/2	1	0	1	0	0
s_3	1/2	5/2	3/2	0	0	-1	1	1

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	-10	57	9	24	0	0	0	0
s_1	1/2	-11/2	-5/2	9	1	0	0	0
s_2	1/2	-3/2	-1/2	1	0	1	0	0
s_3	1	1	1	1	0	0	1	1

Vi bemærker at ved den sidste tabel, får vi den samme optimale værdi af objektfunktionen Z, som vi havde til at starte med, ved den samme basale mulige løsning $(0,0,0,0,0,0,1)^T$. Det vil sige, at vi nu går i ring, og at vi ikke vil komme frem til en optimal løsning, hvis vi fortsætter med samme metode. Der findes dog en metode, som løser et sådan problem, som nævnt i afsnit 5.3, nemlig Blands regel. Der skal vi vælge det første indeks, hvor $\tilde{c}_j < 0$. Desuden skal variablen, som træder ud af basen, også have det mindste indeks. Betragter vi den andensidste tabel, ser vi, at hvis vi benytter Blands regel, vil pivot-søjlen være søjlen under x_1, da det er det mindste indeks j, som opfylder $\tilde{c}_j < 0$. Dette er markeret med en cirkel i nedenstående tabel. Da ratioen for begge af elementerne x_4 og s_2 er det samme, vil pivot-rækken dermed være uden for rækken af x_4, da x_4's indeks er mindre end s_2's.

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	-22	93	21	0	0	-24	0	0
s_1	4	8	2	0	1	-9	0	0
x_4	1/2	-3/2	-1/2	1	0	1	0	0
s_3	1/2	5/2	3/2	0	0	-1	1	1

Vi vil i dette tilfælde have den ikke basale variabel x_1 til at tiltræde basen og den basale variabel x_4 til at træde ud, hvor vi før havde, at s_2 tiltrådte basen. Regner vi videre fra dette trin, vil vi få følgende resultater:

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	0	27	-1	44	0	20	0	0
s_1	0	-4	-2	8	1	-1	0	0
x_1	1	-3	-1	2	0	2	0	0
s_3	0	4	2	-1	0	-2	1	1

	x_1	x_2	x_3	x_4	s_1	s_2	s_3	
	0	29	0	87/2	0	19	1/2	1/2
s_1	0	0	0	7	1	-3	1	1
x_1	1	-1	0	3/2	0	1	1/2	1/2
x_3	0	2	1	-1/2	0	-1	1/2	1/2

Vi kan nu se ud fra vores sidste tabel, at vi ikke længere kan minimere objektfunktionen, da der ikke er flere negative værdier i den reducerede omkostningsvektor. Den optimale værdi er derved $-1/2$ ved den basale mulige løsning $(1/2,0,1/2,0,1,0,0)^T$. På den måde har vi, ved brug af Blands regel, undgået at gå i ring.

7 | Diskussion og perspektivering

Simplex-metoden har siden sin udvikling vist sig så brugbar, at den i dag anvendes i mange forskellige sammenhænge i industrien, og der er flere forskellige programmer, som har en implementation af Simplex-metoden indbygget. Endvidere er Simplex-metoden let at anvende for folk uden matematisk baggrund. Der findes derfor også i dag mange hjemmesider med 'Simplex-beregnere' og pakker til Python, Excel osv., som kan løse lineære programmeringsproblemer ved hjælp af Simplex-metoden.

Simplex-metoden er dog ikke perfekt. Vi gennemgår ikke kompleksiteten af de forskellige implementationer af Simplex-metoden, samt effekten af valg af pivotregel i dette projekt. Dykker man ned i kompleksiteten viser Simplex-metoden sig at være ineffektiv til at løse store problemer med mange variable. Den bruges derfor kun på relativt simple problemer. Ved meget store problemer, vil man ofte bruge indre-punkts-metoder i stedet, da de har lavere kompleksitet. Hvor Simplex-metoden bevæger sig fra hjørnepunkt til hjørnepunkt langs randen af polyederet, kigger indre-punkts-metoder efter en optimal løsning ved at bevæge sig inden for polyederet. Dette har så den svaghed, at man ikke altid finder den optimale løsning i et hjørnepunkt men en løsning meget tæt på - i modsætning til Simplex-metoden. Simplex-metoden vil altså i nogle tilfælde finde en løsning, som er en lille smule bedre. Dette er dog ofte en ubetydelig forskel.

En anden problemstilling er, hvor ofte virkelige problematikker kan beskrives ved optimering af en lineær funktion underlagt lineære begrænsninger. Når man modellerer virkeligheden med matematik, skal man næsten altid lave nogle afgrænsninger. Dog kan modellen blive mere eller mindre præcis efter valg af model. F.eks. hvis man vil optimere sin kost, vil det give mening at tage hensyn til, hvordan samspillet mellem de forskellige næringsstoffer påvirker optagelsen. F.eks. bidrager vitamin C til en mere effektiv optagelse af jern. Sådanne effekter kan være svære at medregne i lineær programmering. Mange problemer kan altså beskrives mere præcist ved en ikke-lineær funktion. Endvidere kan mange problemer kun beskrives ved hjælp af heltal. F.eks. kan en fabrik ikke lave en halv bil, eller ansætte en kvart medarbejder.

