
Optimering og lineær programmering

PROJEKTRAPPORT P2
GRUPPE A403



Aalborg Universitet

Første Studieår v/ Det Tekniske-Naturvidenskabelige Fakultet
Strandvejen 12-14 • 9000 Aalborg



AALBORG UNIVERSITET
STUDENTERRAPPORT

Basis

Matematik
Badehusvej
9000 Aalborg
<http://math.aau.dk>

Titel:

Optimering

Projekt:

P2-projekt

Projektperiode:

Februar - maj 2019

Projektgruppe:

A403

Deltagere:

Benjamin Roger Ninn Fasken
Don Andreas Pape Jørgensen
Oscar Dalgaard
Nomaan Khuwaja
Richard í Dali
Rasmus Buus Simonsen
Violeta Constantinescu

Vejleder:

Anne Marie Svane
Henrik Worm Routhé

Sidetæl: 0 + Formalia, appendiks og bilag.

Afsluttet ??

Abstract:

This paper gives an introduction to linear programming, which is a method for solving linear optimization problems. Geometrically, the constraints of the problem bound a polytope in \mathbb{R}^n . The possible solutions to a linear optimization problem lie within this feasible region, and if the problem has an optimal solution, this solution is always found at a vertex of the feasible region. The simplex algorithm utilizes this fact by working through the basic feasible solutions of the problem in the direction of optimization. The least index implementation of the algorithm avoids cycling and is guaranteed to find an optimal solution if such a solution exists. Simplex has exponential worst-case complexity, but in actual use it will typically terminate in linear time. Our implementation of the simplex algorithm in Python 3.6 is attached as an appendix.

Rapportens indhold er frit tilgængeligt, men offentliggørelse (med kildeangivelse) må kun ske efter aftale med forfatterne.

Forord

Dette P2-projekt er udarbejdet i perioden 04/02/19 – 24/05/19 af en gruppe på seks 2. semesters-matematikstuderende og en sidefagsstuderende. Projektet behandler lineær optimering og programmering. Det pålagte perspektiv forholder sig især til de geometriske egenskaber, et lineært optimeringsproblem kan siges at have, og i denne forbindelse fremhæves og gennemgås simplex-metoden. Projektet beskæftiger sig ikke med et reelt scenario eller problem, men tjener alene til at give et eksempel på brugen af simplex-metoden og dermed validere det teoretiske grundlag. Vi vil gerne takke vores vejleder Anne Marie Svane og bi-vejleder Henrik Worm Routhe, der begge har været behjælpelige med henholdsvis det faglige og gruppesamarbejdet. Hvis ikke andet, har dette projekt i hvert fald været en lærerig oplevelse på både det faglige- og samarbejds-mæssige plan. God læselyst.

Underskrifter

Benjamin Roger Ninn Fasken

Don Andreas Pape Jørgensen

Oscar Dalgaard

Nomaan Khuwaja

Richard í Dali

Rasmus Buus Simonsen

Violeta Constantinescu

Indholdsfortegnelse

Forord	i
1 Indledning	1
2 Problemformulering	2
3 Lineær Algebra	3
3.1 Vektorer og Matricer	3
3.1.1 Addition og multiplikation af matricer med en skalar	4
3.1.2 Transponerede matricer	5
3.1.3 Augmenterede matricer	5
3.2 Vektorer	6
3.2.1 Matrix-multiplikation	7
3.3 Lineære ligninger	8
3.3.1 Matrix-repræsentation	9
3.4 Løsningsmetode til systemer af lineære ligninger	10
3.4.1 Elementære rækkeoperationer	10
3.4.2 Trappeform og reduceret trappeform	11
3.4.3 Pivot-indgange	11
3.4.4 Gauss-elimination	12
3.5 Inverse matricer	13
3.5.1 Elementære matricer	14
3.5.2 Invertibilitet og løsning af lineære ligninger	15
3.5.3 Lineær afhængighed og uafhængighed	16
3.5.4 Span, underrum, basis og dimension	18
4 Lineær Programmering	20
4.1 Optimering	20
4.1.1 Generel definition	21
4.1.2 Standardform	22
4.2 Dualitet	23
4.3 Geometri	27
4.3.1 Konvekse mængder	28
4.3.2 Ekstremumpunkter	30
4.3.3 Algebraisk definition	31
4.4 Polytoper på standardform	35
4.5 Degenerering	38

4.6	Simplex-algoritmen	39
4.6.1	Simplex-tabellen	39
4.6.2	Pivotering	40
4.6.3	Auxiliære problemer	43
4.6.4	Shortcut-metoden	45
4.6.5	Ubegrænsethed	49
4.6.6	Pseudokode	49
4.6.7	Cykling	50
4.6.8	Simplex-eksempel	53
4.6.9	Kompleksitet	57
4.6.10	Dual simplex-metoden	57
4.6.11	Resten af beviset for Dualitetssætningen	57
4.7	Implementering af simplex-algoritmen	60
5	Konklusion	63
6	Bilag	64
6.1	Appendiks	64
7	Bibliografi	68

1 | Indledning

Lineær programmering er en matematisk metode til at maksimere eller minimere en lineær objektfunktion under en række betingelser. Metoden blev brugt af Leonid Kantorovich og Wassily Leontief i 1930'erne, men det var især under Anden Verdenskrig, at lineær programmering blev anvendt til planlægning og optimering af logistiske opgaver (*Linear programming* [2019]). I 1947 introducerede George Danzig simplex-metoden, som gjorde det langt lettere at løse optimeringsproblemer.

Objektfunktionen for et maksimeringsproblem kan skrives som:

$$z = c_1x_1, \dots, c_nx_n,$$

og er underlagt en række bibetingelser:

$$\begin{aligned} a_{1,1}x_1 + \dots + a_{1,n}x_n &\leq b_1 \\ &\vdots \\ a_{m,1}x_1 + \dots + a_{m,n}x_n &\leq b_m, \end{aligned}$$

samt ikke-negativitetsbetingelserne, $x_1, \dots, x_n \geq 0$. Alle a , b og c 'erne er konstanter, som repræsenterer kapacitet, omkostning, behov, profit eller andre relevante faktorer for det givne problem. Den grundlæggende antagelse er, at alle forhold er lineære. Løsningen på problemet er en vektor, $\mathbf{x}^T = [x_1, \dots, x_n]$, som giver den optimale (f.eks. maksimale udbytte eller minimale spild eller omkostning) værdi af z .

Geometrisk set afgrænser optimeringsproblemets betingelser et polytop i \mathbb{R}^n . Den optimale løsning kan altid findes i et af polytopets hjørner, og simplex-algoritmen fungerer, kort fortalt, ved at arbejde sig igennem polytopets hjørner i optimerende retning, indtil den optimale løsning findes. I værste fald har algoritmen eksponentiel kompleksitet, men i praksis er simplex effektiv og vil typisk finde løsningen i lineær tid.

Lineær programmering anvendes indenfor mange brancher. Industriel produktion, varetransport, mødeplanlægning og trafikafvikling kan optimeres herved, eller der kan benyttes lineær programmering til at finde en optimal perfekt parring som led i Christofides algoritme, når en løsning til Den Handelsrejsendes Problem approksimeres.

2 | Problemformulering

Hvad er simplex-metoden, og hvordan kan den anvendes til at løse lineære optimeringsproblemer? For at besvare dette spørgsmål, vil vi:

- Gennemgå relevant lineær algebra og indføre dertilhørende terminologi.
- Afklare, hvad lineær programmering består i, og herunder undersøge den tilsvarende geometri.
- Undersøge simplex-metoden og illustrere anvendelsen af denne med et eksempel, samt implementere algoritmen i Python.

3 | Lineær Algebra

For at kunne behandle et optimeringsproblem, er det først nødvendigt at redegøre for grundlæggende lineær algebra og indføre essentiel notation, hvilket dette kapitel tjener til. Det er ikke alle sætninger, som vil blive bevist. Dette skyldes flere årsager: Først og fremmest er det ikke projektets primære formål at eftervise teorien bag lineær algebra. Derudover er mange af beviserne relativt simple. Sidst, men ikke mindst, et ønske om at spare plads. Kapitlet tager udgangspunkt i Spence et al. [2015] og Bertsimas and Tsitsiklis [1997].

3.1 Vektorer og Matricer

Definition 3.1.1. Matricer

En $m \times n$ matrix, A , er en rektangulær tabel af tal, hvor $m \times n$ beskriver dimensionerne af matricen. En $m \times n$ matrix indeholder i alt $m \cdot n$ værdier, også kaldet skalarer, opstillet i m rækker og n søjler. Skalaren på i 'te række og j 'te søjle kaldes (i, j) -indgangen i matricen, og betegnes $a_{i,j}$, hvor $1 \leq i \leq m$ og $1 \leq j \leq n$.

Her ses en matrix, A :

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

Den j 'te søjle i A betegnes $\mathbf{s}_{A,j} = \begin{bmatrix} a_{1,j} \\ a_{2,j} \\ \vdots \\ a_{m,j} \end{bmatrix}$,

og matricens i 'te række er $\mathbf{r}_{A,i} = \begin{bmatrix} a_{i,1} & a_{i,2} & \cdots & a_{i,n} \end{bmatrix}$.

To matricer, A og B , siges at være ens, hvis de har samme antal rækker og søjler samt $a_{i,j} = b_{i,j}$ for alle i og j . Dette skrives $A = B$.

Nedenfor er angivet et udsnit af forskellige typer af matricer:

- En *kvadratisk matrix* er en matrix med dimensionerne $n \times n$, dvs. antallet af rækker er lig antallet af søjler.
- En *diagonalmatrix*, D , er en matrix, hvor indgangene, $d_{i,j}$, med $i \neq j$, udenfor diagonalen, alle er lig 0.
- En *identitetsmatrix* er en diagonalmatrix, hvor indgangene langs diagonalen er 1. En $n \times n$ identitetsmatrix betegnes I_n .
- En *nulmatrix* er en matrix, hvor alle indgange er lig 0. Den betegnes O .

Her illustreres den generelle form for en diagonalmatrix, D , og en identitetsmatrix, I_n :

$$D = \begin{bmatrix} d_{1,1} & 0 & \cdots & 0 \\ 0 & d_{2,2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & d_{n,n} \end{bmatrix} \quad I_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

3.1.1 Addition og multiplikation af matricer med en skalar

To matricer må adderes, hvis de begge har samme dimensioner, $m \times n$. Desuden kan enhver konstant, $c \in \mathbb{R}$, ganges på en matrix, ved at alle indgange i matricen ganges med c .

$$\begin{aligned} (A \pm B)_{i,j} &= a_{i,j} \pm b_{i,j} \\ (c A)_{i,j} &= c a_{i,j} \end{aligned}$$

Der gælder følgende regneregler for regneoperationerne addition af matricer og multiplikation med skalar:

Sætning 3.1.2. *Egenskaber for regneregler*

Lad A , B og C være $m \times n$ matricer, og lad a, b og c være vilkårlige skalarer. Da gælder:

- $A + B = B + A$
- $(A + B) + C = A + (B + C)$
- $A + O = A$
- Til enhver matrix, A , findes en modsat matrix: $-A = (-1) \cdot A$, så $A + (-A) = O$
- $c(A + B) = cA + cB$
- $(a + b)A = aA + bA$

Der eksisterer beviser for ovenstående egenskaber, men eftersom disse består af sammensætninger af simple regneoperationer som addition, subtraktion og multiplikation, vil de ikke blive vist.

3.1.2 Transponerede matricer

Definition 3.1.3. *Transponeret matrix*

Lad A være en $m \times n$ matrix. Den transponerede matrix, betegnet A^T , opnås ved at lave rækkerne i matricen om til søjler og søjlerne om til rækker. Indgangen $(a^T)_{j,i}$ i A^T svarer således til $a_{i,j}$ i A . Hvor A er en $m \times n$ matrix, er A^T en $n \times m$ matrix.

En matrix kaldes *symmetrisk*, hvis $A = A^T$, og *skævsymmetrisk*, hvis $A = -A^T$. Betragt matricerne på Figur 3.1. Matrix A er ikke symmetrisk, da $A \neq A^T$. Matrix B er derimod symmetrisk, da matricen ikke ændres ved at blive transponeret. Matrix C er skævsymmetrisk, eftersom den modsatte transponerede matrix, $-C^T$, er lig den originale matrix, C .

$$A = \begin{bmatrix} 3 & 2 & 4 \\ 5 & 1 & 2 \end{bmatrix} \quad A^T = \begin{bmatrix} 3 & 5 \\ 2 & 1 \\ 4 & 2 \end{bmatrix} \quad B = \begin{bmatrix} 3 & 2 \\ 2 & 1 \end{bmatrix} = B^T \quad C = \begin{bmatrix} 0 & 2 \\ -2 & 0 \end{bmatrix} = -C^T$$

Figur 3.1: Matricerne A , B , C , og deres transponerede.

Sætning 3.1.4. *Regneregler for transponering*

Lad A og B være $m \times n$ matricer, og lad c være en vilkårlig skalar. Da gælder:

- $(A + B)^T = A^T + B^T$
- $(cA)^T = cA^T$
- $(A^T)^T = A$

Sætningen ovenfor viser, at transponering kommuterer med matrix addition og skalar multiplikation. Regnereglerne vil ikke blive bevist af samme årsag som ved Sætning 3.1.2.

3.1.3 Augmenterede matricer

Definition 3.1.5. *Augmenteret matrix*

Lad A og B være matricer med samme antal rækker. Den augmenterede matrix af B på A , betegnet $[A \ B]$, er en matrix med søjlerne af A efterfulgt af søjlerne af B .

Nedenfor ses to matricer, A og B , samt den augmentedede matrix af B på A :

$$A = \begin{bmatrix} 6 & 2 \\ 5 & 4 \end{bmatrix} \quad B = \begin{bmatrix} 1 & 2 & 3 \\ 5 & 4 & 3 \end{bmatrix} \quad [A \ B] = \begin{bmatrix} 6 & 2 & 1 & 2 & 3 \\ 5 & 4 & 5 & 4 & 3 \end{bmatrix}$$

(a) Matricerne, A og B (b) B augmentedet på A

3.2 Vektorer

Definition 3.2.1. Vektorer

En $m \times 1$ matrix kaldes en søjlevektor, og en $1 \times n$ matrix kaldes en rækkevektor. Den i 'te-indgang i \mathbf{u} betegnes u_i .

I det følgende, hvis ikke andet er angivet, vil der ved vektor menes en søjlevektor.

En vektor, hvori alle elementer har værdien 0, kaldes *nulvektoren* \mathbf{o} .

Da en vektor er en speciel form for matrix, kan regneoperationer, såsom addition eller multiplikation af matricer med en skalar, stadig benyttes, men i stedet kaldes det *vektor addition* og *skalar multiplikation*. Disse regneoperationer følger også reglerne i Sætning 3.1.2.

Betragt vektorerne, $\mathbf{v}^T = [4 \ 7 \ 3 \ 2 \ 11]$ og $\mathbf{u}^T = [2 \ 10 \ 4 \ 5 \ 1]$. Ved at anvende skalar multiplikation beregnes $2\mathbf{v}$ ved at gange hver indgang, v_i , med 2. Det vil sige, $2\mathbf{v}^T = [8 \ 14 \ 6 \ 4 \ 22]$. Summen af vektorerne, $2\mathbf{v}$ og \mathbf{u} , beregnes ved at addere alle indgange, $2v_i + u_i$, således $2\mathbf{v} + \mathbf{u} = [10 \ 24 \ 10 \ 9 \ 23]^T$.

En *linearkombination* af vektorerne, $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3, \dots, \mathbf{u}_k$, er en vektor, der kan skrives på formen, $\mathbf{v} = c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \dots + c_k\mathbf{u}_k$, hvor c_1, c_2, \dots, c_k er skalarer. Nedenfor ses eksempelvis, at vektoren, $\mathbf{v} = [3, 4, 5]^T$, er en linearkombination af vektorerne, $\mathbf{u}_1 = [2, 4, 2]^T$ og $\mathbf{u}_2 = [1, 1, 2]^T$, hvor $c_1 = 1/2$ og $c_2 = 2$.

$$\begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} = 1/2 \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$$

Standardvektoren, \mathbf{e}_i , i \mathbb{R}^n defineres som en vektor med 1 i i 'te indgang og 0 i de resterende indgange.

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \dots, \mathbf{e}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}.$$

Enhver vektor i \mathbb{R}^n kan skrives som linearkombination af standardvektorerne, som følger:

$$\mathbf{v} = v_1 \mathbf{e}_1 + v_2 \mathbf{e}_2 + \dots + v_n \mathbf{e}_n.$$

3.2.1 Matrix-multiplikation

Definition 3.2.2. Matrix-vektorprodukt

Lad A være en $m \times n$ matrix og \mathbf{v} en $n \times 1$ vektor. Matrix-vektorproduktet af A og \mathbf{v} er vektoren, der fremkommer som en linearkombination af søjlerne i A og skalarene i \mathbf{v} på formen:

$$A\mathbf{v} = v_1 \mathbf{s}_{A,1} + v_2 \mathbf{s}_{A,2} + \dots + v_n \mathbf{s}_{A,n}$$

Som eksempel kan 2×3 matrixen A ganges med 3×1 vektoren \mathbf{v} . Produktet er 2×1 vektoren $A\mathbf{v}$:

$$A = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 1 & 4 \end{bmatrix} \quad \mathbf{v} = \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix}$$

$$A\mathbf{v} = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 \\ -1 \\ 2 \end{bmatrix} = 3 \begin{bmatrix} 4 \\ 2 \end{bmatrix} - 1 \begin{bmatrix} 2 \\ 1 \end{bmatrix} + 2 \begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} 12 \\ 13 \end{bmatrix}.$$

På samme måde som en matrix kan ganges med en vektor, kan produktet af to matrixer også tages, hvis blot den første matrix har samme antal søjler, som den anden matrix har rækker.

Definition 3.2.3. Matrixprodukt

Lad A være en $m \times n$ matrix og B en $n \times p$ matrix. Produktet, AB , er den $m \times p$ matrix, som har $A\mathbf{s}_{B,j}$ som sin j 'te søjle:

$$AB = [A\mathbf{s}_{B,1} \quad A\mathbf{s}_{B,2} \quad \dots \quad A\mathbf{s}_{B,p}]$$

For eksempel ganges 2×3 matrixen A på 3×2 matrixen $B = \begin{bmatrix} 3 & 0 \\ -1 & 2 \\ 2 & 3 \end{bmatrix}$, og produktet er

2×2 matricen:

$$AB = \begin{bmatrix} 4 & 2 & 1 \\ 2 & 1 & 4 \end{bmatrix} \begin{bmatrix} 3 & 0 \\ -1 & 2 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} 12 & 7 \\ 13 & 14 \end{bmatrix}.$$

Matrixmultiplikation er ikke kommutativ. Det vil sige, at der ikke nødvendigvis gælder $AB = BA$.

3.3 Lineære ligninger

En *lineær ligning* er en ligning på formen:

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

hvor x_1, x_2, \dots, x_n er ligningens variable, og a_1, a_2, \dots, a_n er koefficienter, der, ligesom b , er reelle tal.

Et *system af lineære ligninger* er et sæt af m lineære ligninger i de samme variable, x_1, \dots, x_n , hvor m og n er positive heltal. Systemet skrives:

$$\begin{aligned} a_{1,1}x_1 + a_{1,2}x_2 + \dots + a_{1,n}x_n &= b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \dots + a_{2,n}x_n &= b_2 \\ &\vdots \\ a_{m,1}x_1 + a_{m,2}x_2 + \dots + a_{m,n}x_n &= b_m. \end{aligned}$$

Løsningen til dette system af lineære ligninger er en $n \times 1$ vektor svarende til antallet af variable, sådan at alle m ligninger er opfyldt. Ethvert lineært ligningssystem har enten ingen løsning, præcis én løsning eller uendeligt mange løsninger.

Et ligningssystem, som har én eller flere løsninger, kaldes *konsistent*, ellers kaldes det *inkonsistent*. Som eksempel betragtes følgende tre ligninger med tre ubekendte:

$$\begin{aligned} 4x_1 + 5x_2 - x_3 &= 13 \\ -3x_1 + x_2 + 2x_3 &= 12 \\ -x_1 + x_2 + x_3 &= 8. \end{aligned}$$

Ovenstående ligningssystem har præcis én løsning: $x_1 = 1, x_2 = 3, x_3 = 6$.

Hvis et ligningssystem har mere end én løsning, har det uendeligt mange løsninger, og i et sådant tilfælde vil der i løsningsmængden optræde en eller flere såkaldt *frie variable*, som bruges til at udtrykke de øvrige grundlæggende variable, også kaldet *bundne variable*.

Betragt følgende system:

$$\begin{aligned}4x_1 - 2x_2 - 5x_3 - 10x_4 &= -5 \\x_2 + x_3 - x_4 &= 2 \\2x_1 - x_3 - 4x_4 &= 1.\end{aligned}$$

Systemet har uendeligt mange løsninger, og den *generelle løsning* skrives:

$$\begin{aligned}x_1 &= 2 + 2x_4 \\x_2 &= -1 - x_4 \\x_3 &= 3 \\x_4 &= x_4.\end{aligned}$$

At $x_4 = x_4$ vil sige, at variablen, x_4 , kan antage en hvilken som helst værdi, og x_4 er dermed en fri variabel. Nogle af de øvrige variable er udtrykt vha. den frie variabel. For at få en specifik løsning til systemet, kan en værdi for x_4 fastsættes, og dermed bliver de bundne variable, x_1 , x_2 og x_3 , også bestemt. Følgende ligningssystem er et eksempel på et inkonsistent system:

$$\begin{aligned}3x_1 + 2x_2 + 2x_3 &= 9 \\x_1 + x_2 + x_3 &= 4 \\x_1 - 2x_2 - 2x_3 &= 1.\end{aligned}$$

Ligningerne kan løses enkeltvis eller parvis, men det er ikke muligt at finde værdier, som kan opfylde alle tre ligninger på én gang.

