

TAOP14: Optimeringslära

SAMMANFATTNING

OSKAR QVIST: OSKQV953@STUDENT.LIU.SE

Innehållsförteckning

Allmänt	2
Om optimering	3
Matematiska formuleringar av optimeringsproblem.....	3
Linjärprogrammering (LP)	4
Grafisk lösning av LP-problem	4
Det tillåtna området	5
Målfunktionsgradienten.....	5
Formulering av LP-problem	5
Skattningar av LP-problem	6
Optimistisk skattning.....	7
Pessimistisk skattning.....	7
Simplexmetoden	7
Standardform	7
Baslösning.....	8
Simplexalgoritmens sökmetod	10
Algebraisk version av simplexmetoden.....	12
Fas I.....	12
Att tolka simplextablån	15
Sammanfattning	15
Dualitet.....	16
Att ta fram det duala problemet	17
Relationen mellan primal och dual.....	18
Känslighetsanalys	21
Relaxation och Restriktion.....	21
Skuggpriset och dess intervall	21
Förändring av en målfunktionskoefficient	21
Införandet av en ny variabel	22
Nytt bivillkor	22
Icke-linjär programmering (ILP).....	22
Konvexitet.....	23
Konvex funktion.....	23
Konvexa mängder.....	24
Obegränsad optimering & sökmetoder	25
Sökriktning.....	25
Steglängd	27

Brantaste lutningsmetoden.....	29
Newtonmetoder	30
KKT-villkoren.....	30
KKT-villkoren i icke-konvexa problem och i LP-problem	33
Frank-Wolfe metoden	33
Lagrangedualitet & Lagrangerelaxation	34
Generell lösningsstrategi.....	36
Definitioner	37
Konvexitet.....	37
Optimalitet	38
Sökdiriktningar	39
Övrigt.....	39
Satser	43
Konvexitet.....	43
Dualitet.....	45
Obegränsad optimering.....	47
Övrigt.....	47
Källor.....	47

Allmänt

Det här häftet sammanfattar kursen TAOP14, Optimeringslära grundkurs, vid LiTH. Den är hastigt ihopsatt och riskerar därför innehålla ett eller annat fel – undertecknad vill reservera sig för det ☺. Stöter ni på några grövre fel eller har åsikter om vad som kan göras annorlunda/utvecklas så tar jag gärna emot förbättringsförslag per mejl. I huvudsak är den skriven för egen vinning och eget lärande, men jag hoppas också att den kan komma någon annan till glädje. Det är fritt fram att dela den vidare om ni tycker att den är något att ha!

Figurer och exempel är (med några få undantag) inte mina egna. Strukturen på sammanfattningen är densamma som för föreläsningarna och det mesta direkt på vad som har gått igenom på föreläsningarna. Dessvärre tröttnade jag lite mot slutet, så ambitionsnivån sjönk något – men det går alltid att komplettera innehållet här med läroboken.

I slutet av sammanfattningen har jag samlat de flesta av de satser och definitioner som förekommer. I löpande text hänvisar jag ibland dit, så det kan vara bra att känna till. De flesta satser och definitioner är dessutom kommenterade för extra tydlighet!

God läsning,
Oskar

Om optimering

Optimeringslära är den matematiska lära som beskriver olika metoder för hur ett optimalt värde (maximum/minimum) kan hittas i en funktion, oftast givet ett antal förutsättningar och restriktioner (bivillkor).

Det finns flera olika undergrenar av optimeringslära, men i den här sammanfattningen fokuseras det huvudsakligen på *linjärprogrammering* (LP) och *icke-linjär programmering* (ILP).

Matematiska formuleringar av optimeringsproblem

Optimeringsproblem kan rent matematiskt uttryckas på flera olika sätt. De vanligaste (enda?) är med hjälp av *vanliga likheter och olikheter*, på *summationsform* eller på *matrisform*. Summationsformen är i de flesta fall att föredra, då det är väldigt opraktiskt att formulera större optimeringsproblem på annat sätt. Det bör dock påpekas, hur grundläggande det än må vara, att det finns ett tydligt samband mellan de olika skrivsätten. Oavsett vilken form man får en uppgift på så kan man alltid på ett enkelt sätt skriva om det på vilket annat sätt som helst. Således kan det finnas en poäng i att känna till på vilket sätt detta görs, även om det inte är något som detta avsnitt kommer att fästa någon särskild tyngd vid. I detta avsnitt följer en kortare genomgång av de olika skrivsätten.