Til videre undersøgelse kunne man derfor kigge på indre-punkts-metoder, ikke-lineær programmering og heltalsprogrammering, da man derfor vil have et bredere spektrum af kompetencer til at kunne beskrive og løse virkelige problemer mere præcist. Man kunne også undersøge kompleksiteten af de forskellige implementationer af Simplex-metoden, samt hvordan og hvorfor nogle pivotregler påvirker kompleksiteten. Her er særligt Blands regel interessant, da den gør, at man undgår at gå i ring samt øger hastigheden af Simplex-metoden.

8 Konklusion

Vi har i projektet defineret et arbitrært lineært programmeringsproblem som et problem, hvor man ønsker at maksimere eller minimere en given lineær funktion, kaldet *objektfunktionen* Z på formen $\vec{c} \cdot \vec{x} = Z$, bestående af flere variable. Denne objektfunktion er da underlagt et antal lineære begrænsninger. Disse begrænsninger findes i arbitrære lineære programmeringsproblemer på formen $\vec{a}_i \cdot \vec{x}(\leq, =, \geq)b_i$.
Da disse optimeringsproblemer kan bestå af både ligheder og uligheder, kan disse blive problematiske at løse. Derfor har vi også vist, at disse kan omskrives til ækvivalente lineære programmeringsproblemer på standardform, hvor begrænsningerne kun består af ligheder og den ikke-negative begrænsning. Denne omskrivning sker, ved at tilføje de såkaldte underskuds- og/eller overskudsvariable.
Arbitrære lineære programmeringsproblemer har ikke et krav om, at dets beslutningsvariable er begrænsede. Disse kan dog omskrives til standard maksimerings- eller minimeringsproblemer, sådan at objektfunktionen også underlægges den ikke-negative begrænsning, hvilket er et krav på standardform. En ubegrænset beslutningsvariabel fjernes ved at omskrive den til differencen af to kunstige variable, hvor de to nye kunstige variable er underlagt den ikke-negative begrænsning. Beslutningsvariable, der er underlagt den ikke-positive begrænsning, ganges da igennem med -1.

Vi kan betragte et lineært programmeringsproblems mulighedsmængde, som en geometrisk figur, bestående af hyperplaner og fællesmængden af halvrum. Disse figurer har vi defineret som konvekse polyedre, hvor halvrummene betegner det lineære programmeringsproblems hovedbegrænsninger.
Da vi kigger på lineære programmeringsproblemer på standardform, betyder dette, at de basale løsninger skal opfylde ligheder, og dermed ligge på randen af polyedret, hvilket betyder, at hovedbegrænsningerne er aktive. Dette giver os resultatet, at lineære programmeringsproblemer på standardform har n lineært uafhængige begrænsninger, og at der findes en entydig løsning. Hvis denne løsning derudover også overholder alle begrænsningerne i et lineært programmeringsproblem på standardform, siges den at være en basal mulig løsning.
Hvis disse betingelser skal kunne opfyldes, betyder det, at den basale mulige løsning skal være et hjørnepunkt, samt et kritisk punkt. Vi finder derfor ud af, at hvis vi skal finde en optimal løsning, skal vi undersøge hjørnepunkterne i polyederet.

Da vi nu ved, at vi skal finde den optimale løsning til et lineært programmeringsproblem på standardform i et hjørnepunkt i dets tilhørende polyeder, ønsker vi at finde en metode, hvor vi bevæger os mellem hjørnepunkter og undersøger disse.
Vi har derfor defineret en mulig retning $\vec{d}$ kaldet en basal retning fra en basal mulig løsning $\vec{x}$, videre til en anden basal mulig løsning i polyederet. For at bevæge os i den j'te basale

retning $\vec{d}$, går vi mod løsningen $\vec{x} + \theta\vec{d}$ ved at øge værdien af θ så meget som muligt.
Vi definerer da også en reduceret omkostning $\tilde{c}_j$, som ændringen af værdien af objektfunktionen, når vi bevæger os i den j'te basale retning. Her beviser vi, at hvis $\tilde{\vec{c}} \geq \vec{0}$, så er den basale mulige løsning optimal, da objektfunktionen ikke kan reduceres yderligere. Hvis dette ikke er tilfældet, ønsker vi da at bevæge os hen til et andet hjørnepunkt for at undersøge, om den reducerede omkostningsvektor her er ikke-negativ. Her er det nødvendigt at bevæge os så langt som muligt. Til dette bruger vi skalaren θ^*, som skalerer den basale retning, uden at bringe os ud af polyederet.
De basale søjler, som er lineært uafhængige, hvis indeks er løsningens base, opstilles i den invertible $m \times m$-matrix B, basismatricen. Når vi bevæger os hen til den nye basale mulige løsning, træder den basale variabel $x_{B(l)}$, valgt ved udregningen af θ^*, ud af basen. I stedet tiltræder den førvalgte ikke-basale variabel x_j, idet den nu har en positiv værdi. Da udskifter vi indeks $B(l)$ til j i basismatricen, og vi får da en ny basismatrix $\tilde{B}$. Denne metode kaldes Simplex-metoden.