3.3.1 Matrix-repræsentation

Systemer af lineære ligninger kan repræsenteres vha. matricer, og løsningen skrives som en vektor. Betragt følgende ligningssystem:

$$\begin{aligned}4x_1 + 5x_2 - x_3 &= 13 \\-3x_1 + x_2 + 2x_3 &= 12 \\-x_1 + x_2 + x_3 &= 8.\end{aligned}$$

Systemet kan repræsenteres ved *matrix-ligningen* $A\mathbf{x} = \mathbf{b}$, hvor A kaldes *koefficientmatricen*, og er i dette tilfælde:

$$\begin{bmatrix} 4 & 5 & -1 \\ -3 & 1 & 2 \\ -1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 13 \\ 12 \\ 8 \end{bmatrix}.$$

Den augmentedede matrix, $[A \ \mathbf{b}]$, også kaldet *totalmatricen*, bliver:

$$[A \ \mathbf{b}] = \begin{bmatrix} 4 & 5 & -1 & 13 \\ -3 & 1 & 2 & 12 \\ -1 & 1 & 1 & 8 \end{bmatrix}.$$

Totalmatricen for det generelle ligningssystem er:

$$\begin{bmatrix} a_{1,1} & \cdots & a_{1,n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m,1} & \cdots & a_{m,n} & b_m \end{bmatrix}.$$

Løsningen til et ligningssystem findes vha. *rækkeoperationer*. Via *Gauss-elimination* kan totalmatricen omdannes til en anden form, kaldet *trappeform*, og så yderligere til *reduceret trappeform*. I næste afsnit bliver det gennemgået og illustreret med et eksempel.

3.4 Løsningsmetode til systemer af lineære ligninger

Elementære rækkeoperationer benyttes til at finde løsninger til ligningssystemer, hvorfor disse operationer vil blive gennemgået i det følgende afsnit.

3.4.1 Elementære rækkeoperationer

Ved *skalar multiplikation* af en række, $\mathbf{r}_{A,i}$, ganges en skalar, c , hvor $c \in \mathbb{R}$, $c \neq 0$, med $\mathbf{r}_{A,i}$. Dette skrives som $r_i \rightarrow cr_i$, og det svarer til at gange den tilsvarende lineære ligning med en skalar.

Rækkeaddition tillader en række, $\mathbf{r}_{A,i}$, ganget med en skalar, c , at blive lagt til rækken, $\mathbf{r}_{A,j}$, hvilket skrives som $r_j \rightarrow r_j + cr_i$. Hvis en matrix eksempelvis indeholder rækken $[a_{i,1} \ a_{i,2} \ \dots \ a_{i,n}]$ og $[a_{j,1} \ a_{j,2} \ \dots \ a_{j,n}]$, så lægges c gange den i 'te række til den j 'te række, og dermed erstattes $\mathbf{r}_{A,j}$ med $[c a_{i,1} + a_{j,1} \ c a_{i,2} + a_{j,2} \ \dots \ c a_{i,n} + a_{j,n}]$. Dette svarer til at lægge c gange den tilsvarende lineære ligning i til ligning j .

Ved *rækkeombytning* byttes en række, $\mathbf{r}_{A,i}$, med $\mathbf{r}_{A,j}$. Dette skrives $r_i \leftrightarrow r_j$. De to rækker bytter plads i matricen uden ellers at ændres. Rækkeombytning svarer til at bytte rækkefølgen på ligningerne.

Hvis en række i matricen udelukkende består af 0'er (evt. som følge af rækkeoperationer), kaldes det for en *nulrække*. På samme måde benævnes en søjle, hvor alle indgange er 0, som en *nulsøjle*.

Alle rækkeoperationerne er naturligvis reversible. Hvis for eksempel rækken $\mathbf{r}_{A,i}$ multipliceres med skalar c , kan rækken blot ganges med $1/c$ for at omgøre operationen.

Betragt matricen nedenfor, hvor der bliver anvendt hver af de forskellige rækkeoperationer. Først ganges r_1 med en skalar, $c = 2$, derefter lægges r_3 ganget med en skalar, $c = 3$ til rækken r_2 , og til sidst ombyttes rækkerne r_3 og r_2 :

$$\begin{bmatrix} 3 & 4 & 9 \\ 2 & 4 & 10 \\ 7 & 8 & 3 \end{bmatrix} \xrightarrow{r_1 \rightarrow 2r_1} \begin{bmatrix} 6 & 8 & 18 \\ 2 & 4 & 10 \\ 7 & 8 & 3 \end{bmatrix} \xrightarrow{r_2 \rightarrow r_2 + 3r_3} \begin{bmatrix} 6 & 8 & 18 \\ 23 & 28 & 19 \\ 7 & 8 & 3 \end{bmatrix} \xrightarrow{r_3 \leftrightarrow r_2} \begin{bmatrix} 6 & 8 & 18 \\ 7 & 8 & 3 \\ 23 & 28 & 19 \end{bmatrix}.$$

Det gælder for alle de rækkeoperationer, der kan udføres på en matrix, at de ikke ændrer på løsningsmængden for det tilsvarende lineære ligningssystem. Rækkeoperationerne bruges således først og fremmest til at gøre systemet enklere og mere overskueligt, så eventuelle løsninger er nemmere at aflæse.

3.4.2 Trappeform og reduceret trappeform

Definition 3.4.1. *Trappeform/reduceret trappeform*

En matrix siges at være på trappeform, hvis og kun hvis:

- 1 Eventuelle nulrækker er nederst i matricen.
- 2 Den første indgang i en række, der ikke er lig 0, ligger til højre for første indgang i rækken ovenfor, der ikke er lig 0.

En matrix siges at være på reduceret trappeform, hvis den er på trappeform og:

- 3 Den første indgang, der ikke er lig 0, i en række er 1.
- 4 En sådan indgang er den eneste indgang, der ikke er lig 0, i den pågældende søjle.

Enhver totalmatrix, A , bibeholder samme løsningsmængde, selvom den er på reduceret trappeform. Beviset for dette kommer i Afsnit 3.5.1.

3.4.3 Pivot-indgange

Den første indgang fra venstre forskellig fra 0 i en række i en matrix, A , på trappeform kaldes en *pivot-indgang*. En søjle, som indeholder en pivot-indgang, kaldes en *pivot-søjle*, og en række, som indeholder en pivot-indgang, kaldes en *pivot-række*.

For en vilkårlig matrix, som ikke er på trappeform, er det umiddelbart svært at se hvilke indgange, der er pivot-indgange. Først når matricen er på trappeform eller reduceret trappeform, er det nemt at aflæse, hvor der er pivot i matricen.

En pivot-søjle angiver en bunden variabel i det tilsvarende ligningssystem, mens en søjle, som ikke indeholder en pivot, angiver en fri variabel i ligningssystemet. *Rangen* af en

matrix angiver antallet af pivot-søjler og skrives $\text{rang}(A)$. *Nulliteten* angiver antallet af søjler, der ikke er pivot-søjler, og skrives $\text{null}(A)$.

3.4.4 Gauss-elimination

Gauss-elimination er en algoritme til at få en matrix på trappeform, eller eventuelt reduceret trappeform, vha. elementære rækkeoperationer. Algoritmen har følgende trin:

- 1 Find ikke-nul-søjlen længst mod venstre. Den øverste indgang i denne pivot-søjle er en pivot-indgang.
- 2 Hvis pivot-indgangen har værdi 0, ombyttes rækken med en vilkårlig række, hvor indgangen i pivot-søjlen er forskellig fra 0, og som ikke er over pivot-rækken.
- 3 Ved hjælp af rækkeaddition ændres hver indgang under pivot-indgangen til 0.
- 4 Fra nu af ignoreres pivot-rækken og alle rækker over denne. Hvis der fortsat er en eller flere ikke-nulrækker, som ikke ignoreres, gentages trin 1-4 på den resterende delmatrix.

Når trin 1-4 er gennemført, er matricen på trappeform. Hvis matricen skal bringes på reduceret trappeform tilføjes følgende trin, hvor der arbejdes nedefra og op. Dvs. der begynder med nederste ikke-nulrække og arbejdes opad:

- 5 Hvis den første indgang, som er forskellig fra 0, er forskellig fra 1, ændres denne til 1 vha. skalar multiplikation. Herefter ændres alle indgange over pivot-indgangen til 0 vha. rækkeaddition.
- 6 Gentag trin 5 på rækken ovenfor, indtil matricen har opnået reduceret trappeform.

En illustration af Gauss-eliminationsalgoritmen gives ved at bearbejde et eksempel. Dette bruges også til forklaring af proceduren for løsningsmetoden til et ligningssystem. Det tidligere ligningssystem, som har totalmatricen $[A \ b]$, betragtes. Bemærk, at der er brugt en horisontal linje for adskillelse, når der anvendes to rækkeoperationer:

Vha. rækkeoperationer og den nævnte algoritme reduceres $[A \ b]$ til den reducerede trappeform, $[R \ c]$. Allerede når totalmatricen er på trappeform, kan det ses, hvor mange løsninger ligningssystemet har. Hvis $[R \ c]$ indeholder en række, hvor den eneste indgang forskellig fra 0 ligger i sidste søjle, dvs. at der er pivot i sidste søjle, så er ligningssystemet inkonsistent. Ellers har systemet mindst én løsning og er dermed konsistent. Endvidere gælder det, at hvis der er pivot-indgange i alle søjler, på nær den sidste, er der en entydig løsning og dermed ingen frie variable. Er der ikke pivot i alle søjler, er der derimod uendelig mange løsninger og dermed en eller flere frie variable.

For at finde løsningen, skrives ligningssystemet, der svarer til matricen, $[R \ c]$, og løses for de bundne variable i forhold til de frie variable for at få den generelle løsning.

$$\begin{aligned}
[A \ b] &= \begin{bmatrix} 4 & 5 & -1 & 13 \\ -3 & 1 & 2 & 12 \\ -1 & 1 & 1 & 8 \end{bmatrix} \quad r_3 \leftrightarrow -r_1 \quad \begin{bmatrix} 1 & -1 & -1 & -8 \\ -3 & 1 & 2 & 12 \\ 4 & 5 & -1 & 13 \end{bmatrix} \\
\frac{r_2 \rightarrow r_2 + 3r_1}{r_3 \rightarrow r_3 - 4r_1} &\quad \begin{bmatrix} 1 & -1 & -1 & -8 \\ 0 & -2 & -1 & -12 \\ 0 & 9 & 3 & 45 \end{bmatrix} \quad \frac{r_2 \rightarrow \frac{-1}{2}r_2}{r_3 \rightarrow r_3 + \frac{9}{2}r_2} \quad \begin{bmatrix} 1 & -1 & -1 & -8 \\ 0 & 1 & 1/2 & 6 \\ 0 & 0 & -3/2 & -9 \end{bmatrix} \\
r_3 \rightarrow \frac{-2}{3}r_3 &\quad \begin{bmatrix} 1 & 0 & -1/2 & -2 \\ 0 & 1 & 1/2 & 6 \\ 0 & 0 & 1 & 6 \end{bmatrix} \quad \frac{r_2 \rightarrow \frac{-1}{2}r_3 + r_2}{r_1 \rightarrow r_3 + \frac{1}{2}r_2} \quad \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 6 \end{bmatrix}
\end{aligned}$$

I dette konkrete eksempel er der ikke pivot i sidste søjle, og dermed er det afgjort, at der er tale om et konsistent ligningssystem. Da alle de andre tre søjler er pivot-søjler, kan det ses, at der ikke er frie variable, og dermed at der er en entydigt løsning, nemlig $\mathbf{x} = [1 \ 3 \ 6]^T$.

3.5 Inverse matricer

Definition 3.5.1. *Inverse matricer*

En $n \times n$ matrix, A , siges at være invertibel, hvis der er en kvadratisk matrix, B , sådan at

$AB = BA = I_n$. I så fald siges det, at B er invers til A og skrives også A^{-1} .

Enhver invertibel matrix har en entydig invers matrix. Hvis matricerne, B og C , begge er inverse til A , sådan at $AB = BA = I_n$, og $AC = CA = I_n$, da gælder:

$$B = BI_n = B(AC) = (BA)C = I_n C = C.$$

Her blev der benyttet, at matrixproduktet af enhver matrix og identitetsmatricen giver matricen selv. Nedenfor er der eksempelvis givet en invertibel matrix, B , og dens inverse, A :

$$B = \begin{bmatrix} 1 & 3 \\ 1 & 2 \end{bmatrix}, A = \begin{bmatrix} -2 & 3 \\ 1 & -1 \end{bmatrix}.$$

A er invers til B , da:

$$AB = BA = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I_2.$$

Følgende regneregler gælder for inverse matricer:

Sætning 3.5.2.

Lad A og C være invertible $n \times n$ matricer:

- a) A^{-1} er invertibel, og $(A^{-1})^{-1} = A$.
- b) AC er invertibel, og $(AC)^{-1} = C^{-1}A^{-1}$.
- c) A^T er invertibel, og $(A^T)^{-1} = (A^{-1})^T$.

3.5.1 Elementære matricer

De elementære rækkeoperationer redegjort for i Afsnit 3.4.1 kan også udføres ved matrixmultiplikation.

For eksempel: skalar multiplikation med en skalar, c , på 2. række i en matrix, $A = \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix}$,

svarer til:

$$E_1 A = \begin{bmatrix} 1 & 0 \\ 0 & c \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ ca_{2,1} & ca_{2,2} \end{bmatrix}.$$

Rækkeaddition, hvor c gange 1. række lægges til 2. række i A , svarer til:

$$E_2 A = \begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} = \begin{bmatrix} a_{1,1} & a_{1,2} \\ ca_{1,1} + a_{2,1} & ca_{1,2} + a_{2,2} \end{bmatrix}.$$

Rækkeombytning af 1. og 2. række i A :

$$E_3 A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \end{bmatrix} = \begin{bmatrix} a_{2,1} & a_{2,2} \\ a_{1,1} & a_{1,2} \end{bmatrix}.$$

Matricerne, E_1 , E_2 og E_3 , er eksempler på *elementære matricer*, hvilket fører videre til en formel definition:

Definition 3.5.3. Elementær matrix

En $n \times n$ matrix, A , er elementær, hvis og kun hvis I_n kan opnås ved at udføre en enkelt rækkeoperation på A . En elementær matrix betegnes E .

En matrix, A , kan dermed bringes på reduceret trappeform, R , ved multiplikation med elementære matricer:

$$E_k E_{k-1} \cdots E_1 A = R$$

Produktet af de elementære matricer $E_k E_{k-1} \cdots E_1$ kan udtrykkes som en matrix P , og da elementære matricer per definition er invertible, og et produkt af invertible matricer også er invertibelt ifølge Sætning 3.5.2, fører det frem til følgende:

Sætning 3.5.4.

For en matrix, A , med reduceret trappeform R , eksisterer der en invertibel matrix, P , hvor $PA = R$.

Bevis. Da P er invertibel, kan ligningerne $PA = R$, og $A\mathbf{x} = \mathbf{b}$ skrives: $A = P^{-1}R$, og $\mathbf{b} = P^{-1}\mathbf{c}$.

Efter at dette er på plads, antages det, at \mathbf{v} er en løsning til $A\mathbf{x} = \mathbf{b}$. Så er:

$$R\mathbf{v} = (PA)\mathbf{v} = P\mathbf{b} = \mathbf{c}.$$

Ligeledes antages nu, at \mathbf{v} er en løsning til $R\mathbf{x} = \mathbf{c}$. Da er:

$$A\mathbf{v} = (P^{-1}R)\mathbf{v} = P^{-1}\mathbf{c} = \mathbf{b}.$$

Dermed er det bevist, at løsningsmængderne for $A\mathbf{x} = \mathbf{b}$ og $R\mathbf{x} = \mathbf{c}$ er ens. □

3.5.2 Invertibilitet og løsning af lineære ligninger

Sætning 3.5.5.

En matrix, A , er invertibel, hvis og kun hvis $R = I_n$.

Eftersom sætningen er en biimplikation, opdeles udsagnet i to, der begge skal være sande. Først vises, at hvis A er invertibel, så gælder $R = I_n$. Dernæst vises, at hvis $R = I_n$, så er A invertibel.

Bevis. Antag først, at $n \times n$ matricen A er invertibel. Lad \mathbf{v} være en vektor i \mathbb{R}^n , sådan at $A\mathbf{v} = \mathbf{o}$. Der ganges A^{-1} på, så $\mathbf{v} = A^{-1}\mathbf{o} = \mathbf{o}$. Dermed er $\mathbf{v} = \mathbf{o}$ den eneste løsning til $A\mathbf{v} = \mathbf{o}$. Dvs. at $[A \ \mathbf{o}]$ reduceres til $[R \ \mathbf{o}]$, hvor der i R findes pivot i hver søjle. Men det at have en entydig løsning, er ækvivalent med at have pivot i hver søjle, hvilket vil sige, at $\text{rang}(A) = n$. En matrix på reduceret trappeform med n pivot-søjler er I_n .

Antag nu, at den reducerede trappeform af A er $R = I_n$. Fra Sætning 3.5.4 vides det, at der er en invertibel matrix, P , sådan at $PA = I_n$. Så er:

$$A = I_n A = (P^{-1}P)A = P^{-1}(PA) = P^{-1}I_n = P^{-1}.$$

Da P er invertibel, er P^{-1} også, og dermed er A invertibel. □

Dette gør det muligt at opstille en effektiv metode til at bestemme, om A er invertibel, og dermed også til at finde A^{-1} . Anvend elementære rækkeoperationer på $[A \ I_n]$ til den reducerede trappeform $[R \ B]$. Hvis $R = I_n$, medfører det, at A er invertibel, og at B er invers til A . Derimod, hvis $R \neq I_n$, medfører det, at A ikke er invertibel.

For at illustrere metoden, anvendes den nedenfor på 2×2 matricen A . Først er I_2 blevet augmenteret på A :

$$A = \begin{bmatrix} 1 & 3 \\ 1 & 2 \end{bmatrix} \quad [A \ I_2] = \begin{bmatrix} 1 & 3 & 1 & 0 \\ 1 & 2 & 0 & 1 \end{bmatrix}.$$

Dernæst udføres rækkeoperationerne, $r_2 \rightarrow r_2 - r_1$, $r_1 \rightarrow r_1 + 3r_2$ og $r_2 \rightarrow -r_2$, sådan at formen $[R \ B]$ opnås:

$$[R \ B] = \begin{bmatrix} 1 & 0 & -2 & 3 \\ 0 & 1 & 1 & -1 \end{bmatrix}.$$

Da $R = I_2$, er A invertibel, og følgende er derfor gældende:

$$B = A^{-1} = \begin{bmatrix} -2 & 3 \\ 1 & -1 \end{bmatrix}.$$

Inverse matricer kan også bruges til at finde løsningen til et lineært ligningssystem, $A\mathbf{x} = \mathbf{b}$, hvor A er invertibel. Ved at gange den inverse til A på, fås:

$$\begin{aligned} A^{-1}(A\mathbf{x}) &= A^{-1}\mathbf{b} \\ (A^{-1}A)\mathbf{x} &= A^{-1}\mathbf{b} \\ I_n\mathbf{x} &= A^{-1}\mathbf{b} \\ \mathbf{x} &= A^{-1}\mathbf{b}. \end{aligned}$$

Eksempelvis betragtes det lineære ligningssystem, $A\mathbf{x} = \mathbf{b}$, hvor matricen, A , er fra det sidste eksempel, og $\mathbf{b} = \begin{bmatrix} 4 \\ 8 \end{bmatrix}$. Ligningen omskrives til $\mathbf{x} = A^{-1}\mathbf{b}$ og løses:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -2 & 3 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 4 \\ 8 \end{bmatrix} = \begin{bmatrix} 16 \\ -4 \end{bmatrix}.$$

Denne måde at løse lineære ligningssystemer er dog ikke så generel som metoden beskrevet i Afsnit 3.4, da den er afhængig af A 's invertibilitet.

3.5.3 Lineær afhængighed og uafhængighed

Undervejs kan det blive nødvendigt at udtrykke en af søjlerne i en matrix som en linearkombination af de andre søjler. For at bestemme om dette overhovedet er muligt for en given matrix, vil lineær afhængighed og uafhængighed blive introduceret i dette afsnit.

Definition 3.5.6. Lineær uafhængighed

En mængde af k vektorer, $\mathbf{u}_1, \dots, \mathbf{u}_k$, i \mathbb{R}^n kaldes *lineært uafhængige*, hvis ligningen $c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \dots + c_k\mathbf{u}_k = \mathbf{o}$ kun har den trivielle løsning $c_1 = c_2 = \dots = c_k = 0$.

Omvendt kaldes en mængde af k vektorer $\mathbf{u}_1, \dots, \mathbf{u}_k$ *lineært afhængige*, hvis der findes koefficienter c_1, c_2, \dots, c_k , der ikke alle er 0, sådan at $c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \dots + c_k\mathbf{u}_k = \mathbf{o}$.

Ligningen, $c_1\mathbf{u}_1 + c_2\mathbf{u}_2 + \dots + c_k\mathbf{u}_k = \mathbf{o}$, kan også skrives som et matrix-vektorprodukt, $[\mathbf{u}_1, \dots, \mathbf{u}_k][c_1, c_2, \dots, c_k]^T = \mathbf{o}$. Vektorerne, $\mathbf{u}_1, \dots, \mathbf{u}_k$, er dermed lineært afhængige, hvis der findes en løsning \mathbf{x} , hvor $\mathbf{x} \neq \mathbf{o}$.

En hvilken som helst af vektorerne kan dog ikke nødvendigvis udtrykkes som en linearkombination af de andre vektorer.

Når en matrix er reduceret til trappeform, kan det uden videre aflæses, om matrixens søjlevektorer er lineært afhængige. Hvis matricen har pivot i alle søjler, er det naturligvis umuligt at udtrykke en given søjlevektor som en linearkombination af matrixens øvrige søjler. Søjlerne er dermed lineært uafhængige, og det tilsvarende ligningssystem har en entydig løsning.

Har matricen derimod ikke pivot i alle søjler, kan nogle af vektorerne udtrykkes som linearkombinationer af hinanden. Hvis matricen er på reduceret trappeform, kan det uden videre ses, hvordan ikke-pivot-søjlerne er linearkombinationer af pivot-søjlerne. Betragt som eksempel følgende matrix:

$$A = \begin{bmatrix} 1 & -1 & 2 & -3 \\ 3 & -2 & 6 & -8 \\ -2 & 1 & -3 & 5 \end{bmatrix},$$

som har den reducerede trappeform:

$$R = \begin{bmatrix} 1 & 0 & 0 & -2 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

Af den reducerede form kan det uden videre ses, at $\mathbf{s}_{R,4} = -2\mathbf{s}_{R,1} + \mathbf{s}_{R,2}$. Eftersom løsningsmængden er uændret af rækkeoperationerne, er forholdet mellem søjlerne bevaret, og denne linearkombination gælder derfor også for matricen, A , således:

$$\mathbf{s}_{A,4} = -2\mathbf{s}_{A,1} + \mathbf{s}_{A,2} = -2 \begin{bmatrix} 1 \\ 3 \\ -2 \end{bmatrix} + \begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix} = \begin{bmatrix} -3 \\ -8 \\ 5 \end{bmatrix}.$$

Sætning 3.5.7.