Vanliga likheter och olikheter

De vanliga likheterna/olikheterna är det mest enklaste sättet att beskriva små problem på. Det är dock ganska opraktiskt vid större problem, då det blir onödigt mycket att skriva.

$$\begin{array}{rcl} \max & z = & 4x_1 + 3x_2 \\ \text{då} & & 2x_1 + 3x_2 \leq 30 \\ & & x_1 \leq 6 \\ & & 6x_1 + 4x_2 \leq 50 \\ & & x_1, x_2 \geq 0 \end{array}$$

Figur 1: Det "vanliga" skrivsättet. Holmberg, 2015

Matrisform

På matrisform samlas de olika koefficienterna och variablerna i det "vanliga" skrivsättet för att få en mer kompakt beskrivning av problemet. Koefficienterna och variablerna återfinns i matriser, vektorer och skalärer, istället för som ovan. Detta skrivsätt är egentligen inte särskilt praktiskt vid faktisk problemlösning, men det används med fördel vid rent algebraiska beskrivningar av olika satser och dylikt. Ett problem på matrisform ser ut som följer (med siffror från det föregående exemplet):

$$\begin{array}{l} \max z = \mathbf{c}^T \mathbf{x} \\ \text{då} \quad \mathbf{Ax} \leq \mathbf{b} \\ \quad \mathbf{x} \geq \mathbf{0} \end{array} \quad \mathbf{c}^T = (4 \quad 3) \quad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \quad \mathbf{A} = \begin{pmatrix} 2 & 3 \\ 1 & 0 \\ 6 & 4 \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} 30 \\ 6 \\ 50 \end{pmatrix}$$

Figur 2: Exempelproblemet på matrisform. Från FÖ1 (matriserna är dock mina egna).

Kommentar: Multiplicerar man ihop matriserna och vektorerna så får man ut problemet på "vanlig" form. Vid matematiska bevis och formler finns det en tydlig fördel i att kunna behandla problemet på denna form (utan att specificera vad de olika matriserna innehåller för data). Denna

form är dessutom, om man är bekväm med den, att föredra för sin "fusklapp" till tentamen, då den tar betydligt mycket mindre plats.

Summationsform

Summationsformen är den som generellt sett föredras i optimeringssammanhang. Den använder sig också av matriser och vektorer för att specificera koefficienter o.d., men använder indexering. Skrivsättet är såväl tydligt som kompakt, vilket är en stor fördel. Summationsformen är dessutom den mest använda i optimeringsprogram på datorer – och det är där de flesta problem behandlas i praktiken. Ett problem på summationsform (samma exempel som tidigare):

$\max z = \sum_{j \in J} c_j x_j$	$J = \{1 \ 2\},$	$I = \{1 \ 2 \ 3\}$
$\text{då } \sum_{j \in J} a_{ij} x_j \leq b_i, \quad i \in I$	$c_j = (4 \ 3),$	d.v.s: $c_1 = 4$ och $c_2 = 3$
$x_j \geq 0, \quad j \in J$	$a_{ij} = \begin{pmatrix} 2 & 3 \\ 1 & 0 \\ 6 & 4 \end{pmatrix}$	d.v.s. $a_{11} = 2$ och $a_{12} = 3$, etc.
	$b = \begin{pmatrix} 30 \\ 6 \\ 50 \end{pmatrix}$	d.v.s. $b_1 = 30, b_2 = 6$ & $b_3 = 50$

Figur 3: Exemplet på summationsform.
Indexmatriserna är mina egna. Från FÖ1.

Kommentar: Det är mycket viktigt att behärska summationsformen, då den är det viktigaste skrivsättet. Notera också att om man skriver ut summatecknen till fullt utskrivna summor så får man det "vanliga" skrivsättet. Observera att det i detta exempel inte undlättar särskilt mycket, men såfort problemen blir större så finns det enorma fördelar med detta skrivsätt.