Denne metode kan opskrives som en iteration i en algoritme, kaldet den naive Simplex-metode. Denne algoritme vil stoppe efter et endeligt antal interationer, idet enhver basal mulig løsning er ikke-degenereret.
Hvis vi i stedet har et problem, med degenererede løsninger, er der risiko for, at metoden vil gå i ring. Vi får derfor brug for pivotregler til at afgøre, hvilke indeks der skal tiltræde og fratræde basen. Her ser vi først på de to klassiske pivot-regler, hvor vi enten vælger den mest negative reducerede omkostning, eller det indeks, hvor man får den største reduktion af omkostningen. Disse to regler har dog den ulempe, at ingen af dem kan sikre os, at algoritmen ikke går i ring. Hvis vi derimod vælger Blands regel, hvor man blot vælger det mindste indeks, hvor $\tilde{c}_j < 0$, kan dette undgås.

I en anden algoritme, kaldet den Reviderede Simplex-metode, undgår vi at udregne B^{-1} i starten af hver iteration. I stedet udregnes $\tilde{B}^{-1}$ i slutningen af hver iteration, ved at udføre elementære rækkeoperationer på den oprindelige basismatrix' inverse B^{-1}, ved brug af produktet af en følge af elementærmatricer, Q. Algoritmen starter på samme måde som den naive, dog bestemmer vi her en $1 \times m$-matrix K, til at udregne den reducerede omkostning. På denne måde undgår vi, at udregne $\vec{c}_B^T B^{-1}$ i hver iteration. Til sidst i iterationen omskrives søjlen $\vec{u} = B^{-1}\vec{A}_j$ til standardvektoren $\vec{e}_l$, og den inverse til den nye basismatrix, $\tilde{B}^{-1}$ kan findes ved at gange Q på fra venstre.

Det kan blive besværligt at benytte den Reviderede Simplex-metode i hånden, idet man skal holde styr på mange forskellige matricer. I stedet kan det være behjælpeligt at bruge Simplex på Tabelform, som løbende opdater matricen $B^{-1}[A \mid \vec{b}\,]$. Her betragter vi de reducerede omkostninger i den 0'te række, hvor vi vælger et indeks j, hvor $\tilde{c}_j < 0$. Vektoren $\vec{u}$ kaldes nu pivot-søjlen. Når alle de reducerede omkostninger er ikke-negative, kan vi aflæse værdien af objektfunktionen og variablene ud fra tabellen.

Hvis vi ikke på forhånd kender en basal mulig løsning, kan vi benytte To-Fase Simplex-metoden. Her opstiller man et støtteproblem, hvor nogle støtte-variable er tilføjet, som objektfunktionen afhænger af. Ved at anvende Simplex-metoden til at løse støtteproblemet, finder man en basal mulig løsning til det oprindelige problem, hvis det oprindelige problem er løseligt.

Vi kan altså løse lineære programmeringsproblemer med forskellige implementeringer af Simplex-metoden. Valget af metode afgøres af, om vi som udgangspunkt kender det første hjørnepunkt, som vi kan starte algoritmen ud fra, og dermed systematisk kan undersøge resten af polyederets andre hjørnepunkter, for at finde den optimale løsning. Med dette har vi nu beskrevet, hvordan vi ved hjælp af den bagvedliggende geometri i lineære programmeringsproblemer, kan udlede en algoritmisk løsningsmetode.

9 | Bibliografi

[1] Luenberger DG, Ye Y. Linear and Nonlinear Programming. International Series in Operations Research & Management Science. Springer International Publishing; 2015. Tilgængelig fra: https://books.google.dk/books?id=_VwICgAAQBAJ.

[2] Bertsimas D, Tsitsiklis JN. Introduction to Linear Optimization. Massachusetts Institute of Technology; 1997.

[3] Olav Geil. Elementary Linear Algebra 2015. 2nd ed. Pearson Education Limited; 2015.

[4] Thie PR. An introduction to linear programming and game theory. No. vb. 57;vb. 5774 in Wiley international edition. Wiley; 1979. Tilgængelig fra: https://books.google.dk/books?id=WmdRAAAAMAAJ.

[5] University of Kwazulu-Natal. Linear Programming;. Tilgængelig fra: http://www.maths.unp.ac.za/coursework/MATH331/2012/linearprogramming.pdf.

[6] Ferguson TS. Linear programming. In: Handbook of Approximation Algorithms and Metaheuristics; 2007. .

[7] Beck A. Convex Optimization. In: Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics; 2014. .

[8] Poulsen ET. Funktioner af en og flere variable: indledning til matematisk analyse. Gad; 2001. Tilgængelig fra: https://books.google.dk/books?id=VmflAAAACAAJ.