En $m \times n$ matrix kan maksimalt have $\min\{m, n\}$ lineært uafhængige søjler/rækker.

Bevis. Betragt en $m \times n$ matrix A . Hvis $n \geq m$, kan A maksimalt have m lineært uafhængige søjler, da der højst kan være m pivot-indgange i matrixen. Ligeledes kan der, hvis $m \geq n$, højst være n lineært uafhængige søjler. \square

3.5.4 Span, underrum, basis og dimension

Før at undersøgelsen af lineær optimering og simplex-metoden endelig kan påbegyndes, følger her den sidste sektion af definitioner og sætninger.

Definition 3.5.8. *Span*

Lad $S = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$ være en ikke-tom mængde af vektorer i \mathbb{R}^n . Mængden af alle linearkombinationer af $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ betegnes som *spannet af S* eller $\text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$.

Søjlerummet af en matrix, A , er spannet af søjlerne i A , og tilsvarende er *rækkerummet* spannet af rækkerne i A .

Dvs. at rækkerummet således svarer til søjlerummet af den transponerede matrix.

Definition 3.5.9. *Underrum*

En delmængde, H , af vektorer i \mathbb{R}^n kaldes et *underrum af \mathbb{R}^n* , hvis det opfylder følgende tre betingelser:

- (a) $\mathbf{o} \in H$.
- (b) Hvis $\mathbf{v}, \mathbf{u} \in H$, så skal $\mathbf{v} + \mathbf{u} \in H$.
- (c) Hvis c er en skalar, og $\mathbf{u} \in H$, så $c\mathbf{u} \in H$.

Sætning 3.5.10.

Spannet af en endelig ikke-tom delmængde af \mathbb{R}^n er et underrum i \mathbb{R}^n .

Eller sagt på anden vis: Hvis $\mathbf{u}_1, \dots, \mathbf{u}_k$ er vektorer i \mathbb{R}^n , så er $H = \text{span}\{\mathbf{u}_1, \dots, \mathbf{u}_k\}$ et underrum i \mathbb{R}^n .

I forlængelse af Sætning 3.5.10 og Definition 3.5.9 kvalificerer en mængde af vektorer, $V = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$, sig som et underrum af sig selv, idet $V \in \text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\}$, og sammen med \mathbf{o} siges disse to underrum at være *trivielle*, fordi de er indlysende.

Definition 3.5.11. *Basis for underrum*

Vektorene $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ kaldes en *basis for et underrum, H* , hvis:

- (a) $\text{span}\{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k\} = H$.
- (b) $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ er lineært uafhængige.

En basis (flertal; baser) er således en mængde, der spanner et underrum, hvor mængden indeholder så få vektorer som muligt.

Definition 3.5.12. Dimension

Antallet af vektorer i en basis for et ikke-nul underrum, H , af \mathbb{R}^n , kaldes dimensionen af H .

Sætning 3.5.13.

Lad $S = \{\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p\}$ være en mængde af vektorer. Antag, at spannet af S har dimensionen m . Da gælder:

- (a) Der eksisterer en basis for S bestående af m af vektorerne, $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_p$.
- (b) Hvis $k \leq m$, og $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ er lineært uafhængige, kan en basis for S dannes ved at starte med $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_k$ og dernæst vælge $m - k$ af vektorerne $\mathbf{u}_{k+1}, \dots, \mathbf{u}_p$.

Bevis. Kun (b) vil blive bevist, eftersom (a) er et særtilfælde af (b), hvor $k = 0$. Hvis hver vektor, $\mathbf{u}_{k+1}, \dots, \mathbf{u}_p$, kan udtrykkes som en linearkombination af $\mathbf{u}_1, \dots, \mathbf{u}_k$, så er enhver vektor i spannet af $\mathbf{u}_1, \dots, \mathbf{u}_p$ også en linearkombination af $\mathbf{u}_1, \dots, \mathbf{u}_k$, og sidstnævnte vektorer danner en basis, hvor $k = m$. Hvis ikke dette er gældende, er mindst en af vektorerne, $\mathbf{u}_{k+1}, \dots, \mathbf{u}_p$, lineært uafhængig fra $\mathbf{u}_1, \dots, \mathbf{u}_k$. Ved at vælge sådan en vektor, haves nu $k + 1$ af vektorerne $\mathbf{u}_1, \dots, \mathbf{u}_p$, som er lineært uafhængige. Ved at gentage denne proces $m - k$ gange, fås den ønskede basis af S . \square

4 | Lineær Programmering

I kapitlet defineres et optimeringsproblem, og det vises, hvordan simplex-metoden løser optimeringsproblemet, hvis der findes en løsning. Kapitlet tager udgangspunkt i Bertsimas and Tsitsiklis [1997], hvis ikke andet er angivet.

4.1 Optimering

Lineær programmering handler i bund og grund om optimeringen af en funktion under hensynstagen til en række betingelser. Selvom dette arbitrært kunne overføres til en række virkelighedsnære problemer såsom effektiviseringen af en bestemt produktionslinje, eller f.eks. udbringelsen af et antal varer til forskellige lokationer hurtigst muligt, er det ikke principielt nødvendigt at beskæftige sig med, hvad de forskellige variable repræsenterer, for at arbejde med lineær programmering. At der er tale om *lineær* programmering vil sige, at både den funktion, som ønskes optimeret, samt de begrænsninger eller bibetingelser, der skal opfyldes, er lineære.

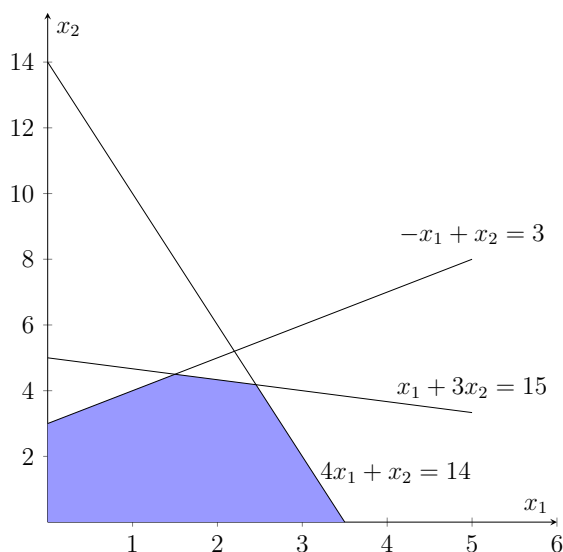
Den funktion, som ønskes maksimeret eller minimeret, kaldes *objektfunktionen*. Objektfunktionens maksimum (evt. minimum) kaldes problemets *optimale værdi*.

Opgaven kan f.eks. bestå i at maksimere funktionen $f(x_1, x_2) = x_1 + 2x_2$, som er objektfunktionen, hvor der haves *ikke-negativitetsbetingelserne* $x_1 \geq 0$ og $x_2 \geq 0$, under hensyntagen til *bibetingelserne*:

$$\begin{aligned}4x_1 + x_2 &\leq 14 \\x_1 + 3x_2 &\leq 5 \\-x_1 + x_2 &\leq 3.\end{aligned}$$

I dette tilfælde skal funktionen altså maksimeres under betingelse af fem uligheder. Formuleret mere i tråd med et generelt lineært programmeringsproblem, så skal: *objektfunktionen optimeres under bibetingelserne og ikke-negativitetsbetingelserne*.

Et simpelt problem, som det ovenstående, kan illustreres gennem en geometrisk tolkning. Der er kun to variable, så problemet kan skitseres i \mathbb{R}^2 -planen. De to ikke-negativitetsbetingelser betyder, at der kun arbejdes i første kvadrant:



Det blå område, som er markeret på tegningen ovenfor, angiver, hvor betingelserne er opfyldt, og er dermed det område, som løsningen på maksimeringsproblemet skal findes indenfor.

4.1.1 Generel definition

I et generelt lineært optimeringsproblem forsøges det at optimere en objektfunktion, $\mathbf{c}^T \mathbf{x} = \sum_{i=1}^n c_i x_i$, hvor $\mathbf{c}^T = (c_1, c_2, \dots, c_n)$, over alle n -dimensionelle vektorer, $\mathbf{x}^T = (x_1, x_2, \dots, x_n)$, betinget af de førnævnte bibetingelser og ikke-negativitetsbetingelser, som er en mængde af lineære *ligheds-* og *ulighedsbetingelser*.

Lad M_1 , M_2 og M_3 være endelige indeks-mængder, og for hvert i i hver af disse mængder, gives der en n -dimensionel vektor, \mathbf{a}_i , og en skalar, b_i , som indgår i den i 'te betingelse. Lad også N_1 og N_2 være delmængder af $1, 2, \dots, n$, som angiver, hvilke variable, x_j , som er begrænset til at være henholdsvis ikke-negative eller ikke-positive. Betragt nu det generelle problem:

$$\begin{array}{ll} \text{Maksimér} & \mathbf{c}^T \mathbf{x} \\ \text{betinget af} & \mathbf{a}^T \mathbf{x} \geq b_i, \quad i \in M_1 \\ & \mathbf{a}^T \mathbf{x} \leq b_i, \quad i \in M_2 \\ & \mathbf{a}^T \mathbf{x} = b_i, \quad i \in M_3 \\ & x_j \geq 0, \quad j \in N_1 \\ & x_j \leq 0, \quad j \in N_2 \end{array}$$

En vektor, \mathbf{x} , som opfylder alle betingelserne kaldes en *mulig løsning* eller en *mulig vektor*. Mængden af alle mulige løsninger kaldes *mulighedsområdet*. Opgaven i et såkaldt *maksimeringsproblem* består i at finde en vektor, der maksimerer værdien af objektfunktionen indenfor mulighedsområdet. Ved et *minimeringsproblem* ønskes det derimod at

finde en vektor, som mindsker værdien af objektfunktionen mest muligt. En mulig vektor, der giver objektfunktionens optimale værdi, kaldes en *optimal løsning/vektor* (eller blot *optimal*).

Et maksimeringsproblem siges at være *ubegrænset*, hvis objektfunktionen kan antage arbitrært store funktionsværdier. Ellers kaldes problemet *begrænset*. Omvendt er minimeringsproblemet ubegrænset, hvis objektfunktionen kan antage vilkårligt små værdier. Ellers er det begrænset. Et problem er *konsistent*, hvis mulighedsområdet ikke er tomt, og *inkonsistent*, hvis det er tomt.

4.1.2 Standardform

Bibetingelserne i det tidligere bearbejdede eksempel er skrevet som en lineær ulighed, $A\mathbf{x} \leq \mathbf{b}$, hvilket er en form, der ofte vil forekomme i projektet. En anden fremstillingsform er *standardformen*. Det er denne form, problemet skal antage, for at problemet kan løses ved anvendelse af simplexmetoden, som vil blive redegjort for i Afsnit 4.6.

Definition 4.1.1. Maksimering/minimering på standardform

Et maksimeringsproblem siges at være skrevet på standardform, når det er skrevet på formen:

$$\begin{aligned} \text{Maksimer } z &= \mathbf{c}^T \mathbf{x} \\ \text{betinget af } A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Ligeledes er et minimeringsproblem på standardform skrevet som:

$$\begin{aligned} \text{Minimer } z &= \mathbf{c}^T \mathbf{x} \\ \text{betinget af } A\mathbf{x} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Hvor z kaldes *objektvariablen*, og maksimeringsproblemet på standardform kan omskrives til et minimeringsproblem ved at multiplicere \mathbf{c} med -1 .

Ethvert lineært programmeringsproblem kan omskrives til at være på standardform. Mængden af mulige løsninger forbliver den samme. For at et problem kan komme på standardform, skal følgende opfyldes:

- 1) Enhver bibetingelse i optimeringsproblemet skal være en lighed. Hvis der er angivet en bibetingelse på formen, $\sum_{j=1}^n a_{i,j}x_j \leq b_i$, introduceres en *slack-variabel*, $x_{n+i} \geq 0$, sådan at bibetingelsen kan omskrives til $\sum_{j=1}^n a_{i,j}x_j + x_{n+i} = b_i$. Ligeledes omskrives en bibetingelse på formen $\sum_{j=1}^n a_{i,j}x_j \geq b_i$ vha. en *overskudsvariabel*, $x_{n+i} \geq 0$, til $\sum_{j=1}^n a_{i,j}x_j - x_{n+i} = b_i$. Omvendt kan en lighedsbetingelse omskrives til to ulighedsbetingelser, sådan at ligheden, $\sum_{j=1}^n a_{i,j}x_j = b_i$, omskrives til $\sum_{j=1}^n a_{i,j}x_j \leq b_i$ og $\sum_{j=1}^n a_{i,j}x_j \geq b_i$.

- 2) Alle variable, x_j , hvor $j \in \{1, \dots, n\}$, skal være betinget af en ikke-negativitetsbetingelse. Hvis x_i ikke er betinget af en ikke-negativitetsbetingelse, omskrives den ved: $x_j = x_j^+ - x_j^-$, hvor $x_j^+, x_j^- \geq 0$.

Betragt følgende eksempel på et maksimeringsproblem, der ønskes omskrevet, således det kommer på standardform:

$$\begin{aligned} &\text{Maksimér } x_1 + 2x_2 + x_3 \\ &\text{betinget af } x_1 + x_2 + 3x_3 \leq 14 \\ &\quad 2x_1 + 3x_2 + x_3 \geq 20 \\ &\quad x_1, x_3 \geq 0 \end{aligned}$$

Da bibetingelserne er formuleret som uligheder, skal disse først omskrives til ligheder.

Dette gøres ved at introducere en ikke-negativ slack-variabel, x_4 , og en ikke-negativ overskudsvariabel, x_5 . Variablen x_2 , er ikke betinget af en ikke-negativitetsbetingelse, hvorfor x_2 omskrives til $x_2^+ - x_2^-$, hvor $x_2^+, x_2^- \geq 0$. Nedenfor er det oprindelige maksimeringsproblem skrevet på standardform:

$$\begin{aligned} &\text{Maksimér } x_1 + 2(x_2^+ - x_2^-) + x_3 \\ &\text{betinget af } x_1 + (x_2^+ - x_2^-) + 3x_3 + x_4 = 14 \\ &\quad 2x_1 + 3(x_2^+ - x_2^-) + x_3 - x_5 = 20 \\ &\quad x_1, x_2^+, x_2^-, x_3, x_4, x_5 \geq 0 \end{aligned}$$

4.2 Dualitet

Som nævnt, har alle lineære optimeringsproblemer til formål at maksimere eller minimere en objektfunktion under problemets bibetingelser. For at løse optimeringsproblemer er dualitetsbegrebet et stærkt værktøj. For ethvert lineært problem, også kaldet *primal-problem*, findes en tilhørende *dual*. Afsnittet tager udgangspunkt i Ferguson [2017] og Chiarandini [2015].

Definition 4.2.1. Primal-/dual-problem

Et primalt maksimeringsproblem:

$$\begin{array}{ll} \text{Maksimer} & z = \mathbf{c}^T \mathbf{x} \\ \text{betinget af} & A\mathbf{x} \leq \mathbf{b} \\ & \mathbf{x} \geq \mathbf{0}, \end{array}$$

har en dual, som er minimeringsproblemet:

$$\begin{array}{ll} \text{Minimer} & w = \mathbf{b}^T \mathbf{y} \\ \text{betinget af} & A^T \mathbf{y} \geq \mathbf{c} \\ & \mathbf{y} \geq \mathbf{0}. \end{array}$$

Maksimeringsproblemet er ligeledes også dualen til minimeringsproblemet.

For eksempel, maksimeringsproblemet:

$$\begin{array}{ll} \text{Maksimer} & x_1 + 2x_2 \\ \text{betinget af} & 4x_1 + x_2 \leq 14 \\ & x_1 + 3x_2 \leq 5 \\ & -x_1 + x_2 \leq 3 \\ & x_1, x_2 \geq 0, \end{array}$$

har følgende dual:

$$\begin{array}{ll} \text{Minimer} & 14y_1 + 5y_2 + 3y_3 \\ \text{betinget af} & 4y_1 + y_2 - y_3 \geq 1 \\ & y_1 + 3y_2 + y_3 \geq 2 \\ & y_1, y_2, y_3 \geq 0. \end{array}$$

Nedenfor vil der blive set på relationen mellem primære problemer og deres dualer.

Sætning 4.2.2. Svag dualitet

Hvis \mathbf{x} er en mulig vektor til primalproblemet og \mathbf{y} en mulig vektor til dualen, da gælder

$$\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}.$$

Bevis. I minimeringsproblemet angives begrænsningerne, $\mathbf{c} \leq A^T \mathbf{y}$, og $\mathbf{y} \geq 0$. I maksimeringsproblemet angives begrænsningerne, $A\mathbf{x} \leq \mathbf{b}$, og $\mathbf{x} \geq 0$. Derved kan det ses, at

$$\mathbf{c}^T \mathbf{x} \leq A^T \mathbf{y} \mathbf{x} \leq \mathbf{b}^T \mathbf{y}.$$

□

Korollar 4.2.3.

Hvis både primalproblemet samt dualen er konsistente, så er begge problemer begrænsede.

Bevis. Per antagelse findes en mulig vektor, \mathbf{y} , til dualen. Ifølge Sætning 4.2.2 gælder $\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}$ for alle mulige \mathbf{x} . Derfor kan det ikke lade sig gøre, at $\mathbf{c}^T \mathbf{x}$ får en højere værdi end $\mathbf{b}^T \mathbf{y}$. Det modsatte gælder tilsvarende for minimumsværdien. \square

Sagt på anden vis: Hvis f.eks. maksimeringsproblemet er ubegrænset, og dermed ikke har en optimal værdi, vil dualen være inkonsistent. Dette skyldes, at $\mathbf{c}^T \mathbf{x}$ vil kunne antage vilkårligt høje værdier, og dermed ville der ikke findes et \mathbf{y} , som kan opfylde $\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}$ for alle \mathbf{x} .

Korollar 4.2.4.

Lad \mathbf{x}_0 og \mathbf{y}_0 være løsninger til hhv. primalproblemet og dualen, og antag, at $\mathbf{c}^T \mathbf{x}_0 = \mathbf{b}^T \mathbf{y}_0$. Så er \mathbf{x}_0 og \mathbf{y}_0 optimale løsninger for deres respektive problemer.

Bevis. Hvis \mathbf{x} er en løsning til primalproblemet, så gælder: $\mathbf{c}^T \mathbf{x} \leq \mathbf{b}^T \mathbf{y}_0$. Idet maksimeringsproblemet er begrænset opadtil af $\mathbf{b}^T \mathbf{y}_0$, må $\mathbf{c}^T \mathbf{x} = \mathbf{b}^T \mathbf{y}_0$ medføre, at \mathbf{x} er optimal. Hvis \mathbf{y} er en løsning til dualen, så gælder: $\mathbf{b}^T \mathbf{y} \geq \mathbf{c}^T \mathbf{x}_0$. Da problemet er begrænset nedadtil af $\mathbf{c}^T \mathbf{x}_0$, så er $\mathbf{c}^T \mathbf{x}_0 = \mathbf{b}^T \mathbf{y}$ problemets minimum, hvorved \mathbf{y} er optimal. \square

Sætning 4.2.5. Dualitetssætning

Hvis et standard lineært programmeringsproblem er konsistent og begrænset, så er dualen det også. Der eksisterer optimale vektorer til hver af de respektive problemer, og problemernes optimale værdier er ens.

Bevis. Der er tre muligheder for et lineært programmeringsproblem: Det kan være konsistent og begrænset(kb), konsistent og ubegrænset(kub) eller ikke konsistent(ik). For primalproblemet og dualen er der altså ni måder at kombinere de tre tidligere nævnte muligheder. Af disse ni kombinationer er der kun fire, der kan forekomme. Disse fremgår som de farvede søjler i tabellen nedenfor.

Varianter	1	2	3	4	5	6	7	8	9
Primal	kb	kb	kb	kub	kub	kub	ik	ik	ik
Dual	kb	kub	ik	kb	kub	ik	kb	kub	ik

Tabel 4.1: Kombinationer mellem løsningerne til det primale og duale problem

Ifølge Korollar 4.2.3 er varianterne 2, 4 og 5 ikke mulige, for hvis både primalproblemet og dualen er konsistente, så skulle begge også være begrænsede. Derudover siger den første del af Dualitetssætningen 4.2.5, at hvis et problem er konsistent og begrænset(kb), kan dualen ikke være inkonsistent(ik). Dermed er der to andre varianter, der ikke kan opstå, nemlig varianterne 3 og 7. De resterende fire muligheder, 1, 6, 8 og 9, er således de

eneste, der kan forekomme. Med dette er den første del af sætningen bevist (at hvis et standard lineært programmeringsproblem er konsistent og begrænset, så er dualen det også). Se Afsnit 4.6.11 for anden del af beviset (at der eksisterer optimale vektorer til hver af de respektive problemer, og at problemernes optimale værdier er ens). \square

Sætning 4.2.6. *Ligevægtssætning*

Lad \mathbf{x} og \mathbf{y} være mulige vektorer for hhv. et maksimeringsproblem og dets duale. Så er \mathbf{x} og \mathbf{y} optimale, hvis og kun hvis:

$$y_i = 0, \text{ når } \sum_{j=1}^n a_{i,j}x_j < b_i$$

og

$$x_j = 0, \text{ når } \sum_{i=1}^m a_{i,j}y_i > c_j.$$

Bevis. Først vises, at $y_i = 0$, medmindre der er lighed i $\sum_{j=1}^n a_{i,j}x_j \leq b_i$. Dermed fås:

$$\sum_{i=1}^m y_i b_i = \sum_{i=1}^m y_i \sum_{j=1}^n a_{i,j}x_j = \sum_{i=1}^m \sum_{j=1}^n a_{i,j}x_j y_i.$$

På samme måde viser den anden antagelse, at:

$$\sum_{i=1}^m \sum_{j=1}^n a_{i,j}x_j y_i = \sum_{j=1}^n c_j x_j.$$

Da haves der fra Korollar 4.2.4, at \mathbf{x} og \mathbf{y} er optimale.