Linjärprogrammering (LP)

Linjära optimeringsproblem är de mest grundläggande inom optimeringsläran. Här behandlar vi linjära målfunktioner och bivillkorsmängder, där det finns en övre och undre gräns för alla variabler. Dessa problem är förhållandevis enkla att lösa, i synnerhet då linjära problem per definition (se: [länk till definition]) alltid är konvexa. Vi vet också att optimum för linjära problem *alltid* kommer att ligga i (minst) en av bivillkorsmängdens hörnpunkter. Enklare linjära problem kan man, åtminstone när bara är två variabler, lösa [grafiskt](#). Den grafiska lösningen återkommer även inom den icke-linjära programmeringen då man ibland skapar delproblem som kan vara linjära.

I detta avsnitt kommer vi att titta på hur man [formulerar LP-problem](#) (modellering) och hur man löser dem med [simplexmetoden](#). Vi kommer också att kika närmare på det centrala begreppet [dualitet](#) och gå igenom hur man utför en enklare [känslighetsanalys](#).

Grafisk lösning av LP-problem

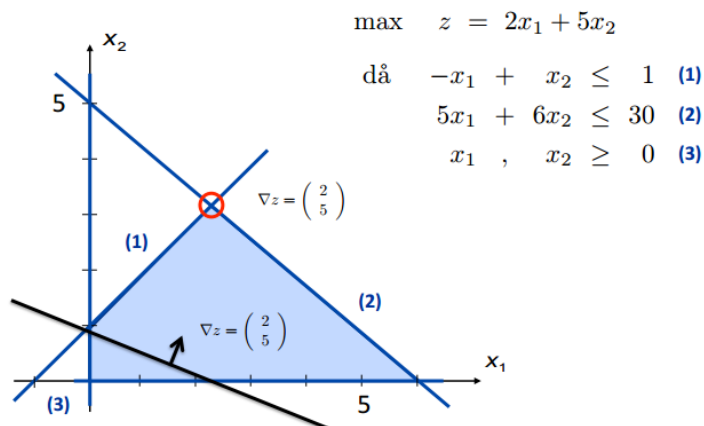
Att grafiskt lösa ett LP-problem i två variabler (i fler variabler blir det genast klurigare (eller omöjligt), men i denna kurs stöter vi bara på problemet i två) är mycket enkelt, varpå vi inte kommer att gå in särskilt djupt på det här. I korta drag kan man sammanfatta det som två steg:

1. Rita upp det tillåtna området (som definierat av bivillkoren).
2. Använd målfunktionens gradient för att avgöra vart optimum finns.

Vi tar det ett steg i taget och avslutar med ett helt exempel:

Det tillåtna området

Betrakta följande problem (från föreläsning 1):



Figur 4: Grafisk framställning av det tillåtna området och målfunktionsgradienten. Från FÖ1.

$$\max z = 2x_1 + 5x_2$$

$$\text{då } \begin{aligned} -x_1 + x_2 &\leq 1 & \text{(1)} \\ 5x_1 + 6x_2 &\leq 30 & \text{(2)} \\ x_1, x_2 &\geq 0 & \text{(3)} \end{aligned}$$

Här har vi två bivillkor som, tillsammans med teckenkraven på x_1 & x_2 , definierar det tillåtna området. För att rita upp detta kan vi antingen utnyttja det faktum att problemet är linjärt och "nolla" den ena variabeln för att hitta skärningspunkten för den andra (och vice versa), för att sedan dra ett rakt sträck emellan dem, eller skriva x_2 som en funktion av x_1 (för varje bivillkor).

För att få fram villkor två kan vi alltså göra på två sätt. Det ena är att skriva om det till en funktion av x_2 , det vill säga den klassiska "y=kx+m"-formen: $5x_1 + 6x_2 \leq 30 \Leftrightarrow x_2 \leq 5 - \frac{5}{6}x_1$. Ett andra alternativ är att bevara ursprungsolikheten (fast betrakta den som en likhet) och konstatera att om $x_2 = 0$ så måste $x_1 = 6$ och om $x_1 = 0$ så måste $x_2 = 5$. På så sätt har vi skärningspunkterna, och eftersom problemet är linjärt kan vi helt enkelt dra ett sträck emellan punkterna (6, 0) och (0, 5). Villkor ett kan lösas analogt.