Det var første del af biimplikationen i ligevægtssætningen. Anden del starter med Sætning 4.2.2 om svag dualitet, hvorfra der fås:

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m \sum_{j=1}^n a_{i,j}x_j y_i \leq \sum_{i=1}^m y_i b_i.$$

Ifølge Dualitetssætningen, Sætning 4.2.5, er venstre side lig højre, når \mathbf{x} og \mathbf{y} er optimale. Ligheden kan skrives om til:

$$\sum_{j=1}^n (c_j - \sum_{i=1}^m y_i a_{i,j})x_j = 0.$$

Siden der vides, at både \mathbf{x} og \mathbf{y} er mulige vektorer, er hvert led i summen ikke-negativt, og summen kan kun blive 0, hvis hvert led er 0. Derfor gælder, at hvis $\sum_{i=1}^m y_i a_{i,j} > c_j$, så er $x_j = 0$.

Tilsvarende fremgangsmetode viser, at hvis $\sum_{j=1}^n a_{i,j}x_j < b_i$, så er $y_i = 0$. \square

4.3 Geometri

Afsnittet behandler geometrien i lineær optimering og fungerer dermed som grundlag for at forstå simplex-metoden. Først introduceres *polytop*, *hyperplan* og *halvrum*:

Definition 4.3.1. *Polytop*

En *polytop* er en mængde, som kan skrives på formen:

$$\{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} \geq \mathbf{b}\},$$

hvor A er en $m \times n$ matrix, og \mathbf{b} er en vektor i \mathbb{R}^m .

Det gælder for alle lineære optimeringsproblemer, at mulighedsområdet kan beskrives ved betingelser på formen, $A\mathbf{x} \geq \mathbf{b}$. Mulighedsområdet er derfor en polytop. I Afsnit 3.1 var der et eksempel på et mulighedsområde, som udgjorde en femkant i \mathbb{R}^2 .

Definition 4.3.2. *Begrænset mængde*

En mængde, $S \subset \mathbb{R}^n$, er *begrænset*, hvis der findes en konstant, k , så den absolutte værdi af enhver komponent af ethvert element i S er mindre end eller lig k .

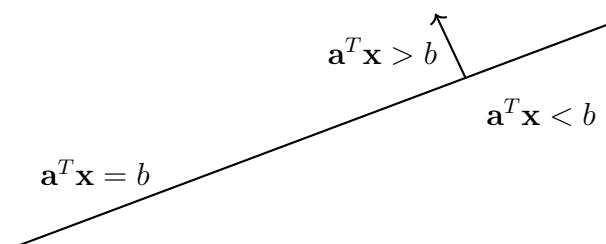
For polytope, der er bestemt ved en enkelt lineær betingelse, gælder følgende definition:

Definition 4.3.3. *Hyperplan og halvrum*

Lad \mathbf{a} være en ikke-nulvektor i \mathbb{R}^n og b en skalar.

- (a) Mængden $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T \mathbf{x} = b\}$ kaldes en *hyperplan*.
- (b) Mængden $\{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{a}^T \mathbf{x} \geq b\}$ kaldes et *halvrum*.

En polytop er derfor lig fællesmængden for et endeligt antal halvrum. Hyperplanen er randen for det tilsvarende halvrum, og i illustrationen nedenfor fremstår hyperplanen derfor som en linje, medens halvrummet er ovenfor og under hyperplanen:



Sætning 4.3.4.

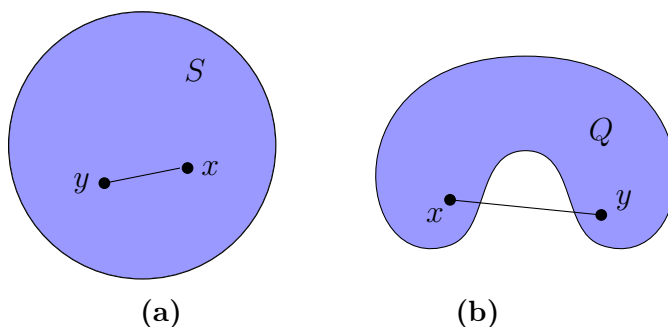
Vektoren, \mathbf{a} , i definitionen af hyperplanen er ortogonal med hyperplanen.

Bevis. Lad \mathbf{x} og \mathbf{y} tilhøre samme hyperplan, så er $\mathbf{a}^T \mathbf{x} = \mathbf{a}^T \mathbf{y}$, og dermed $\mathbf{a}^T (\mathbf{x} - \mathbf{y}) = \mathbf{0}$, så \mathbf{a} er ortogonal på enhver vektor i hyperplanen. \square

4.3.1 Konvekse mængder**Definition 4.3.5. Konveks mængde**

En mængde, $S \subset \mathbb{R}^n$, er konveks, hvis der for alle $\mathbf{x}, \mathbf{y} \in S$, og ethvert $\lambda \in [0, 1]$ gælder: $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in S$.

Det vil sige, at i en konveks mængde gælder det, at det linjestykke, som forbinder vilkårlige vektorer, \mathbf{x} og \mathbf{y} , tilhører S . I figur 4.1 ses to mængder, S og Q . Mængden, S , er konveks, da linjestykket mellem to vilkårlige vektorer, $\mathbf{x}, \mathbf{y} \in S$, er indeholdt i S . Det samme er ikke tilfældet ved mængden Q , hvorfor denne ikke er konveks.



Figur 4.1: Illustration af en konveks mængde, S , og en ikke-konveks mængde, Q .

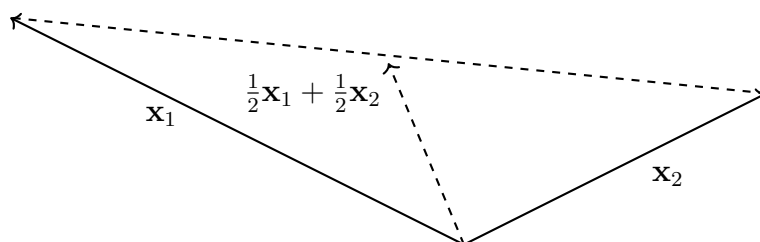
Nu defineres det *konvekse hylster* af en mængde af vektorer:

Definition 4.3.6. Det konvekse hylster

Lad $\mathbf{x}_1, \dots, \mathbf{x}_k$ være vektorer i \mathbb{R}^n og $\lambda_1, \dots, \lambda_k$ være ikke-negative skalarer, hvor $\sum_{i=1}^k \lambda_i = 1$. Da gælder:

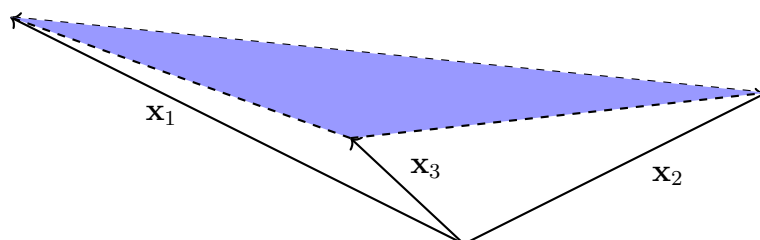
- (a) Vektoren, $\sum_{i=1}^k \lambda_i \mathbf{x}_i$, siges at være en konveks kombination af $\mathbf{x}_1, \dots, \mathbf{x}_k$.
- (b) Det konvekse hylster af vektorerne, $\mathbf{x}_1, \dots, \mathbf{x}_k$, er mængden af alle konvekse kombinationer af disse vektorer.

Hvis der blot betragtes to vektorer i \mathbb{R}^n , \mathbf{x}_1 og \mathbf{x}_2 , er det konvekse hylster af disse et linjestykke:



Vist ovenfor er to vilkårlige vektorer, \mathbf{x}_1 og \mathbf{x}_2 , samt en konveks kombination, hvor $\lambda_1 = \lambda_2 = \frac{1}{2}$. Alle de konvekse kombinationer af \mathbf{x}_1 og \mathbf{x}_2 udspænder tilsammen linjestykket mellem de to vektorer.

Hvis der i stedet betragtes tre lineært uafhængige vektorer i \mathbb{R}^n , bliver mængden af konvekse kombinationer et trekantet udsnit af en todimensionel flade:



Af ovenstående definitioner følger følgende resultater:

Sætning 4.3.7.

- (a) *Fællesmængden af konvekse mængder er konveks.*
- (b) *Polytoper er konvekse mængder.*
- (c) *En konveks kombination af et endeligt antal elementer af en konveks mængde tilhører mængden.*
- (d) *Det konvekse hylster af et endeligt antal vektorer er en konveks mængde.*

Bevis. (a) Lad S_i , $i \in I$, være en konveks mængde, hvor I er en indeksmængde, og antag at \mathbf{x} og \mathbf{y} tilhører fællesmængden $\bigcup_{i \in I} S_i$. Lad $\lambda \in [0, 1]$. Siden hver S_i er konveks og indeholder \mathbf{x} og \mathbf{y} , gælder det, at $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in S_i$, hvilket beviser, at $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}$ også tilhører fællesmængden af mængderne S_i . Derfor er $\bigcup_{i \in I} S_i$ konveks.

- (b) Lad \mathbf{a} være en vektor, og lad b være en skalar. Antag at \mathbf{x} og \mathbf{y} opfylder $\mathbf{a}^T \mathbf{x} \leq b$ og $\mathbf{a}^T \mathbf{y} \leq b$, og derfor tilhører samme halvrum. Lad $\lambda \in [0, 1]$. Så er $\mathbf{a}^T (\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda b + (1 - \lambda) b = b$, hvilket beviser, at $\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}$ også tilhører det samme halvrum. Altså er et halvrum konvekst. Da en polytop er fællesmængden af et endeligt antal halvrum, følger resultatet af (a).

- (c) En konveks kombination af to elementer af en konveks mængde ligger i mængden ud fra definitionen af konveksitet. Antag som induktionshypotese, at en konveks kombination af k elementer af en konveks mængde tilhører mængden. Betragt $k + 1$ elementer, $\mathbf{x}_1, \dots, \mathbf{x}_{k+1}$, af en konveks mængde, S , og lad $\lambda_1, \dots, \lambda_{k+1}$ være ikke-negative skalarer med sum 1. Det antages, at $\lambda_{k+1} \neq 1$. Så gælder der:

$$\sum_{i=1}^{k+1} \lambda_i \mathbf{x}_i = \lambda_{k+1} \mathbf{x}_{k+1} + (1 - \lambda_{k+1}) \sum_{i=1}^k \frac{\lambda_i}{1 - \lambda_{k+1}} \mathbf{x}_i. \quad (4.1)$$

Koefficienterne, $\lambda_i / (1 - \lambda_{k+1})$, $i = 1, \dots, k$, er ikke-negative og summerer til 1. Ved brug af induktionshypotesen fås $\sum_{i=1}^k \lambda_i \mathbf{x}_i / (1 - \lambda_{k+1}) \in S$. Det, at S er konveks og (4.1) medfører, at $\sum_{i=1}^{k+1} \lambda_i \mathbf{x}_i \in S$, og induktionstrinnet er færdigt.

- (d) Lad S være det konvekse hylster af vektorerne, $\mathbf{x}_1, \dots, \mathbf{x}_k$, og lad $\mathbf{y} = \sum_{i=1}^k \zeta_i \mathbf{x}_i$, og $\mathbf{z} = \sum_{i=1}^k \Theta_i \mathbf{x}_i$ være to elementer i S , hvor $\zeta_i \geq 0$, $\Theta_i \geq 0$, og $\sum_{i=1}^k \zeta_i = \sum_{i=1}^k \Theta_i = 1$. Lad $\lambda \in [0, 1]$. Så gælder:

$$\lambda \mathbf{y} + (1 - \lambda) \mathbf{z} = \lambda \sum_{i=1}^k \zeta_i \mathbf{x}_i + (1 - \lambda) \sum_{i=1}^k \Theta_i \mathbf{x}_i = \sum_{i=1}^k (\lambda \zeta_i + (1 - \lambda) \Theta_i) \mathbf{x}_i.$$

Bemærk, at koefficienterne, $\lambda \zeta_i + (1 - \lambda) \Theta_i$, $i = 1, \dots, k$, er ikke-negative og har sum 1. Det viser, at $\lambda \mathbf{y} + (1 - \lambda) \mathbf{z}$ er en konveks kombination af $\mathbf{x}_1, \dots, \mathbf{x}_k$ og dermed tilhører S . Altså er S konveks.

□

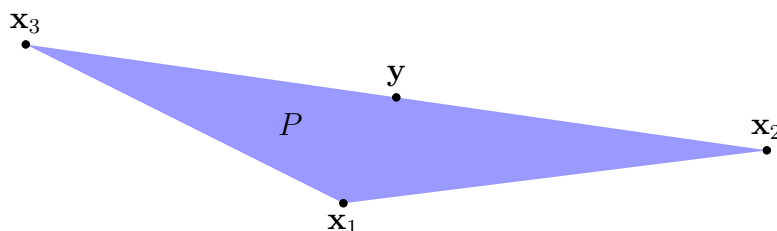
4.3.2 Ekstremumpunkter

Geometrisk set findes løsningen til et optimeringsproblem i et hjørne eller, hvis der er mange optimale løsninger, langs en af kanterne på polytopet. Derfor er det væsentligt at beskæftige sig med polytopets *ekstremumpunkter*.

Definition 4.3.8. *Ekstremumpunkt*

Lad P være en polytop. En vektor, $\mathbf{x} \in P$, er et ekstremumpunkt i P , hvis der ikke findes to vektorer, $\mathbf{y}, \mathbf{z} \in P$, forskellige fra \mathbf{x} , og en skalar, $\lambda \in [0, 1]$, så $\mathbf{x} = \lambda \mathbf{y} + (1 - \lambda) \mathbf{z}$.

Øverst på næste side ses en figur, hvor \mathbf{x}_1 , \mathbf{x}_2 og \mathbf{x}_3 udgør ekstremumpunkter, hvorimod \mathbf{y} ikke er et ekstremumpunkt, da $\mathbf{y} = \frac{1}{2} \mathbf{x}_2 + (1 - \frac{1}{2}) \mathbf{x}_3$.

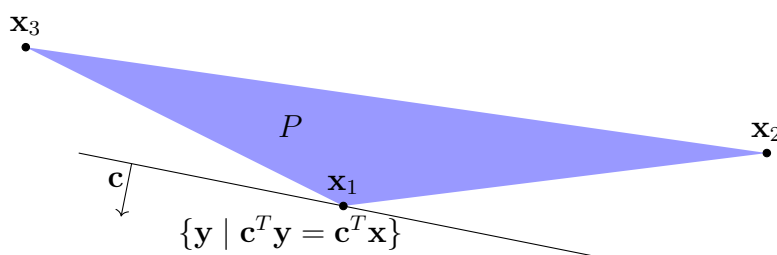


Alternativt kan et hjørnepunkt i en polytop, P , defineres som den entydige optimale løsning på et lineært optimeringsproblem, hvor mulighedsområdet er givet ved P :

Definition 4.3.9. Hjørnepunkt

Lad P være en polytop. En vektor, $\mathbf{x} \in P$, er et hjørnepunkt i P , hvis der eksisterer et \mathbf{c} , så $\mathbf{c}^T \mathbf{x} > \mathbf{c}^T \mathbf{y}$ for alle $\mathbf{y} \in P$, $\mathbf{y} \neq \mathbf{x}$.

Det vil sige, at \mathbf{x} er et hjørnepunkt i P , hvis og kun hvis P er på den ene side af en hyperplan, $\{\mathbf{y} \mid \mathbf{c}^T \mathbf{y} = \mathbf{c}^T \mathbf{x}\}$, som kun møder P i punktet \mathbf{x} .



Hyperplanen rører P i et enkelt punkt, og \mathbf{x}_1 er dermed et hjørnepunkt.

4.3.3 Algebraisk definition

De geometriske definitioner på ekstremumpunkter og hjørnepunkter er ikke altid praktiske at arbejde med. Derfor indføres en algebraisk definition baseret på repræsentationen af polytopet i form af lineære betingelser. Først indføres yderligere terminologi:

Betragt en polytop, $P \subset \mathbb{R}^n$, defineret via betingelserne:

$$\mathbf{a}_i^T \mathbf{x} \geq b_i, \quad i \in M_1$$

$$\mathbf{a}_i^T \mathbf{x} \leq b_i, \quad i \in M_2$$

$$\mathbf{a}_i^T \mathbf{x} = b_i, \quad i \in M_3$$

Hvor M_1 , M_2 og M_3 er endelige indeks-mængder, hvert \mathbf{a}_i er en vektor i \mathbb{R}^n , og hvert b_i er en skalar.

Definition 4.3.10. Aktiv/bindende betingelse

Hvis en vektor, \mathbf{x}^* , opfylder $\mathbf{a}_i^T \mathbf{x}^* = b_i$ for et i i M_1 , M_2 eller M_3 , siges den pågældende betingelse at være aktiv eller bindende i \mathbf{x}^* .

Hvis der er n betingelser, som er aktive ved en vektor, \mathbf{x}^* , så opfylder \mathbf{x}^* et system på n lineære ligninger med n ubekendte. Systemet har en entydig løsning, hvis og kun hvis vektorerne, \mathbf{a}_i , svarende til de n ligninger, er lineært uafhængige. I denne forbindelse kan det også siges, at ligningerne er lineært uafhængige. Hermed menes, at vektorerne, svarende til ligningerne, er lineært uafhængige.

Sætning 4.3.11.

Lad \mathbf{x}^* være et element i \mathbb{R}^n , og lad $I = \{i \mid \mathbf{a}_i^T \mathbf{x}^* = b_i\}$ være mængden af indeks af betingelser, som er aktive ved \mathbf{x}^* . Så er følgende udsagn ækvivalente:

- (a) Der findes n vektorer i mængden $\{\mathbf{a}_i^T \mid i \in I\}$, som er lineært uafhængige.
- (b) Spannet af vektorerne \mathbf{a}_i , $i \in I$, er hele \mathbb{R}^n . Dvs. at ethvert element i \mathbb{R}^n kan udtrykkes som en linearkombination af vektorerne \mathbf{a}_i , $i \in I$.
- (c) Systemet af ligningerne $\mathbf{a}_i^T \mathbf{x} = b_i$, $i \in I$, har en entydig løsning.

Nu kan der laves en algebraisk definition på hjørnepunkter, nemlig som en mulig løsning, hvor der er n lineært uafhængige aktive betingelser. Da interessen kun er i løsninger indenfor mulighedsområdet, skal alle bibetingelser opfyldes, hvilket medfører, at lighedsbetingelser skal være aktive.

Der kan ledes efter hjørnepunkter på følgende måde: Lad lighedsbetingelserne være aktive og lad så tilstrækkeligt mange yderligere betingelser være aktive til, at der er samlet n lineært uafhængige aktive betingelser. Så kan en entydig vektor, \mathbf{x}^* , bestemmes. Der er dog ingen garanti for at finde en mulig vektor, \mathbf{x}^* , hvilket er tilfældet, hvis en eller flere af de inaktive betingelser ikke er overholdt. I så fald siges løsningen at være en *basal*, men ikke en *mulig*, løsning.

Definition 4.3.12. Basal/basal mulig løsning

Lad P være en polytop defineret ved lineære betingelser (ligheder og uligheder), og lad \mathbf{x}^* være et element i \mathbb{R}^n .

- (a) Vektoren \mathbf{x}^* er en basal løsning, hvis:
 - (1) Alle lighedsbetingelser er aktive.
 - (2) Af de aktive betingelser i \mathbf{x}^* er n af dem lineært uafhængige.
- (b) Hvis \mathbf{x}^* er en basal løsning, som opfylder alle bibetingelser og ikke-negativitetsbetingelser, siges \mathbf{x}^* at være en basal mulig løsning.

Sætning 4.3.13.

De tre forskellige definitioner i form af ekstremumpunkter, hjørnepunkter og basale mulige løsninger er ækvivalente.

Bevis. Beviset deles op i tre udsagn, A, B og C, og dermed også tre dele, $A \Rightarrow B$, $B \Rightarrow C$, og $C \Rightarrow A$.

Det antages, at P er repræsenteret ved betingelser på formen $\mathbf{a}_i^T \mathbf{x} \leq b_i$, og $\mathbf{a}_i^T \mathbf{x} = b_i$.

Hjørnepunkt \Rightarrow Ekstremumpunkt. (A \Rightarrow B)

Antag, at $\mathbf{x}^* \in P$ er et hjørnepunkt. Så findes der et $\mathbf{c} \in \mathbb{R}^n$, så $\mathbf{c}^T \mathbf{x}^* > \mathbf{c}^T \mathbf{y}$, for ethvert $\mathbf{y} \in P$, $\mathbf{y} \neq \mathbf{x}^*$. Hvis $\mathbf{y} \in P$, $\mathbf{z} \in P$, $\mathbf{y} \neq \mathbf{x}^*$, $\mathbf{z} \neq \mathbf{x}^*$, og $0 \leq \lambda \leq 1$, $\mathbf{c}^T \mathbf{x}^* > \mathbf{c}^T \mathbf{y}$, og $\mathbf{c}^T \mathbf{x}^* > \mathbf{c}^T \mathbf{z}$, forudsætter det, at

$$\mathbf{c}^T \mathbf{x}^* > \mathbf{c}^T (\lambda \mathbf{y} + (1 - \lambda) \mathbf{z}),$$

og derfor $\mathbf{x}^* \neq \lambda \mathbf{y} + (1 - \lambda) \mathbf{z}$. Så \mathbf{x}^* kan ikke udtrykkes som en konveks kombination af to andre elementer i P , og \mathbf{x}^* er dermed et ekstremumpunkt, jævnfør Definition 4.3.8.

Dvs. hvis en vektor er et hjørnepunkt, er vektoren et ekstremumpunkt.

Ekstremumpunkt \Rightarrow Basal mulig løsning. (B \Rightarrow C)

Antag, at $\mathbf{x}^* \in P$ ikke er en basal mulig løsning. Det skal da vises, at \mathbf{x}^* ikke er et ekstremumpunkt.

Lad $I = \{i \mid \mathbf{a}_i^T \mathbf{x}^* = b_i\}$. Da \mathbf{x}^* ikke er en basal mulig løsning, findes der ikke n lineært uafhængige vektorer i mængden, \mathbf{a}_i , $i \in I$. Dvs. at vektorerne, \mathbf{a}_i , $i \in I$, ligger i et ikke-trivielt underrum af \mathbb{R}^n , og der findes en ikke-nulvektor, $\mathbf{d} \in \mathbb{R}^n$, sådan at $\mathbf{a}_i^T \mathbf{d} = 0$, for alle $i \in I$. Lad ϵ være et lille positivt tal. Betragt vektorerne, $\mathbf{y} = \mathbf{x}^* + \epsilon \mathbf{d}$ og $\mathbf{z} = \mathbf{x}^* - \epsilon \mathbf{d}$. Bemærk, at $\mathbf{a}_i^T \mathbf{y} = \mathbf{a}_i^T \mathbf{x}^* = b_i$ for $i \in I$.

Yderligere gælder det for $i \notin I$, at $\mathbf{a}_i^T \mathbf{x}^* < b_i$, og hvis ϵ er lille, gælder der også, $\mathbf{a}_i^T \mathbf{y} < b_i$. (Det er tilstrækkeligt at vælge ϵ , så $\epsilon |\mathbf{a}_i^T \mathbf{d}| < b_i - \mathbf{a}_i^T \mathbf{x}^*$ for alle $i \notin I$). Når ϵ er tilstrækkeligt lille, gælder der, $\mathbf{y} \in P$, og tilsvarende $\mathbf{z} \in P$. Det kan da ses, at $\mathbf{x}^* = (\mathbf{y} + \mathbf{z})/2$, hvilket vil sige, at \mathbf{x}^* ikke er et ekstremumpunkt.

Dvs. hvis en vektor er et ekstremumpunkt, er vektoren en basal mulig løsning.

Basal mulig løsning \Rightarrow Hjørnepunkt. (C \Rightarrow A)

Lad \mathbf{x}^* være en basal mulig løsning, og lad $I = \{i \mid \mathbf{a}_i^T \mathbf{x}^* = b_i\}$. Lad $\mathbf{c} = \sum_{i \in I} \mathbf{a}_i$. Så haves:

$$\mathbf{c}^T \mathbf{x}^* = \sum_{i \in I} \mathbf{a}_i^T \mathbf{x}^* = \sum_{i \in I} b_i. \quad (4.2)$$

Yderligere gælder det for ethvert $\mathbf{x} \in P$ og ethvert i , at $\mathbf{a}_i^T \mathbf{x} \leq b_i$ og

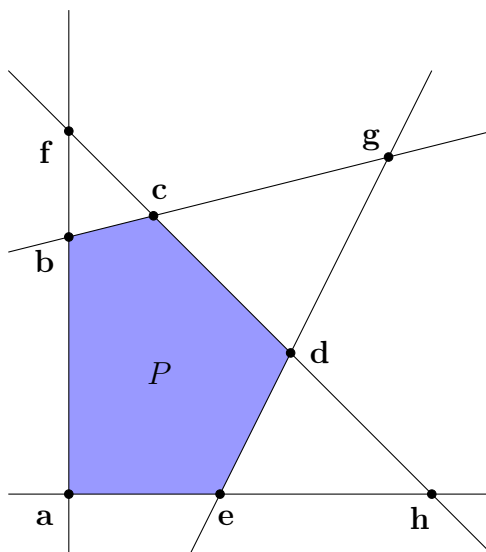
$$\mathbf{c}^T \mathbf{x} = \sum_{i \in I} \mathbf{a}_i^T \mathbf{x} \leq \sum_{i \in I} b_i. \quad (4.3)$$

Det viser, at \mathbf{x}^* er en optimal løsning på problemet med at maksimere $\mathbf{c}^T \mathbf{x}$ over mulighedsområdet P . Yderligere gælder ovenstående lighed i (4.3), hvis og kun hvis $\mathbf{a}_i^T \mathbf{x} = b_i$ for alle $i \in I$. Da \mathbf{x}^* er en basal mulig løsning, findes der n lineært uafhængige betingelser, som er aktive ved \mathbf{x}^* , og \mathbf{x}^* er den entydige løsning til systemet af ligninger, $\mathbf{a}_i^T \mathbf{x} = b_i$, $i \in I$, jævnfør Sætning 4.3.11. Det følger heraf, at \mathbf{x}^* er den vektor som entydigt maksimerer $\mathbf{c}^T \mathbf{x}$ over mængden P , og derfor er \mathbf{x}^* et hjørnepunkt i P .

Hvis en vektor er en basal mulig løsning, er vektoren altså et hjørnepunkt. \square

En vektor er en basal mulig løsning, hvis og kun hvis, den er et ekstremumpunkt. Da definitionen af et ekstremumpunkt ikke refererer til en specifik fremstilling af polytopet, er det, at være en basal mulig løsning, også uafhængigt af fremstillingen.

Betragt Figur 4.2, hvor der ses polytopet, P , som er afgrænset af fem betingelser. Punkterne, **a**, **b**, **c**, **d** og **e**, angiver basale mulige løsninger, mens **f**, **g** og **h** er basale løsninger, og ligger udenfor mulighedsområdet.



Figur 4.2: En afgrænset polytop samt tilhørende basale og basale mulige løsninger

Korollar 4.3.14.

Givet et endeligt antal lineære ulighedsbetingelser, kan der kun være et endeligt antal basale eller basale mulige løsninger.

Bevis. Betragt et system på m lineære ulighedsbetingelser, som begrænser en vektor, $\mathbf{x} \in \mathbb{R}^n$. Ved en hvilken som helst basal løsning, er der n lineært uafhængige aktive betingelser. Da enhver mængde på n lineært uafhængige aktive betingelser definerer et entydigt punkt, følger det, at forskellige basale løsninger svarer til forskellige mængder på n lineært uafhængige aktive betingelser. Derfor er antallet af basale løsninger begrænset opadtil af det antal måder, hvorpå der kan vælges n ud af m betingelser, hvilket er endeligt. \square

Løsningsmetoden for et lineært optimeringsproblem er normaltvis at gå igennem de basale mulige løsninger. Antallet af basale mulige løsninger kan dog være højt, så at finde funktionsværdien for samtlige basale mulige løsninger kan være en upraktisk strategi til at løse et optimeringsproblem. Derfor kan der være brug for en algoritme/metode, som kan finde den optimale løsning uden at gennemgå samtlige mulige basale løsninger (se simplex-algoritmen i Afsnit 4.6).

Definition 4.3.15. Naboer

To forskellige basale løsninger til en mængde af lineære betingelser i \mathbb{R}^n er naboer, hvis der kan findes $n - 1$ lineært uafhængige betingelser, som er aktive ved dem begge. Hvis sådanne to naboer er mulige, er linjen mellem dem en rand på det mulige område.

På Figur 4.2 ovenfor kan det ses, at **d** har **c**, **e**, **g** og **h** som naboer. Linjestykkerne fra **c** til **d** og fra **d** til **e** udgør hver en rand på polytopet.

4.4 Polytoper på standardform

Den tidligere definition, Definition 4.3.12, på basale løsninger er anvendelig på polytoper på generel form. Denne definition kan gøres mere præcis, når der arbejdes med polytoper på standardform, hvilket fører frem til resultater, der er vigtige indenfor simpleksmetoden.

Definition 4.4.1. Polytop på standardform

En polytop, P , siges at være på standardform, hvis den er skrevet på formen:

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

hvor A er en $m \times n$ matrix, og $\mathbf{b} \in \mathbb{R}^m$.

Når der arbejdes med problemer på standardform antages der, at alle m lighedsbetingelser er lineært uafhængige. Intuitivt kan det forklares ved, at enhver betingelse der ikke er lineært uafhængig er triviel, hvilket kan gøres tydeligt efter anvendelse af rækkeoperationer.

Sætning 4.4.2.

Lad $P = \{\mathbf{x} \in \mathbb{R}^n \mid A\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, hvor A er en $m \times n$ matrix, $P \neq \emptyset$. Lad rækkerne, $\mathbf{r}_{A,i_1}, \dots, \mathbf{r}_{A,i_k}$, være lineært uafhængige, hvor $i_1, \dots, i_k \in \mathbb{N}$, der henviser til hvert sit rækkeindeks, og lad $\text{rang}(A) = k < m$. Betragt polytopet:

$$Q = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{r}_{A,i_1}\mathbf{x} = b_{i_1}, \dots, \mathbf{r}_{A,i_k}\mathbf{x} = b_{i_k}, \mathbf{x} \geq \mathbf{0}\}.$$

Så gælder der $Q = P$.

Bevis. Der arbejdes med det tilfælde, hvor de første k rækker i A er lineært uafhængige, så $i_1 = 1, i_2 = 2, \dots, i_k = k$. Hvis dette ikke er opfyldt, anvendes rækkeombytning indtil det gælder, hvilket gør, at følgende er gældende generelt.

For at bevise $Q = P$, vises at $P \subseteq Q$, og $Q \subseteq P$. Det, at P er en delmængde af Q , er intuitivt, da enhver løsning, der opfylder betingelserne i P , også vil opfylde betingelser i Q . I det følgende vises derfor, at $Q \subseteq P$:

Da $\text{rang}(A) = k$, har rækkerummet af A dimension k , og de k lineært uafhængige rækker er en basis for rækkerummet. Dermed kan rækkerne i A beskrives ved, $\mathbf{r}_{A,i} = \sum_{j=1}^k c_{i,j} \mathbf{r}_{A,j}$, hvor $c_{i,j}$ er skalarer. Lad $\mathbf{x} \in P$, da opfylder b_i :

$$b_i = \mathbf{r}_{A,i} \mathbf{x} = \sum_{j=1}^k c_{i,j} \mathbf{r}_{A,j} \mathbf{x} = \sum_{j=1}^k c_{i,j} b_j.$$

Lad $\mathbf{y} \in Q$:

$$\mathbf{r}_{A,i} \mathbf{y} = \sum_{j=1}^k c_{i,j} \mathbf{r}_{A,j} \mathbf{y} = \sum_{j=1}^k c_{i,j} b_j = b_i.$$

Dermed er $\mathbf{y} \in P$ og $Q \subseteq P$ er bevist. \square

Betragt som eksempel følgende betingelser, som definerer en polytop, P :

$$\begin{aligned} 2x_1 + 2x_2 &= 2 \\ 2x_1 + 2x_3 &= 2 \\ 4x_1 + 2x_2 + 2x_3 &= 4 \\ x_1, x_2, x_3 &\geq 0 \end{aligned}$$

De to første betingelser er lineært uafhængige. Den sidste række kan udtrykkes som summen af de to første rækker. Altså efter to rækkeoperationer, $r_3 \rightarrow r_3 - r_1$ og $r_3 \rightarrow r_3 - r_2$, er sidste række en nulrække og dermed også trivial.

Nu hvor begrundelsen bag antagelsen, $\text{rang}(A) = m$, er redegjort for, arbejdes der videre indenfor basale løsninger for problemer på standardform.

Definitionen på en basal løsning kræver, at n lineært uafhængige betingelser er aktive. Under antagelsen, $\text{rang}(A) = m$, er der allerede m aktive lineært uafhængige betingelser. Dette medfører, at $n - m$ ikke-negativitetsbetingelser skal være aktive, hvilket opnås ved at vælge $n - m$ variable og sætte dem til 0. Disse variable kan dog ikke vælges vilkårligt, hvilket beskrives i nedenstående sætning.

Sætning 4.4.3.

Givet en polytop på standardform og antagelsen, at $\text{rang}(A) = m$, er vektoren $\mathbf{x} \in \mathbb{R}^n$ en basal løsning, hvis og kun hvis $A\mathbf{x} = \mathbf{b}$, og der eksisterer en indeksliste, $i_1, \dots, i_m \in \mathbb{N}$, som opfylder følgende:

- (a) Søjlerne, $\mathbf{s}_{A,i_1}, \dots, \mathbf{s}_{A,i_m}$, er lineært uafhængige.
- (b) $x_j = 0$, når $j \neq i_1, \dots, i_m$.

Bevis. Der bevises først, at betingelserne (a) og (b) medfører, at \mathbf{x} er en basal løsning. Lad $\mathbf{x} \in \mathbb{R}^n$ og antag, at i_1, \dots, i_m opfylder betingelserne, (a) og (b). Da $x_j = 0$,

$$\sum_{y=1}^m \mathbf{s}_{A,i_y} x_{i_y} = \sum_{y=1}^n \mathbf{s}_{A,y} x_y = A\mathbf{x} = \mathbf{b}.$$

Det lineære ligningssystem dannet ud af de lineært uafhængige søjler, $\mathbf{s}_{A,i_1}, \dots, \mathbf{s}_{A,i_m}$, har en entydig løsning, som ifølge Sætning 4.3.11 er ækvivalent med, at der er m lineært uafhængige betingelser, som i dette tilfælde også er aktive, hvilket medfører, at \mathbf{x} er en basal løsning.

Det bevises nu, at hvis \mathbf{x} er en basal løsning, medfører det, at betingelserne, (a) og (b), er sande. Lad \mathbf{x} være en basal løsning og x_{i_1}, \dots, x_{i_k} være indgangene, der er forskellige fra 0. Da \mathbf{x} er en basal løsning, har følgende en entydig løsning, $\sum_{y=1}^n \mathbf{s}_{A,y} x_y = \mathbf{b}$, $x_j = 0$, når $j \neq i_1, \dots, i_k$ ifølge Sætning 4.3.11. Derfor har $\sum_{y=1}^k \mathbf{s}_{A,i_y} x_{i_y} = \mathbf{b}$ en entydig løsning. Dette betyder, at søjlerne, $\mathbf{s}_{A,i_1}, \dots, \mathbf{s}_{A,i_k}$, er lineært uafhængige, da det ellers vil modstride løsningens entydighed. Det, at søjlerne er lineært uafhængige, medfører, at $k \leq m$, da der ikke kan være flere lineært uafhængige søjler end bibetingelser. Antagelsen, at $\text{rang}(A) = m$, medfører, at der er m lineært uafhængige søjler, der udspænder \mathbb{R}^m . Det er givet i Sætning 3.5.13, (b), at det er muligt at finde $m - k$ yderligere søjler, så der i alt er m lineært uafhængige søjler, og dermed er betingelse, (a), i Sætning 4.4.3 opfyldt. Hvis $j \neq i_1, \dots, i_m$, så er $j \neq i_1, \dots, i_k$, og det er blevet antaget, at indgangene, x_{i_1}, \dots, x_{i_k} , er de eneste forskellige fra nul. Derfor gælder, at $x_j = 0$, og betingelse, (b), er dermed opfyldt. \square

Resultaterne fra Sætning 4.4.3 giver en god metode til at bestemme basale løsninger til et optimeringsproblem på standardform. Metoden er givet ved følgende tre trin:

- 1 Vælg m lineært uafhængige søjler, $\mathbf{s}_{A,i_1}, \dots, \mathbf{s}_{A,i_m}$.
- 2 Lad $x_j = 0$ for alle $j \neq i_1, \dots, i_m$.
- 3 Løs for variable, x_{i_1}, \dots, x_{i_m} , i bibetingelserne, $A\mathbf{x} = \mathbf{b}$.

Dette giver ikke nødvendigvis en basal mulig løsning, men da alle basale mulige løsninger også er basale løsninger, kan enhver basal mulig løsning bestemmes vha. denne metode. Ved anvendelse af metoden er \mathbf{x} en basal løsning. Variablerne, x_{i_1}, \dots, x_{i_m} , kaldes de *basale variable* og som svarer til de såkaldte slack-variable, og variablerne, x_j med $j \neq i_1, \dots, i_m$ som bliver sat til nul, kaldes *ikke-basale variable*. Søjlerne i A , der svarer til koefficienterne for de basale variable, kaldes *basale søjler*, og sammen danner det en basis.

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 0 & 0 & 0 \\ 0 & 1 & 6 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} = \begin{bmatrix} 8 \\ 12 \\ 4 \\ 6 \end{bmatrix}.$$

Søjlerne, $\mathbf{s}_{A,4}, \mathbf{s}_{A,5}, \mathbf{s}_{A,6}, \mathbf{s}_{A,7}$, er lineært uafhængige og vælges derfor som basale søjler, der sammen danner en basis. De ikke-basale variable sættes til 0, og der løses for de 4 basale

variable, hvilket danner følgende basale løsning, $\mathbf{x}^T = [0, 0, 0, 8, 12, 4, 6]$. Alle variable opfylder ikke-negativitetsbetingelserne, og \mathbf{x} er derfor en mulig basal løsning. Hvis en anden basis vælges, for eksempel $\{\mathbf{s}_{A,3}, \mathbf{s}_{A,5}, \mathbf{s}_{A,6}, \mathbf{s}_{A,7}\}$, dannes der en ikke-mulig basal løsning, $\mathbf{x}^T = [0, 0, 4, 0, -12, 4, 6]$.

Baserne i eksemplet ovenfor bestemte hver deres entydige basale løsning. Dette er ikke altid tilfældet, da der kan være baser, som deler samme basale løsning. Eksempelvis, hvis $\mathbf{b} = \mathbf{o}$, så vil enhver basis have den samme basale løsning. Det, at flere baser har samme basale løsning, skaber problemer for algoritmer, der bestemmer en optimal løsning.

4.5 Degenerering

Ifølge den tidligere definition på en basal løsning, skal der være n lineært uafhængige aktive betingelser. Dette medfører, at antallet af aktive betingelser kan være større end n , i hvilket tilfælde det siges, at løsningen er *degenereret*. Det, at en løsning er degenereret, har stor indflydelse på en lineær programmeringsalgoritmes funktion, eksempelvis simpleks-metoden.

Definition 4.5.1. Degenereret basal løsning

En basal løsning, $\mathbf{x} \in \mathbb{R}^n$, kaldes degenereret, hvis flere end n betingelser er aktive.

Geometrisk vil degenerering opstå i et skæringspunkt med mindst $n + 1$ hyperplaner i n dimensioner.

Dette er den generelle definition, men da simpleks-metoden arbejder med en polytop på standardform, kan en mere specifik definition udledes. I enhver løsning for polytopet på standardform skal der per definition være m aktive betingelser. Så for at en basal løsning kan være degenereret, skal flere end $n - m$ af ikke-negativitetsbetingelserne være aktive, hvilket vil sige, at der er flere end $n - m$ variable med værdi 0.

Definition 4.5.2. Degenereret basal løsning for polytop på standardform

I en polytop på standardform, P , så er en basal løsning, \mathbf{x} , degenereret, hvis flere end $n - m$ af elementerne i \mathbf{x} er 0.

Betragt følgende polytop på standardform:

$$P = \{\mathbf{x} \in \mathbb{R}^3 \mid 3x_1 - 2x_2 = 0, 3x_1 + 2x_2 + 2x_3 = 12, x_1, x_2, x_3 \geq 0\}$$

Her er $n = 3$, $m = 2$, og $n - m = 1$. Løsningen, $\mathbf{x} = [2, 3, 0]^T$, er ikke-degenereret, da kun én af variablene er 0, og der dermed kun er 3 aktive betingelser, som er de to lighedsbetingelser ($m = 2$) og en af ikke-negativitetsbetingelserne, $x_3 \geq 0$. Løsningen, $\mathbf{x} = [0, 0, 6]^T$, er derimod degenereret, da to variable har værdi 0, hvilket medfører, at der er 4 aktive betingelser, der består af to lighedsbetingelser og to ikke-negativitetsbetingelser: $x_1 \geq 0$ og $x_2 \geq 0$.

4.6 Simplex-algoritmen

Som det blev set i den første del af projektet, er matricer og vektorer nogle udmærkede redskaber til at løse hurtigt og uden besvær systemer af lineære ligninger. Projektets opgave beskæftiger sig med at løse et system af flere ligheder/uligheder, hvor en udvidet matrix, som det kaldes simplex-tabel, bliver brugt i afsnittet nedenfor. Afsnittet er udarbejdet ved brug af Hurlbert [2009].

4.6.1 Simplex-tabellen

Med udgangspunkt i det primale problem med indførelsen af $v = 0$, find \mathbf{x} for at:

$$\begin{aligned} \text{Maksimér } z &= v + \mathbf{c}^T \mathbf{x} \\ \text{betinget af } A\mathbf{x} &\leq \mathbf{b} \\ \mathbf{x} &\geq \mathbf{o}. \end{aligned}$$

For at se algebraisk på vores lineære optimeringsproblem med m uligheder og n ubekendte, indføres m slack-variable, sådan at

$\mathbf{x}_{ny} = [x_1, \dots, x_j, \dots, x_n, x_{n+1}, \dots, x_{n+i}, \dots, x_{n+m}]^T$, $A_{ny} = [A \quad I_m]$, og $\mathbf{c}_{ny}^T = [\mathbf{c}^T \quad \mathbf{o}^T]$, hvor nulvektoren har m indgange. Dermed bliver ulighederne lavet til ligheder, og det ordnede problem kan nu skrives på standardform:

$$\begin{aligned} \text{Maksimér } z - \mathbf{c}_{ny}^T \mathbf{x}_{ny} &= v \\ \text{betinget af } A_{ny}\mathbf{x}_{ny} &= \mathbf{b} \\ \mathbf{x}_{ny} &\geq \mathbf{o} \end{aligned}$$

Der er tidligere angivet i Afsnit 4.4, i hvilket tilfælde en variabel er basal eller ikke-basal. Nu indføres *ordningen*, som er et ligningsystem skrevet sådan at:

- Variablerne og objektvariablen, z , skrives på den venstre side af lighedstegnet af ligningen.
- På højre side skrives b -værdierne.

Simplex-metodens første skridt består i at indsætte den omarrangerede ordning af det lineære optimeringsproblem i en udvidet matrix kaldet en *simplex-tabel*. Som det ses i tabellen nedenfor, skrives objektfunktionen i den nederste række, også kaldet *objektrækken*, og slack-variablerne, i form af identitetsmatricen, augmenteres på matricen, A , efterfulgt af z og \mathbf{b} søjlen.

Her opstilles en simplex-tabel lige efter indførelsen af slack-variable kaldet den *initierende* eller *oprindelige tabel*:

$a_{1,1}$	$a_{1,2}$	\cdots	$a_{1,n}$	1	\cdots	0	0	b_1
$a_{2,1}$	$a_{2,2}$	\cdots	$a_{2,n}$	0	1	\cdots	0	b_2
\cdot	\cdot		\cdot	\cdot	\cdot		\cdot	\cdot
\cdot	\cdot		\cdot	\cdot	\cdot		\cdot	\cdot
$a_{m,1}$	$a_{m,2}$	\cdots	$a_{m,n}$	0	0	\cdots	1	b_m
$-c_1$	$-c_2$	\cdots	$-c_n$	0	0	\cdots	0	v

I den oprindelige tabel svarer de ikke-basale variable til de oprindelige variable. Mere generelt, efter der er blevet lavet rækkeoperationer (dvs $a_{i,j}$ er ikke nødvendigvis lig $a'_{i,j}$), kan simplex-tabellen skrives: :

$a'_{1,1}$	$a'_{1,2}$	\cdots	$a'_{1,n}$	$a'_{1,n+1}$	$a'_{1,n+2}$	\cdots	$a'_{1,n+m}$	0	b'_1
$a'_{2,1}$	$a'_{2,2}$	\cdots	$a'_{2,n}$	$a'_{2,n+1}$	$a'_{2,n+2}$	\cdots	$a'_{2,n+m}$	0	b'_2
\cdot	\cdot		\cdot	\cdot	\cdot		\cdot	\cdot	\cdot
\cdot	\cdot		\cdot	\cdot	\cdot		\cdot	\cdot	\cdot
$a'_{m,1}$	$a'_{m,2}$	\cdots	$a'_{m,n}$	$a'_{m,n+1}$	$a'_{m,n+2}$	\cdots	$a'_{m,n+m}$	0	b'_m
$-c'_1$	$-c'_2$	\cdots	$-c'_n$	$-c'_{n+1}$	$-c'_{n+2}$	\cdots	$-c'_{n+m}$	1	v'

En ordning eller tabel kaldes *mulig*, hvis den tilhørende basale løsning er en mulig løsning. Omvendt kaldes en ordning eller tabel *ikke-mulig*, hvis den basale løsning ikke er en mulig løsning.

Når tabellen er ikke-mulig, siges der, at algoritmen er i *Fase 1*. Hvis tabellen er mulig, er den i *Fase 2*. Derudover kaldes tabellen *optimal*, hvis den tilsvarende basale løsning er optimal.

4.6.2 Pivoting

At *pivotere* betyder at erstatte en ikke-basal variabel med en basal variabel. Den indgang, der skal pivoteres i, kaldes *pivoteringsindgangen*. Søjlen og rækken, som pivoteringsindgangen tilhører, kaldes henholdsvis *pivoteringssøjlen* og *pivoteringsrækken*. Den variabel, der er tilknyttet pivoteringssøjlen, er en ikke-basal variabel. Hvis x_j er en ikke-basal variabel og x_u en basal variabel på samme pivoteringsrække, da bliver en pivotering betegnet, $j \rightarrow u$. I denne sammenhæng kaldes x_j den *indgående* variabel, da den ved en pivotering bringes ind i basen og bliver en basal variabel. Tilsvarende kaldes x_u for den *udgående* variabel, da denne forlader basen og bliver til en ikke-basal variabel.

I et ligningssystem svarer en pivotering, $j \rightarrow u$, til at isolere variabelen, x_j , i én ligning, og substituere udtrykket for den ind i de øvrige ligninger. I simplex-tabellen svarer den samme pivotering til, at der laves rækkeoperationer, så pivoteringsindgangen selv bliver et positivt helt tal, og alle andre indgange i pivoteringssøjlen bliver 0. Dette må kun gøres, hvis pivoteringsindgangen ikke er lig 0.

Nedenfor forklares simplex-algoritmens pivoteringsstrategi, det vil sige, hvordan en pivoteringsindgang vælges.

Pivoteringsstrategi for en mulig simplex-tabel (tabel i Fase 2):

1. Vælg en negativ indgang i objektrækken (se bort fra de to værdier længst til højre). Denne indgang bestemmer pivoteringssøjlen.
Hvis der er mere end én negativ indgang i objektrækken, er der flere muligheder for valg af pivoteringssøjlen. I så fald kan én af nedenstående fremgangsmåder følges:
 - Vælg søjlen længst til venstre, dvs. den med det mindste indeks. Metoden kaldes også *Mindste Indeks* (MI).
 - Vælg søjlen med den mest negative c -koefficient også kaldet *Mest Negativ* metoden (MN).
 - Beregn, hvilket pivot fører til største z -værdi og vælg den.

Hvilken af de tre metoder, som benyttes, har selvfølgelig indflydelse på antallet af pivoteringer, før en optimal tabel bestemmes. Der er ingen af de tre metoder, som altid er hurtigst, heller ikke gennemsnitligt. Metoden, som anvendes i dette projekt, er (MI), da denne har den fordel, at der altid kan garanteres, at algoritmen terminerer, hvilket der gøres rede for i et senere afsnit. Andre pivoteringsmetoder kan foretrækkes i praksis, da de kan have lavere kompleksitet, hvilket kort behandles i Afsnit 4.6.9.

Efter valget af pivoteringssøjlen er foretaget, skal pivoteringsindgangen vælges. Fremgangsmåden for dette forklares i det følgende.

2. I den udvalgte pivoteringssøjle vælges en indgang, $a_{i,j} > 0$, hvor b -ratioen er minimal. b -ratioen er givet ved $b_i/a_{i,j}$ og dens minimum kaldes den *mindste ikke-negative ratio*. At vælge den mindste b -ratio sikrer, at algoritmen ikke forlader mulighedsområdet. Dermed sikres det, at den basale løsning, som svarer til tabellen efter pivoteringen, fortsat er mulig. Hvis alle $a_{i,j} \leq 0$, da er problemet ubegrænset. Dette uddybes i Afsnit 4.6.5.

I tilfælde af lighed mellem to eller flere mindste b -ratioer vælges indgangen i rækken, hvori pivot-indgangen er i søjlen med laveste indeks.

Derudover skal det bemærkes, at når $a_{i,j} < 0$, bliver b -ratio en nedre grænse for x_j 's værdi. Hvis $a_{i,j} = 0$ findes variabelen x_j ikke i begrænsningerne svarende til den pågældende pivoteringsrække. Derfor ses der bort fra disse tilfælde.

At der vælges en negativ indgang i objektrækken skyldes, at objektfunktionsværdien, z , øges, når en ikke-basal variabel, $x_j = 0$, øges. Det er samtidigt nødvendigt, at $\mathbf{x} \geq \mathbf{0}$, hvorfor rækken vælges ud fra, hvor b -ratioen er mindst. Dette sikrer nemlig, at man får pivoteret den basale variabel ud, som udgør den mest restriktive begrænsning. Dermed overholdes betingelserne samtidig med at værdien af en ikke-basal variabel forøges, således den basale løsning forbliver mulig. Processen kan derfor gentages, indtil der ingen negativ indgang er i objektrækken, hvormed algoritmen slutter, og en optimal tabel og

henholdsvis en optimal løsning er bestemt. Der er dog tilfælde, hvor algoritmen ikke slutter efter endeligt mange iterationer. Dette fænomen kaldes cykling og forklares i Afsnit 4.6.7.

Betragt følgende ordning (bemærk, at denne ikke nødvendigvis er den første ordning af systemet):

$$\begin{array}{ll} \text{Maksimér} & z + x_2 - x_3 = 85 \\ \text{betinget af} & x_1 + x_3 = 5 \\ & x_4 + x_2 = 7 \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

I ordningen ovenfor er x_2 og x_3 de ikke-basale variable, og x_1 og x_4 er de basale variable. Ud fra basen i ordningen bestemmes en basal mulig løsning, $\mathbf{x} = [5, 0, 0, 7]^T$, med en objektfunktions værdi på 85. Problemet er derfor i Fase 2. Da koefficienten for variabelen x_3 er positiv, kan objektfunktionsværdien øges ved at pivotere sådan, at x_3 er den indgående variabel. Da x_3 kun er udtrykt i første bibetingelse, er x_1 den udgående variabel, altså foretages pivoteringen $3 \rightarrow 1$. Dette fører frem til følgende ordning:

$$\begin{array}{ll} \text{Maksimér} & z + x_1 + x_2 = 90 \\ \text{betinget af} & x_3 + x_1 = 5 \\ & x_4 + x_2 = 7 \\ & \mathbf{x} \geq \mathbf{0} \end{array}$$

Som kan ses, er det nu x_1 og x_2 , der er de ikke-basale variable, og x_3 og x_4 de basale variable. Ordningen er mulig, da den basale løsning er $\mathbf{x} = [0 \ 0 \ 5 \ 7]^T$, hvilket er en ikke-negativ løsning og dermed mulig. Løsningen giver en objektfunktionsværdi på 90. Simplex-tabellen angives derved som:

$$S = \begin{array}{cc|ccc|c} 1 & 0 & 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 1 & 0 & 7 \\ \hline 1 & 1 & 0 & 0 & 1 & 90 \end{array}$$

Bemærk, at der ingen negative indgange er i objektrækken, hvilket stopper pivoteringsprocessen. Dermed er tabellen optimal.

Problemet, der blev angivet i eksemplet ovenfor, var allerede i Fase 2, hvor pivoteringsstrategien for en mulig tabel skal følges. For at løse problemer, som begynder i Fase 1, kan enten den såkaldte *auxiliære metode*, som vil blive forklaret i det kommende afsnit, eller *shortcut-metoden*, der vil blive beskrevet i Afsnit 4.6.4, anvendes. Begge metoder kan få problemet til at komme i Fase 2.

4.6.3 Auxiliære problemer

Hvis tabellen for et optimeringsproblem på standardform ikke er mulig, befinder algoritmen sig i Fase 1. Derfor skal den auxiliære metode bruges til at gøre tabellen mulig, således algoritmen kan påbegynde Fase 2. Det primale problem omskrives ved at indføre en hjælpe-variabel, x_0 . Således har det primale problem, P :

$$\begin{aligned} \text{Maksimer } z &= \sum_{j=1}^n c_j x_j \\ \text{betinget af } \sum_{j=1}^n a_{i,j} x_j &\leq b_i \quad (1 \leq i \leq m) \\ x_j &\geq 0 \quad (1 \leq j \leq n) \end{aligned}$$

et auxiliært problem, Q :

$$\begin{aligned} \text{Maksimer } v &= -x_0 \\ \text{betinget af } -x_0 + \sum_{j=1}^n a_{i,j} x_j &\leq b_i \quad (1 \leq i \leq m) \\ x_j &\geq 0 \quad (0 \leq j \leq n), \end{aligned}$$

som tages i brug, når den basale løsning i problemets oprindelige tabel ikke ligger inden for mulighedsområdet. Simplex-tabellen tilhørende det auxiliære problem kaldes den *auxiliære tabel*.

Den første pivotering, som foretages i den auxiliære tabel, består i at pivotere x_0 ind i basen. Den udgående variabel vælges som den basale variabel tilhørende rækken med den mest negative b -værdi. I tilfælde af lighed mellem b -værdier, vælges den basale variabel med mindst indeks. Efter denne indledende pivotering er foretaget, er den auxiliære tabel mulig. Hvis der findes en negativ indgang i den auxiliære tabels objektrække, på nær i de to sidste søjler, kan Fase 2-pivoteringsstrategien følges. Eftersom $x_j \geq 0 \quad (0 \leq j \leq n)$ skal være opfyldt, er Q 's optimale løsning $v^* = 0$. Dernæst kan der vendes tilbage til det oprindelige problem, P , og benytte den mulige basis, som blev fundet vha. den auxiliære metode.

Sætning 4.6.1.

Lad P være et et maksimeringsproblem, (hvor den tilhørende basale løsning ikke ligger inden for mulighedsområdet). Lad Q være P 's auxiliære problem. P er konsistent, hvis og kun hvis Q er optimal ved $x_0 = 0$.

Bevis. Baseret på [<https://math.stackexchange.com/users/3736/will-orrick>].

Lad $(x_0, x_1, \dots, x_n) = (0, a_1, \dots, a_n)$. Antag, at $(0, a_1, \dots, a_n)$ er en mulig løsning til det auxiliære problem. Hvis denne løsning er sat ind i de auxiliære betingelser, så er alle led,

hvor x_0 indgår, lig 0. Det ses så, at (a_1, \dots, a_n) opfylder betingelserne til det originale problem, og det er derfor en mulig løsning hertil.

Antag nu, at (a_1, \dots, a_n) er en mulig løsning til det originale problem, hvilket betyder, at den opfylder alle betingelserne hertil. Ved at sætte $(0, a_1, \dots, a_n)$ ind i betingelserne for det auxiliære problem, falder x_0 -leddene fra, så betingelserne til det originale problem står tilbage. Siden det blev antaget, at (a_1, \dots, a_n) var en mulig løsning til P , er betingelserne for Q opfyldt.

Hvis den optimale løsning til Q giver den optimale værdi $v^* = 0$, så udgør løsningen til det auxiliære problem en mulig basal løsning til det oprindelige problem. Hvis den optimale løsning til Q giver en optimal værdi, $v^* < 0$, er det ikke muligt at sætte $x_0 = 0$, og så har det oprindelige problem ikke nogen mulig løsning. Dette betyder, at det oprindelige problem, P , er inkonsistent. \square

For at give et eksempel, kan det auxiliære problem bruges til et maksimeringsproblem:

$$\begin{aligned} \text{Maksimér } z &= 3x_1 + 2x_2 \\ \text{betinget af } 2x_1 + x_2 &\leq -12 \\ -x_1 - 2x_2 &\leq -2 \\ x_1 - 3x_2 &\leq -1 \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

som har den initierende tabel:

$$\left[\begin{array}{cc|cccc|c} 2 & 1 & 1 & 0 & 0 & 0 & -12 \\ -1 & -2 & 0 & 1 & 0 & 0 & -2 \\ 1 & -3 & 0 & 0 & 1 & 0 & -1 \\ \hline -3 & -2 & 0 & 0 & 0 & 1 & 0 \end{array} \right] \quad (4.4)$$

Den basale løsning, $\mathbf{x} = [0, 0, -12, -2, -1]^T$, er ikke mulig. Derfor tages den auxiliære metode i brug. For dette eksempel er det:

$$\begin{aligned} \text{Maksimér } v &= -x_0 \\ \text{betinget af } -x_0 + 2x_1 + x_2 &\leq -12 \\ -x_0 - x_1 - 2x_2 &\leq -2 \\ -x_0 + x_1 - 3x_2 &\leq -1 \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

som har simplex tabellen:

$$\left[\begin{array}{c|ccc|cccc|c} -1 & 2 & 1 & 1 & 0 & 0 & 0 & -12 \\ -1 & -1 & -2 & 0 & 1 & 0 & 0 & -2 \\ -1 & 1 & -3 & 0 & 0 & 1 & 0 & -1 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right]$$

Den første pivotering, som laves i tabellen ovenfor, har til formål at bringe problemet ind i det mulige område. Dette gøres ved at pivotere x_0 ind i basen ved at erstatte den med den mest negative basale variabel, dvs. den hvis b -koordinat er mest negativ. I dette eksempel laves en pivotering, $0 \rightarrow 3$, da -12 er den mest negative b -værdi her. Tabellen skrives nu:

$$\left[\begin{array}{ccc|cccc|c} 1 & -2 & -1 & -1 & 0 & 0 & 0 & 12 \\ 0 & -3 & -3 & -1 & 1 & 0 & 0 & 10 \\ 0 & -1 & -4 & -1 & 0 & 1 & 0 & 11 \\ \hline 0 & 2 & 1 & 1 & 0 & 0 & 1 & -12 \end{array} \right]$$

Den basale løsning, $\mathbf{x} = [12, 0, 0, 0, 10, 11]$, opfylder ikke-negativitetsbetingelserne, og er derfor en mulig løsning til det auxiliære problem. Nu følges pivoteringsstrategien nævnt tidligere. Der er ingen negativ indgang i objektrækken undtagen sidste søjle, og tabellen er derfor optimal. Der kan dog ikke arbejdes videre med den originale tabel i Fase 2, da $v^* \neq 0$, og det oprindelige problem, P , derfor er inkonsistent. Det kan selvfølgelig også ses på betingelsen, $2x_1 + x_2 \leq -12$, som aldrig kan blive opfyldt.

4.6.4 Shortcut-metoden

Shortcut-metoden er endnu en pivoteringsstrategi for Fase 1, som kan benyttes for at spare tid og undgå besvær. Metoden har samme formål som den auxiliære metode, nemlig at producere en mulig simplex-tabel. Shortcut-metoden anvendes på den oprindelige simplex-tabel, hvor pivoteringsrækken vælges som rækken med den mest negative b -værdi. I tilfælde af to eller flere identiske b -værdier vælges rækken, hvori pivot-indgangen er i søjlen med laveste indeks. Derefter bestemmes pivoteringssøjlen som den søjle med en negativ koefficient i og laveste indeks. Denne proces gentages indtil der ingen negative b -værdier er tilbage og der fortsættes derefter med Fase 2. For at illustrere forskellen på auxiliær og shortcut-metoden, vil et eksempel på en ikke mulig tabel først gøres mulig vha. auxiliær metode og dernæst shortcut-metoden. Derefter vil der blive redegjort for, hvorfor den auxiliære metode kan reduceres til shortcut-metoden.

Betragt følgende maksimeringsproblem opstillet som en simplex-tabel:

$$\left[\begin{array}{ccc|ccc|c} -3 & -9 & 0 & 1 & 0 & 0 & -1 \\ -5 & -5 & -1 & 0 & 1 & 0 & -1 \\ \hline 90 & 180 & 15 & 0 & 0 & 1 & 0 \end{array} \right]$$

Ved at benytte den auxiliære metode laves pivoteringerne, $0 \rightarrow 4$ og $1 \rightarrow 0$, som danner en optimal tabel, hvor $v^* = 0$:

$$\left[\begin{array}{ccc|ccc|c} 1 & 3 & 9 & 0 & -1 & 0 & 0 & 1 \\ 2 & 0 & 30 & -3 & -5 & 3 & 0 & 2 \\ \hline 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{array} \right]$$

Det ses, at $\{x_1, x_5\}$ udgør en mulig basis for det oprindelige maksimeringsproblem, hvor der først blev forsøgt med $\{x_4, x_5\}$. Denne information bruges nu på den oprindelige tabel for at gøre den mulig. Pivoteringen, $1 \rightarrow 4$, foretages for at bringe x_1 ind i basen, medens x_5 allerede er i basen. Ved at indsætte udtrykket for x_1 i objektfunktionen, $z = -90x_1 - 180x_2 - 15x_3$, ændres dette til $z = -30 + 90x_2 - 15x_3 - 30x_4$. Dette giver en mulig tabel, som er klar til simplex-algoritmens Fase 2. Tabellen er vist nedenfor.

$$\left[\begin{array}{ccc|ccc|c} 3 & 9 & 0 & -1 & 0 & 0 & 1 \\ 0 & 30 & -3 & -5 & 3 & 0 & 2 \\ \hline 0 & -90 & 15 & 30 & 0 & 1 & -30 \end{array} \right]$$

Bemærk, at der er anvendt to pivoteringer, $0 \rightarrow 4$ og $1 \rightarrow 0$, for at opnå en mulig basis vha. auxiliær metode. Ved brug af shortcut-metoden på den initierende tabel, er det kun nødvendigt at lave én pivotering, før tabellen er i Fase 2. Pivoteringsindgangen i den initierende tabel findes ved først at finde pivoteringsrækken. b -værdierne er i dette tilfælde identiske, hvorfor række 1 vælges. Der er flere negative koefficienter i række 1, hvorfor søjlen med det laveste indeks vælges. Pivoteringssøjlen er derfor søjle 1. Pivoteringen $1 \rightarrow 4$ foretages, hvilket giver tabellen:

$$\left[\begin{array}{ccc|ccc|c} 3 & 9 & 0 & -1 & 0 & 0 & 1 \\ 0 & 30 & -3 & -5 & 3 & 0 & 2 \\ \hline 0 & -90 & 15 & 30 & 0 & 1 & -30 \end{array} \right]$$

Tabellen er lig med den tabel, som blev udledt vha. den auxiliære metode, men kun én pivotering er anvendt. Inklusionen af x_0 og den længere auxiliær metode var derfor ikke nødvendig i dette eksempel. Faktisk er det aldrig nødvendigt at bruge auxiliære metode, da shortcut-metoden altid kan erstatte den auxiliære metode med samme resultat.

Sætning 4.6.2.

Shortcut-metoden og den auxiliære metode bestemmer samme indgående og udgående variable, hvis der ses bort fra x_0 , og metoderne kommer derfor frem til samme simplex-tabel.

Bevis. Baseret på *The Equivalence of the Auxiliary and Shortcut Methods for the Simplex Algorithm* [2008].

Udgangspunktet for beviset er, at både den auxiliære metode og shortcut-metoden betragtes som hver sin forenklede udgave af en *udvidet metode*. Den udvidede metode svarer til den auxiliære metode, hvor hver pivotering, $l \rightarrow k$, dog udvides til to pivoteringer, idet en omvej over x_0 tages, i form af pivoteringerne, $0 \rightarrow k$ og $l \rightarrow 0$.

De nu tre pivoteringsstrategier sammenlignes for at vise, at de resulterer i samme netto-pivotering, altså at de resulterer i samme tabel. Betragt en simplex-tabel for et vilkårligt optimeringsproblem i Fase 1. Der er indført slack-variable, der er indført en x_0 -søjle, og

der er to objektrækker - én svarende til det oprindelige problem, P , og én svarende til det auxiliære problem, Q . Hvis tabellen betragtes ud fra den udvidede metode eller den auxiliære metode, skal P -objektrækken ignoreres, da det ikke er det problem, som arbejdes på. Hvis tabellen betragtes ud fra shortcut-metoden, kan både Q -objektrækken og x_0 -søjlen ignoreres, da x_0 -variablen slet ikke indføres her. Nu forestilles der sig en række på hinanden følgende udgaver af denne tabel; T_1 , hvor x_0 ikke er i basis, T_2 , hvor en pivotering, $0 \rightarrow k$, i T_1 er foretaget, og T_3 , som er fremkommet ved at foretage en pivotering, $l \rightarrow 0$, i T_2 , se Figur 4.3. Rækken af tabeller kan fortsætte, sådan at alle T_{ulige} har x_0 som ikke-basal variabel, mens alle T_{lige} har x_0 i basen. Den udvidede metode arbejder sig igennem samtlige af tabellerne i denne række, da den netop skiftevis pivoterer x_0 ind og ud af basen. Den auxiliære metode begynder i en ulige tabel og pivoterer x_0 ind i basen for at nå Q 's mulighedsområde. Herefter springer den over de ulige tabeller (da x_0 ikke forlader basen), indtil den når en optimal tabel. Hvis den optimale tabel har $v^* = 0$, er det en ulige tabel, da x_0 har forladt basen igen. Shortcut-metoden bevæger sig kun igennem tabellerne med ulige indeks, da x_0 slet ikke indgår.

Figur 4.3: Oversigt over de tre Fase 1-strategier

-	Udvidet metode	Auxiliær metode	Shortcut-metode
T_1 x_0 ikke-basal	Række: mest negative b -værdi Søjle: x_0	(Dette trin findes kun, hvis T_1 er den oprindelige tabel. I så fald, er reglen ligesom for den udvidede metode)	Række: mest negative b -værdi Søjle: første negative $a_{r,j}$ -værdi
T_2 x_0 er basal	Søjle: første negative c -værdi Række: x_0	Søjle: første negative c -værdi Række: Laveste ikke-negative b -ratio	Dette trin findes ikke
T_3 x_0 ikke-basal	Række: mest negative b -værdi Søjle: x_0	(Dette trin findes ikke) Hvis/når algoritmen fjerner x_0 fra basen, er tabellen optimal, og algoritmen går til Fase 2	Række: mest negative b -værdi Søjle: første negative $a_{r,j}$ -værdi

Hvis shortcut-metoden og den udvidede metode sammenlignes, kan det ses, at den udvidede metode bruger to pivoteringer på at foretage den netto-pivotering, som shortcut-metoden gennemfører med én pivotering. Shortcut-metoden vælger først en pivoteringsrække (mest negative b -værdi), og dermed den variabel, som skal forlade basen. Derefter vælges en pivoteringssøjle, svarende til den først negative koefficient i pivoteringsrækken, og dermed bestemmes den variabel, som skal flyttes ind i basen. Den udvidede metode bruger først samme regel til at vælge den udgående variabel, og pivoterer x_0 ind, derefter pivoteres x_0 ud igen, hvor den indgående variabel vælges ud fra den første negative værdi i objektrækken. Dette svarer fuldstændig til shortcut-metodens valg af første negative koefficient i pivoteringsrækken, da koefficienterne i den udvidede metodes objektrække i en lige tabel har samme værdi som koefficienterne i den foregående ulige tabels pivoteringsrække. Hvis x_0 -søjlen og Q -objektrækken ignoreres, er de to metoders T_{ulige} -tabeller ens.

Hvis den auxiliære metode og den udvidede metode sammenlignes, ses det, at den udvidede metode, efter den første pivotering, udfører samme netto-pivotering i to trin, som den auxiliære metode udfører i én pivotering. Pivoteringssøjlen, og dermed den indgående variabel, vælges i de to metoder ud fra samme regel om første (laveste indeks) negative c -værdi; den udvidede metode vælger automatisk x_0 som udgående variabel, hvor den auxiliære metode vælger række ud fra den laveste ikke-negative b -ratio. Når den udvidede metode herefter pivoterer x_0 ud igen, vælges rækken ud fra den mest negative b -værdi. Det springende punkt i forhold til at vise, at den auxiliære metode og shortcut-metoden når frem til samme resultat, består altså i at vise, at de to regler for at vælge udgående variabel er ækvivalente. Det skal vises, at variabelen, bestemt ud fra den mest negative b -værdi i en ulige tabel, er den samme som variabelen, valgt ud fra laveste ikke-negative b -ratio i en lige tabel.

Det antages, at der på et tidspunkt under anvendelsen af den auxiliære metode opnås en mulig tabel, samt at den udvidede metode er kommet frem til den samme tabel. Et udsnit af en sådan tabel ses nedenfor (inklusive den variabel, som er på vej ind i basen, og **b**-søjlen):

$$\begin{array}{ccc|c} a & . & & b \\ c & & . & d \\ e & & . & f \end{array}$$

Det vides, at $b, d, f \geq 0$, da tabellen er mulig. Antag, at den auxiliære metode vælger at pivotere i a , selvom c og e også er mulige pivoteringsindgange. Det vil sige, at $a, c, e > 0$, og $0 \leq \frac{b}{a} \leq \frac{d}{c}$.

Udsnittet af den tilsvarende tabel for den udvidede metode er identisk med udsnittet fra den auxiliære metode. Det antages, at den nederste række indeholder den basale variabel, x_0 . Da x_0 er basal, var $-f$ den mest negative indgang i sin søjle i den forrige tabel i den udvidede metode, så det gælder også, at $f > 0$. Den udvidede metode vælger altså at pivotere i indgang, e (samme søjle som den auxiliære metode, men x_0 -rækken vælges til at forlade basen), hvilket resulterer i følgende tabel:

$$\begin{array}{ccc|c} 0 & . & & be-af \\ 0 & & . & de-cf \\ e & & . & f \end{array}$$

Antag nu, at $de - cf < 0$. Så vil den udvidede metode nu vælge at pivotere i denne række, medmindre $be - af$ er mere negativ. Det ville være i strid med den auxiliære metode, som, det blev antaget, pivoterer i den øverste række. Men $de - cf < 0$ medfører, at $\frac{d}{c} < \frac{f}{e}$, og $\frac{b}{a} < \frac{d}{c}$ indebærer, at $b < \frac{ad}{c}$. Dermed haves $b - d < \frac{d}{c}(a - c) < \frac{f}{e}(a - c)$, hvilket vil sige, at $e(b - d) < f(a - c)$, eller $be - af < de - cf$. Altså pivoterer den udvidede metode i første række ligesom den auxiliære metode.

Det gælder også den anden vej: for $b, d \geq 0$ og $a, c, e, f > 0$, og $\frac{d}{c} < \frac{f}{e}$, gælder der, $be - af < de - cf$, hvis og kun hvis $\frac{b}{a} < \frac{d}{c}$.

Det vil sige, at den auxiliære metode og shortcut-metoden vælger samme variabel til at forlade basen. Den indgående variabel er den samme for dem, så de to metoder foretager samme pivoteringssekvens. \square

4.6.5 Ubegrænsethed

Hvis optimeringsproblemet er ubegrænset, kan simplex-algoritmen selvfølgelig ikke bestemme en optimal løsning. Antag, at der haves en simplex-tabel i Fase 2. Hvis objektrækken ikke har nogen negative indgange, er tabellen optimal. Ellers vælges den søjle, hvis indgang i objektrækken er negativ og har lavest indeks, som pivoteringssøjle. For at vælge pivoteringsrække, skal en af de positive indgange i pivoteringssøjlen så vælges (den med lavest b -ratio). Men hvis pivoteringssøjlen ikke indeholder nogen positive indgange, vil det sige, at der ikke er nogen begrænsning på den tilsvarende variabel (det vides, at alle b -værdier er positive, da algoritmen er i Fase 2), og optimeringsproblemet er ubegrænset. Hvis pivoteringssøjlen ikke har en positiv indgang, afbryder algoritmen dermed.

4.6.6 Pseudokode

For at skabe overblik over algoritmen, opstilles her en pseudokode for simplex-algoritmen med udgangspunkt i Mindste Indeks-metoden.

Simplex-algoritmen

Input: Simplex-tabel for maksimeringsproblem på standardform, med slack-variable indført.

Fase 1:

- (1) Undersøg mulighed: Hvis indgangene i tabellens sidste søjle (b -værdierne) alle er ikke-negative, er tabellen mulig - gå da til Fase 2.
- (2) Vælg pivoteringsrække: Rækken med mest negative b -værdi vælges. Ved lighed mellem to eller flere b -værdier, vælges rækken, hvori pivot-indgangen er i søjlen med laveste indeks.
- (3) Vælg pivoteringsindgang: I pivoteringsrækken vælges den første (laveste indeks) indgang med negativ værdi.
Hvis ingen indgang i pivoteringsrækken er negativ, afbryder algoritmen og returnerer "Problem inkonsistent".
- (4) Udfør pivot-operationer: Der pivoteres i pivoteringsindgangen og returneres dernæst til punkt (1).

Fase 2:

- (5) Undersøg optimalitet: Hvis alle indgangene i objektrækken er ikke-negative, med undtagelse af de to sidste, er tabellen optimal. Algoritmen afbryder og returnerer løsningen \mathbf{x} og funktionsværdien z .
Ellers:

- (6) Vælg pivoteringssøjle: I objektrækken vælges den første (laveste indeks) indgang med negativ værdi. Den tilsvarende søjle er pivoteringssøjlen.
- (7) Vælg pivoteringsindgang: I søjlen vælges indgangen med laveste ikke-negative b -ratio. Ved lighed vælges indgangen i rækken, hvori pivot-indgangen er i søjlen med laveste indeks.
Hvis alle b -ratioer er negative, afbryder algoritmen og returnerer "Problem ubegrænset".
- (8) Udfør rækkeoperationer: Der pivoteres i pivoteringsindgangen og returneres til punkt (5).

Som det ses af ovenstående pseudo-kode, er der tre mulige udfald for algoritmen; problemet er inkonsistent, problemet er ubegrænset eller problemet har en optimal løsning, som bestemmes af simplex-algoritmen.

4.6.7 Cykling

Når simplex-algoritmen bearbejder et lineært programmeringsproblem, er der en risiko for, at algoritmen ikke terminerer. Dette kaldes *cykling*. Da algoritmen bestemmer de næste handlinger ud fra nogle bestemte egenskaber i en simplex-tabel, er det givet, at hvis algoritmen støder på en tabel, den tidligere har set, vil den udføre nøjagtigt den samme handling. Det er her muligheden for cykling opstår. Hvis cykling ikke opstår, terminerer algoritmen efter et endeligt antal af pivoteringer, da algoritmen itererer igennem basale løsninger, hvilket der er et endeligt antal af. Det vil altså sige, at hvis cykling kan undgås, så er det garanteret, at algoritmen terminerer efter et endeligt antal af pivoteringer.

Årsagen til cykling er degenerering i simplex-tabellen. Når en degeneret basal løsning bestemmes, er der nemlig en risiko for, at algoritmen cykler igennem de samme baser. Betragt følgende cyklus i et maksimeringsproblem, forårsaget af degenerering, hvor der anvendes en Mindste-højeste-pivoteringsregel. Dvs. algoritmen vælger mindste indeks ved valg af pivoteringssøjlen og højeste indeks ved valg af pivoteringsrække:

1	11	18	-5	1	0	0	0		1	11	18	-5	1	0	0	0
-1	-3	2	-1	0	1	0	0	1 → 5	0	8	20	-6	1	1	0	0
-24	-288	270	-123	0	0	1	0		0	-24	702	-243	24	0	1	0

	8	0	-76	26	-3	-11	0	0
2 → 6	0	8	20	-6	1	1	0	0
	0	0	6096	-2088	216	24	8	0

4 → 1	64	0	-608	208	-24	-88	0	0
	48	208	64	0	8	-40	0	0
	16704	0	-192	0	-648	-22344	208	0
3 → 2	520	1976	0	208	52	-468	0	0
	48	208	64	0	8	-40	0	0
	16848	624	0	0	-624	-22464	208	0
5 → 4	520	1976	0	208	52	-468	0	0
	-32	-96	64	-32	0	32	0	0
	23088	24336	0	2496	0	-28080	208	0
6 → 3	1	11	18	-5	1	0	0	0
	-1	-3	2	-1	0	1	0	0
	-24	-288	270	-123	0	0	1	0

Efter 6 pivoteringer har simplex-algoritmen foretaget en cyklus, og tabellen er tilbage til den oprindelige. Bemærk, at de basale løsninger, bestemt fra tabellerne, alle er degenererede og ens, $x = [0, 0, 0, 0, 0, 0, 0]^T$. En pivotering, som ikke ændrer den basale løsning, kaldes en *degenereret pivotering*.

Fordelen ved at benytte Mindste Indeks-metoden, som der i dette projekt tages udgangspunkt i, er, at algoritmen undgår cykling og dermed altid terminerer.

Sætning 4.6.3. *Blands Sætning*

Mindste Indeks-simplex-algoritmen terminerer for ethvert lineært optimeringsproblem på standardform.

Bevis. Baseret på Burke [n.d.].

Antag, at der eksisterer en cyklus af ordninger på standardform, $O_0, O_1, \dots, O_N = O_0$. Når der foretages pivoteringer i denne cyklus, så skal variable, der forlader basen, naturligvis også indgå i basen igen. Disse variable kaldes *omskiftelige*. Det samme gælder for de variable, der kommer ind i basen under en pivotering, som da skal forlade basen igen. Lad F betegne mængden af omskiftelige variable. Alle omskiftelige variable antager værdien 0 i alle basale mulige løsninger i cyklussen, da det er degenererede pivoteringer, og de antager værdien 0, når de ikke er blandt de basale variable. Mængden β indeholder indeksene for de basale variable i O , og δ indeksene for de ikke-basale variable i O . Lad l angive det største indeks i F , og lad O :

$$x_i = b_i - \sum_{j \in \delta} a_{i,j} x_j \quad (i \in \beta)$$

$$z = v + \sum_{j \in \delta} c_j x_j,$$

være et ordnet problem i cyklussen, hvor x_l forlader basen, og lad x_e betegne variabelen, der er indgående i basen, dvs. $e \rightarrow l$.

Da x_e også er omskiftelig, gælder der, at $e < l$.

Da hver ordning i cyklussen angiver den samme basale mulige løsning, forbliver objekt-funktionsværdien konstant igennem cyklussen.

Lad O' være et ordnet problem i cyklussen, hvor x_l er indgående. Objektfunktionen for O' skrives:

$$z = v + \sum_{j=1}^{m+n} c'_j x_j,$$

hvor $c'_j = 0$, når $j \in \beta$. Bemærk, at $c'_j \leq 0$ for alle $j \in F \setminus \{l\}$.

Da O og O' indgår i samme cyklus, har de samme objektfunktionsværdi og angiver samme basale mulige løsning, så derfor skal løsningen for O opfylde O' . Hvorens løsningen for O omskrives, ved at den indgående variabel skrives $x_e = t$, og de ikke-basale variable skrives, $x_j = 0$, hvor $j \in \delta$, og $j \neq e$, hvilket giver:

$$x_i = b_i - a_{i,e}t \quad (i \in \beta)$$

$$z = v + c_e t.$$

Dermed fås:

$$v + c_e t = v + c'_e t + \sum_{i \in \beta} c'_i (b_i - a_{i,e}t) \quad \text{for alle } t.$$

Det omskrives til:

$$(c_e - c'_e + \sum_{i \in \beta} c'_i a_{i,e})t = \sum_{i \in \beta} c'_i b_i \quad \text{for alle } t.$$

Da alle b -ratioer er 0 grundet degenerering, er højre side lig nul, og det samme er koefficienten på venstre side:

$$c_e - c'_e = - \sum_{i \in \beta} c'_i a_{i,e}.$$

Da x_e er indgående i basen for O , så er $c_e > 0$. Variabelen, x_l , er indgående i O' , og $e < l$, da l er det højeste indeks i F . Dette medfører, at $c'_e \leq 0$, da MI-metoden ellers ville vælge x_e , som indgående variabel i O' . Dermed er $c_e - c'_e > 0$, og det betyder, at $\sum_{i \in \beta} c'_i a_{i,e} < 0$.

Så for et $s \in \beta$ gælder, $c'_s a_{s,e} < 0$. Da indekset, s , tilhører mængden, β , så er x_s en basal variabel i O , og fordi $c'_s \neq 0$, så er x_s en ikke-basal variabel i O' . Det medfører, at x_s er omskiftelig og tilhører mængden, F , hvilket medfører $s \leq l$.

Der vises nu, at $s < l$.

Da x_l er den udgående variabel i O , og x_e er den indgående, så er $a_{l,e} > 0$. Da x_l er den indgående i O' , så er $c'_l > 0$. Da begge er større end 0, gælder:

$$c'_l a_{l,e} > 0,$$

som medfører, at $s \neq l$, og dermed $s < l$.

Variablen, x_s , kan ikke betragtes, som en kandidat til at være den indgående variabel i O' , hvis MI-metoden følges, grundet $s < l$. Da x_l er den indgående i O' , gælder der, at $c'_s < 0$.

Tidligere blev $c'_s a_{s,e} < 0$ bestemt, så hvis $c'_s < 0$, så er $a_{s,e} > 0$. Variablen, x_s , er omskiftelig og antager derfor værdien 0 i den basale løsning. Så $b_s = 0$ og $a_{s,e} > 0$ medfører $b_s/a_{s,e} = 0$, hvilket udgør den minimale b-ratio i O . Derfor kan x_s betragtes som en mulig udgående variabel i O , men x_l er den udgående i O' , og $s < l$, hvilket modstrider MI-metoden, og dermed kan ingen cyklus eksistere. \square

I praksis forekommer cykling sjældent, og derfor kan der vælges at benytte andre metoder med lavere kompleksitet.

4.6.8 Simplex-eksempel

Som vist i Afsnit 4.3, findes optimeringsproblemets basale mulige løsninger i hjørnepunkterne af det polytop, som mulighedsområdet udgør. Geometrisk set, fungerer simplex-algoritmen netop ved at gennemløbe polytopets hjørnepunkter for at bestemme den optimale løsning. En pivotering i matrix-repræsentationen af optimeringsproblemet svarer således til at bevæge sig videre til et nabo-hjørnepunkt. Da algoritmen hele tiden bevæger sig i maksimerende retning, slipper den for at gennemløbe alle polytopets hjørner. Simplex-algoritmens fremgangsmåde vil nu blive eksemplificeret vha. et simpelt optimeringsproblem i \mathbb{R}^2 , som dermed også kan illustrere sammenhængen mellem tabel/matrix-fremstilling og en geometrisk fremstilling af problemet.

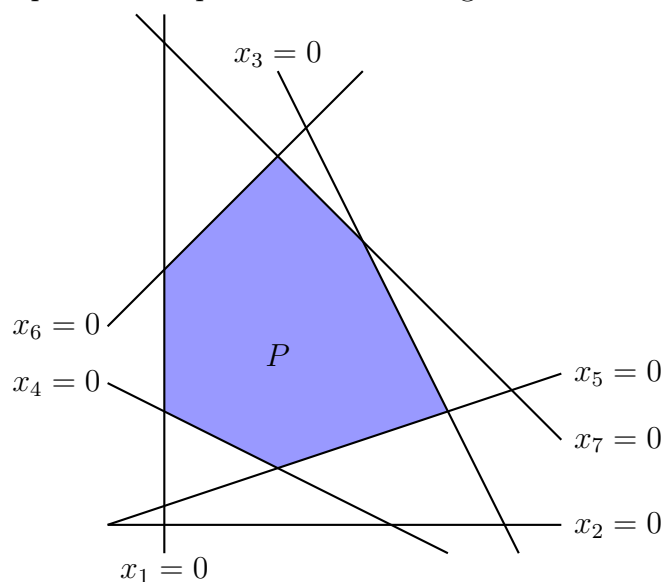
Betragt optimeringsproblemet givet ved:

$$\begin{aligned} \text{Maksimér } z &= 3x_1 + 2x_2 \\ \text{betinget af } 2x_1 + x_2 &\leq 12 \\ -x_1 - 2x_2 &\leq -4 \\ x_1 - 3x_2 &\leq -1 \\ -2x_1 + 2x_2 &\leq 9 \\ 2x_1 + 2x_2 &\leq 17 \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Nu indføres der slack-variable, og variablerne samles på venstresiden:

$$\begin{aligned}
 &\text{Maksimér } z - 3x_1 - 2x_2 = 0 \\
 &\text{betinget af } 2x_1 + x_2 + x_3 = 12 \\
 &\quad -x_1 - 2x_2 + x_4 = -4 \\
 &\quad \quad x_1 - 3x_2 + x_5 = -1 \\
 &\quad -2x_1 + 2x_2 + x_6 = 9 \\
 &\quad \quad 2x_1 + 2x_2 + x_7 = 17 \\
 &\quad \quad \mathbf{x} \geq \mathbf{0}
 \end{aligned}$$

Mulighedsområdet for optimeringsproblemet er skitseret herunder i Figur 4.5. Hvis slack-variablene sættes til nul, udgør de forskellige betingelser selvfølgelig rette linjer i \mathbb{R}^2 . Linjerne er således nulpunkter for problemets forskellige variable.



Figur 4.5: Mulighedsområdet P

Den tilsvarende totalmatrix kommer til at se således ud:

Tabel 4.2

2	1	1	0	0	0	0	0	12
-1	-2	0	1	0	0	0	0	-4
1	-3	0	0	1	0	0	0	-1
-2	2	0	0	0	1	0	0	9
2	2	0	0	0	0	1	0	17
-3	-2	0	0	0	0	0	1	0

Ved at betragte Figur 4.5, kan det uden videre ses, at $\mathbf{x}^T = [0, 0]$ er uden for P . Af tabellen kan det ses, at det er x_4 og x_5 , som overtræder ikke-negativitetsbetingelserne, da

både b_2 og b_3 er negative. Simplex-algoritmen tager altså udgangspunkt i en ikke-mulig basal løsning, og befinder sig dermed i Fase 1. Det kan også ses på problemets oprindelige uligheder, at to af betingelserne er overtrådt, hvis $\mathbf{x} = \mathbf{o}$. Simplex-algoritmen udfører pivotoperationen, $1 \rightarrow 4$ (b_2 er mest negativ, derfor pivot i række to. Søjle 1 vælges, da algoritmen i Fase 1 blot vælger laveste indeks), og resultatet ses på Tabel 4.3:

Tabel 4.3

0	-3	1	2	0	0	0	0	4
1	2	0	-1	0	0	0	0	4
0	-5	0	1	1	0	0	0	-5
0	6	0	-2	0	1	0	0	17
0	-2	0	2	0	0	1	0	9
0	4	0	-3	0	0	0	1	12

Det ses, at b_3 er negativ. Basen er fortsat ikke-mulig, så algoritmen foretager en ny pivotering for at bringe x_2 ind i basen og fjerne x_5 , $2 \rightarrow 5$. Bemærk, at vi herfra tillader os at gange alle indgange i tabellen med et heltal, for at undgå brøker, se Tabel 4.4:

Tabel 4.4

0	0	5	7	-3	0	0	0	35
5	0	0	-3	2	0	0	0	10
0	5	0	-1	-1	0	0	0	5
0	0	0	-4	6	5	0	0	55
0	0	0	8	-2	0	5	0	55
0	0	0	-11	4	0	0	5	40

Tabellen er nu mulig, og simplex-algoritmen er så i Fase 2. Geometrisk set, har algoritmen, via de to pivoteringer, bevæget sig ind i mulighedsområdet ved først at bevæge sig til punktet, $x_2 = x_4 = 0$, og derefter videre til $x_4 = x_5 = 0$. Det er punktet $[x_1, x_2] = [2, 1]$, hvor $z = 8$.

Algoritmens næste trin er at bringe x_4 ind i basen igen og gøre x_3 til en ikke-basal variabel. Algoritmen bevæger sig altså langs linjen, $x_5 = 0$, til punktet, $x_3 = x_5 = 0$. Tabellen kommer til at se ud som vist i Tabel 4.5:

Tabel 4.5

0	0	5	7	-3	0	0	0	35
7	0	3	0	1	0	0	0	35
0	7	1	0	-2	0	0	0	14
0	0	4	0	6	7	0	0	105
0	0	-8	0	2	0	7	0	21
0	0	11	0	-1	0	0	7	133

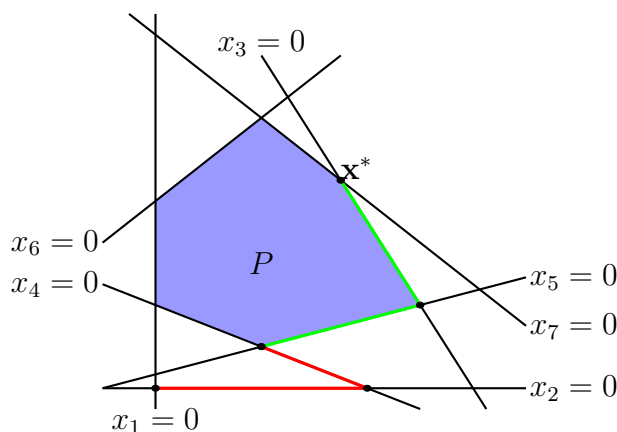
Algoritmen har bevæget sig til punktet $[x_1, x_2] = [5, 2]$, hvor objektfunktionen har værdien $z = 19$. Nu foretages en pivotering, $5 \rightarrow 7$, hvor x_5 bringes ind i basen igen, og algoritmen bevæger sig til $x_3 = x_7 = 0$:

Tabel 4.6

0	0	-14	14	0	0	21	0	133
14	0	14	0	0	0	-7	0	49
0	14	-14	0	0	0	14	0	70
0	0	56	0	0	14	-42	0	84
0	0	-56	0	14	0	49	0	147
0	0	14	0	0	0	7	14	287

Koefficienterne i objektfunktionsrækken er alle positive, så det er ikke længere muligt at øge værdien af objektfunktionen. Tabellen er derfor optimal, og algoritmen stopper. Objektfunktionens optimale værdi kan aflæses til $z^* = \frac{287}{14} = \frac{41}{2}$. Den optimale basale mulige løsning er fundet i punktet $x_3 = x_7 = 0$, eller udtrykt ved de oprindelige objektvariable, $\mathbf{x}^{*T} = [x_1, x_2] = [\frac{7}{2}, 5]$.

Algoritmens gennemløb er illustreret på Figur 4.6 herunder:



Figur 4.6: Simplex-algoritmens bevægelse

Fase 1-trinene er markeret med rødt, mens Fase 2-trinene er markeret med grønt.

4.6.9 Komplexitet

Baseret på Matoušek and Gärtner [2006].

Beregning af kompleksitet af de forskellige algoritmer for simplex-metoden er uden for projektets rammer, men en kort redegørelse tjener til at give en bedre forståelse for, hvorfor nogle algoritmer bliver valgt frem for andre. Efter skabelsen af simplex-algoritmen for 40 år siden, samt efter fundet af mange nye pivoteringsmetoder, er kompleksiteten af mange anvendte algoritmers worst-case scenarie stadig af eksponentiel tid. Det kan alligevel vises, at man sjældent foretager mere end $3m$ pivoteringer under løsningen af et typisk lineært programmeringsproblem, hvilket altså er lineær kompleksitet.

Mindste Indeks-simplex-algoritmen har en kompleksitet på $O\left(e^{C \cdot \sqrt{n \ln(n)}}\right)$. Den er altså forholdsvis ikke lige så effektiv, som andre metoder. Selvom den stopper degenerering, bliver den ofte ikke brugt, da cykling eller ligeværdige pivoteringsmuligheder ikke forekommer særlig ofte, hvorfor andre algoritmer med lavere gennemsnitlig kompleksitet normalt anvendes til udregninger.

4.6.10 Dual simplex-metoden

Dual simplex anvender det duale problem til løsning af optimeringsproblemer. Fremgangsmåden er den samme som ved almindelig simplex. Dog er der en ændring, at der begynder med at skifte det primale problem til det duale. Dette kan være nyttigt, hvis f.eks. $m - n$ er stor, da det duale vil have væsentligt færre basale mulige løsninger.

En anden god grund vil være, hvis en optimal værdi er fundet, men der herefter tilføjes en ny betingelse. Dette er en fordel, fordi det duale forbliver inden for mulighedsområdet. Ved at gå til det duale, kan der derved begynde ved den oprindelige løsning og arbejde derfra, i stedet for at begynde forfra.

Metoden kan også bruges, hvis den oprindelige tabel til det primale problem ikke er mulig. Som alternativ til at benytte algoritmens Fase 1, kan der skiftes til det duale problem, som måske begynder ved en basal mulig løsning.

4.6.11 Resten af beviset for Dualitetssætningen

Nu haves alle oplysninger for at foretage den sidste del af beviset for Dualitetssætningen (4.2.5). Det skal vises, at der eksisterer optimale vektorer til både det primale og det duale problem, og at problemernes optimale værdier er ens. Der tages udgangspunkt i Hurlbert [2009] og Chiarandini [2015].

Det oprindelige primale og det duale problem kan skrives på algebraisk form:

$$\text{Maksimér} \quad z = \sum_{j=1}^n c_j x_j \quad (4.5)$$

$$\sum_{j=1}^n a_{i,j} x_j \leq b_i \quad (1 \leq i \leq m) \quad (4.6)$$

$$x_j \geq 0, \quad (1 \leq j \leq n) \quad (4.7)$$

$$\text{Minimér} \quad w = \sum_{i=1}^m b_i y_i \quad (4.8)$$

$$\sum_{i=1}^m a_{i,j} y_i \geq c_j \quad (1 \leq j \leq n) \quad (4.9)$$

$$y_i \geq 0, \quad (1 \leq i \leq m) \quad (4.10)$$

Bevis. For at vise, at sætningen gælder, skal der for en optimal løsning, $\mathbf{x}^* = [x_1^*, \dots, x_n^*]$, findes en dual optimal løsning, $\mathbf{y}^* = [y_1^*, \dots, y_m^*]$, sådan at $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$.

Simplex-metoden finder altid en optimal løsning, hvis denne eksisterer. Derfor tages metoden i brug for at løse dette problem, og der vil blive brugt et direkte bevis. Lad \mathbf{x}^* være den optimale løsning for det primale problem og antag, at $\mathbf{c}^T \mathbf{x}^* = \mathbf{b}^T \mathbf{y}^*$ eller med andre ord $z^* = w^*$.

Det skal vises, at \mathbf{y}^* er en mulig løsning for det duale, så vil svag dualitet, Sætning 4.2.2, føre til, at \mathbf{y}^* er den optimale løsning.

I den primale optimale simplex-tabel, hvor slack-variablene er blevet indført, skrives følgende ligning svarende til objektfunktionen:

$$z = z^* - \sum_{j=1}^n c_j^* x_j - \sum_{i=1}^m c_{n+i}^* x_{n+i}. \quad (4.11)$$

Bemærk, at c_j^* er koefficienterne for x_j i den optimale objektrække af simplex-tabellen. Da \mathbf{x}^* er den optimale løsning for det primale, haves også ifølge (4.5):

$$z^* = \sum_{j=1}^n c_j x_j^*. \quad (4.12)$$

Tilsvarende kan skrives for den duale (på baggrund af (4.8)) og antagelsen, $z^* = w^*$:

$$z^* = \sum_{i=1}^m y_i^* b_i. \quad (4.13)$$

Der gættes på en smart løsning, som skal give en relation mellem objektfunktionens rækker for de to problemer; det primale og det duale:

$$y_{m+j}^* = c_j^* \text{ og } y_i^* = c_{n+i}^*, \text{ hvor } i = 1, 2, \dots, m \text{ og } j = 1, \dots, n. \quad (4.14)$$

Der skal vises, at \mathbf{y}^* , defineret i (4.14), opfylder både (4.9), (4.10) og ligningen, (4.13). Der startes med at vise, at ligningen, (4.13), gælder.

I ligning (4.11) indsættes (4.14):

$$z = z^* - \sum_{j=1}^n y_{m+j}^* x_j - \sum_{i=1}^m y_i^* x_{n+i}. \quad (4.15)$$

Fra definitionen af slack-variable have:

$$x_{n+i} = b_i - \sum_{j=1}^n a_{i,j} x_j. \quad (4.16)$$

Indføres både (4.5) og (4.16) i (4.15), så fås:

$$\begin{aligned} \sum_{j=1}^n c_j x_j &= z^* - \sum_{j=1}^n y_{m+j}^* x_j - \sum_{i=1}^m y_i^* \left(b_i - \sum_{j=1}^n a_{i,j} x_j \right) \\ &= \left(z^* - \sum_{i=1}^m y_i^* b_i \right) + \sum_{j=1}^n x_j \left(\left(\sum_{i=1}^m a_{i,j} y_i^* \right) - y_{m+j}^* \right). \end{aligned} \quad (4.17)$$

Men da (4.17) skal gælde for alle x_j , sættes $x_j = 0$, og med dét bliver den første parentes lig 0. Dermed er (4.13) bevist.

Efter beviset før reduceres (4.17) til:

$$\sum_{j=1}^n c_j x_j = \sum_{j=1}^n \left(\left(\sum_{i=1}^m a_{i,j} y_i^* \right) - y_{m+j}^* \right) x_j.$$

Ved en lignende tankegang vælges nu $x_j = 1$, så fås:

$$c_j = \sum_{i=1}^m a_{i,j} y_i^* - y_{m+j}^*. \quad (4.18)$$

Denne ligning kan opskrives for forskellige c_j for at se, hvad det betyder. For eksempel for $j = 1$, fås:

$$c_1 = \sum_{i=1}^m a_{i,1} y_i^* - y_{m+1}^*.$$

Det kan vises, at ligningen, (4.9), gælder for alle c_k skrevet på en tilsvarende måde som c_1 .

Erstattes y_{m+j}^* med det tilsvarende c_j^* i ligning, (4.18), fås:

$$c_j = \sum_{i=1}^m a_{i,j} y_i^* - c_j^* \iff c_j - \sum_{i=1}^m a_{i,j} y_i^* = -c_j^*.$$

Det vides at, alle $c_j \geq 0$ i den optimale simplex-tabellen og derfor:

$$-c_j^* \leq 0 \Rightarrow c_j - \sum_{i=1}^m a_{i,j} y_i^* \leq 0 \Rightarrow \sum_{i=1}^m a_{i,j} y_i^* \geq c_j, \quad j = 1, \dots, n.$$

Dette viser, at (4.9) gælder.

For slack-variable:

$$-c_{n+i}^* \leq 0 \Rightarrow y_i^* = c_{n+i}^* \geq 0 \text{ dvs. } y_i^* \geq 0, \quad i = 1, \dots, m.$$

Med dét er (4.10) også vist.

Da ligningerne, (4.9), (4.10) og (4.13), gælder, så er \mathbf{y}^* den optimale løsning for dualen, hvormed Dualitetssætningen er bevist. \square

4.7 Implementering af simplex-algoritmen

Koden, som er udarbejdet i Python 3.6, anvender Blands simplex-algoritme til Fase 2, og shortcut-metoden til Fase 1. Dvs. den svarer til den pseudokode, som blev opstillet i Afsnit 4.6.6. Koden er vedlagt som Bilag 6.1.

Nedenfor er outputtet til konsollen, når koden er anvendt på problemet i Afsnit 4.6.3.

Oprindelig matrix:

```
[[ 2.  1.  1.  0.  0.  0. -12.]
 [ -1. -2.  0.  1.  0.  0. -2.]
 [ 1. -3.  0.  0.  1.  0. -1.]
 [ -3. -2.  0.  0.  0.  1.  0.]]
```

Fase 1:

Systemet er inkonsistent

Process finished with exit code 41

Algoritmen anerkender, at systemet er inkonsistent.

Til simplex-eksemplet i Afsnit 4.6.8 giver koden følgende output:

Oprindelig matrix:

```
[[ 2.  1.  1.  0.  0.  0.  0.  0. 12.]
 [-1. -2.  0.  1.  0.  0.  0.  0. -4.]
 [ 1. -3.  0.  0.  1.  0.  0.  0. -1.]
 [-2.  2.  0.  0.  0.  1.  0.  0.  9.]
 [ 2.  2.  0.  0.  0.  0.  1.  0. 17.]
 [-3. -2.  0.  0.  0.  0.  0.  1.  0.]]
```

Fase 1:

Pivoterer ved søjle 2 og række 1

```
[[ 0. -3.  1.  2.  0.  0.  0.  0.  4.]
 [ 1.  2. -0. -1. -0. -0. -0. -0.  4.]
 [ 0. -5.  0.  1.  1.  0.  0.  0. -5.]
 [ 0.  6.  0. -2.  0.  1.  0.  0. 17.]
 [ 0. -2.  0.  2.  0.  0.  1.  0.  9.]
 [ 0.  4.  0. -3.  0.  0.  0.  1. 12.]]
```

Pivoterer ved søjle 3 og række 2

```
[[ 0.  0.  1.  1.4 -0.6  0.  0.  0.  7. ]
 [ 1.  0.  0. -0.6  0.4  0.  0.  0.  2. ]
 [-0.  1. -0. -0.2 -0.2 -0. -0. -0.  1. ]
 [ 0.  0.  0. -0.8  1.2  1.  0.  0. 11. ]
 [ 0.  0.  0.  1.6 -0.4  0.  1.  0. 11. ]
 [ 0.  0.  0. -2.2  0.8  0.  0.  1.  8. ]]
```

Fase 2:

Pivoterer ved søjle 1 og række 4

```
[[ 0.  0.  0.714  1. -0.428  0.  0.  0.  5. ]
 [ 0.  1.  0.142  0. -0.285  0.  0.  0.  2. ]
 [ 1.  0.  0.428  0.  0.142  0.  0.  0.  5. ]
 [ 0.  0.  0.571  0.  0.857  1.  0.  0. 15. ]
 [ 0.  0. -1.142  0.  0.285  0.  1.  0.  3. ]
 [ 0.  0.  1.571  0. -0.142  0.  0.  1. 19. ]]
```

Pivoterer ved søjle 5 og række 5

```
[[ 0.  0. -1.  1.  0.  0.  1.5  0.  9.5]
 [ 1.  0.  1.  0.  0.  0. -0.5  0.  3.5]
 [ 0.  1. -1.  0.  0.  0.  1.  0.  5. ]
 [ 0.  0.  4.  0.  0.  1. -3.  0.  6. ]
 [ 0.  0. -4.  0.  1.  0.  3.5  0. 10.5]
 [ 0.  0.  1.  0.  0.  0.  0.5  1. 20.5]]
```

Process finished with exit code 0

Som forventet, giver Python-koden samme resultat, $z^* = \frac{41}{2}$, som vi fik ved at arbejde os igennem algoritmen manuelt. Vores test tyder altså på, at koden virker efter hensigten.

5 | Konklusion

Simplex-algoritmen er et effektivt redskab til at løse lineære optimeringsproblemer. Algoritmen fungerer ved at afsøge de basale mulige løsninger, som svarer til hjørnepunkterne i det polytop, som udgør mulighedsområdet. Pivoteringsreglerne sikrer, at algoritmen bevæger sig i maksimerende retning, og at den dermed slipper for at undersøge samtlige basale mulige løsninger. Mindste Indeks-reglen sikrer, at algoritmen undgår at cykle, og det garanteres dermed, at algoritmen altid terminerer.

Simplex-algorithmens *worst case*-kompleksitet er eksponentiel, men i praksis har algoritmen typisk lineær kompleksitet.

Projektgruppen har lavet en implementering af simplex-algoritmen i Python, og testet vores kode på en række eksempler. Konklusionen er, at simplex er effektiv og relativt enkel at implementere.

6 | Bilag

6.1 Appendiks

```
import numpy as np

if __name__ == '__main__':

    print('Oprindelig matrix:')
    # Til hvis systemet er inkonsistent
    A = np.array([
        [2., 1., 1., 0., 0., 0., -12.],
        [-1., -2., 0., 1., 0., 0., -2.],
        [1., -3., 0., 0., 1., 0., -1.],
        [-3., -2., 0., 0., 0., 1., 0.]
    ])
    # Til hvis systemet er ubegrænset
    A = np.array([
        [2., 0., 1., 0., 0., 0., 12.],
        [-1., 0., 0., 1., 0., 0., -2.],
        [1., 0., 0., 0., 1., 0., 20.],
        [-1., -1., 0., 0., 0., 1., 0.]
    ])
    # Til eksemplet med simplex
    A = np.array([
        [ 2.,  1., 1., 0., 0., 0., 0., 0., 12.],
        [-1., -2., 0., 1., 0., 0., 0., 0., -4.],
        [ 1., -3., 0., 0., 1., 0., 0., 0., -1.],
        [-2.,  2., 0., 0., 0., 1., 0., 0.,  9.],
        [ 2.,  2., 0., 0., 0., 0., 1., 0., 17.],
        [-3., -2., 0., 0., 0., 0., 0., 1.,  0.]
    ])

    def erBasal(A, i, j):          #ser om søjle j er basal
        n, m = len(A[0]), len(A)
```

```

    for r in range(0, m):
        if r == i: continue
        if A[r][j] != 0:
            return False
    return True

def pivotSC(A):
    n, m = len(A[0]), len(A)
    mindstB, mindstI, mJ, mI = 0, 0, -1, -1
    for b in range(0, m - 1):
        temp = A[b][n - 1]
        if temp < mindstB:
            mindstB, mI = temp, b
        elif temp == mindstB: #####
            for k in range(0, n - 1): #
                if erBasal(A, mI, k): #
                    break # Til hvis b værdierne
                elif erBasal(A, b, k): # er ens
                    mindstB, mI = temp, b #
                    break #####
    for j in range(0, n - 1):
        temp = A[mI][j]
        if temp < mindstI:
            mindstI, mJ = temp, j
        break

    if mindstI == 0:
        print('Systemet er inkonsistent')
        exit(41)
    return mI, mJ

def pivotSted(A):
    n, m = len(A[0]), len(A)
    global fase1
    # Til fase 1
    for i in range(0, m - 1):
        if A[i][n - 1] < 0:

            if not fase1:
                print('Fase 1:')
                fase1 = True

```

```

        return pivotSC(A)

    # Til fase 2
    if fase1:
        print('Fase 2:')
        fase1 = False
    ratio, mJ, mI = 999999999999999999, 0, 0
    for j in range(0, n - 1):
        if A[len(A) - 1][j] < 0:
            ubegranset = True
            for i in range(0, m - 1):
                if A[i, j] > 0:
                    ubegranset = False
                    temp = ((A[i, n - 1]) / (A[i, j]))
                    if temp < ratio:
                        ratio, mJ, mI = temp, j, i
            if ubegranset:
                print('Systemet er ubegrænset.')
                exit(42)
    return mI, mJ

def pivot(A, i, j):
    n, m, e = len(A[0]), len(A), A[i][j]
    A[i] = A[i] / e
    for r in range(m):
        if r == i: continue
        A[r] = A[r] - A[r][j] * A[i]

def check(A):
    # print('checking')
    n, m = len(A[0]), len(A)
    for i in range(n - 1):
        if A[m - 1][i] < 0:
            return False
    return True

fase1 = False
done = False
print(A)
print('')
```



```
while not done:
    i, j = pivotSted(A)
    print('Pivoterer ved søjle {:d} og række {:d}'.format(i+1, j+1))
    if i == -1 and j == -1:
        print('færdig')
        break
    pivot(A, i, j)
    done = check(A)
    print(A)
    print('')
```

7 | Bibliografi

Bertsimas, D. and Tsitsiklis, J. N. [1997], *Introduction to Linear Optimization*, Massachusetts Institute of Technology.

Burke, J. [n.d.], ‘Jim burke, does the simplex algorithm work?’, University of Washington. Visited: 2019-22-05.

URL: <https://sites.math.washington.edu/~burke/crs/407/lectures/L7-SA-termination.pdf?fbclid=IwAR1X-4mhC2f3l9CdkB9Y742-ZmfFy7BGDvgLiSDz6YTJW35814t6nzNWYI>

Chiarandini, M. [2015], *Linear and Integer Programming- Lecture Notes*, McGraw-Hill Education.

Ferguson, T. S. [2017], *LINEAR PROGRAMMING - A Concise Introduction*.

(<https://math.stackexchange.com/users/3736/will-orrick>), W. O. [n.d.], ‘Why does the “auxiliary problem” method work to find a feasible dictionary?’, Mathematics Stack Exchange. URL:<https://math.stackexchange.com/q/146664> (version: 2012-05-23).

URL: <https://math.stackexchange.com/q/146664>

Hurlbert, G. H. [2009], *Linear Optimization*, Springer.

Linear programming [2019]. 14/5/2019.

URL: <https://www.britannica.com/science/linear-programming-mathematics>

Matoušek, J. and Gärtner, B. [2006], *Understanding and Using Linear Programming*, Springer.

Spence, L. E., Insel, A. J. and Friedberg, S. H. [2015], *Elementary Linear Algebra 2015*, second edition edn, Pearson Prentice Hall.

The Equivalence of the Auxiliary and Shortcut Methods for the Simplex Algorithm [2008]. Visited: 14/5/2019.

URL: <http://www.people.vcu.edu/~%7eghurlbert/papers/EASMSA.pdf>