Målfunktionsgradienten

Då gradienten (se: [hänvisa]) pekar i den riktning som funktionen ökar snabbast kan vi konstatera att det är i den riktningen som vi vill leta i ett maximiproblem (för ett minimiproblem går vi i motsatt riktning, d.v.s. "minus gradienten"). Ritar vi upp gradienten, och drar ett sträck som är vinkelrät mot den (alla punkter längst detta sträck kommer att vara likvärdiga i ett linjärt problem) så kan vi enkelt se vilken punkt i det tillåtna området som kommer att vara den bästa. Kom ihåg att problemet är linjärt, så vi vill gå så långt som vi bara kan i gradientens riktning för att hitta det optimala värdet.

Observera också att optimallösningen för ett LP-problem *ligger i (minst) en hörnpunkt*. Lösningen kan ligga i två hörnpunkter (vilket i sådana fall inkluderar samtliga punkter emellan dessa), vilket blir fallet då målfunktionens gradient ∇f är parallell med en av bivillkorens gradienter ∇g_i .

Formulering av LP-problem

Formulering (modellering) av LP-problem är ibland det klurigaste av allt eftersom det handlar om att konvertera verkliga förhållanden till matematiska uttryck, vilket i tidigare kurser knappt har förekommit.

Vid modellering finns det ett antal begrepp som man bör vara bekant med:

- **Parametrar** är de givna data som beskriver problemet. Det kan t.ex. vara resursbegränsningar, priser, tillgångar eller efterfrågan. I matrisformen och summationsformen (som beskrevs i det första avsnittet) är det parametrar som utgör matrisen \mathbf{A} och vektorerna \mathbf{b} och \mathbf{c} . Vanliga, standardiserade, benämningar på olika parametrar (i summationsform) är t.ex. b_{kt}

(tillgångar), a_{kj} (åtgång), p_{kt} (inköpspris), c_{jt} (försäljningspris), d_{jt} (efterfrågan, "demand"), där k = råvaruindex, t = tidsindex (t.ex. veckonummer) och j = produktindex.

- **Variabler** är det man ska fatta beslut om eller vill ha svar på. Det kan t.ex. vara produktion, försäljning, inköp, lager och transport, eller likvärd en blandning av dessa. De vanligaste benämningarna på variabler är x_{jt} (produktion), y_{kt} (inköp), L_{kt} (lager) och T_{mnt} (transport), där k = råvaruindex, t = tidsindex (t.ex. veckonummer), j = produktindex och m, n = platser.
- **Bivillkor** kopplar samman variabler och parametrar.

Ett exempel som man bör känna till är *lagerbalans*:

$$L_{kt} = L_{k,t-1} + y_{kt} - \sum_{j \in J} a_{kj} x_{jt} \quad \forall k, t$$

Ovanstående formel säger att lagret för varje råvara k i respektive vecka t består av det *ingående lagret* plus de i veckan inköpta råvarorna minus de råvaror som under veckan går åt för att producera produkt j . Här räknar lagersaldot i *slutet av veckan*.

Notera dock att varianter av detta förekommer och att man måste skraddarsy varje bivillkor efter de förutsättningar som anges i problemet.

- **Målfunktionen** brukar t.ex. försöka maximera vinst eller minimera kostnader. Ett typiskt exempel skulle vara: (Försäljning \times Intäkt) – (Inköp \times Inköpspris). Målfunktionen kan alltså bestå av flera olika komponenter, men en bra tumregel är att allt man *tjänar på* och allt som *kostar* ska vara med i målfunktionen. Det är ju trots allt detta som man vill maximera/minimera. Här ska man också, likt i bivillkoren, koppla samman variabler och parametrar. Det ter sig t.ex. naturligt att koppla ihop försäljningspriset med försäljningsmängden och inköpskostnaden med antalet enheter man köper in. Notera dock att exempelvis en produktionskostnad/inköpskostnad inte nödvändigtvis framgår explicit, utan att en produkt t.ex. kan bestå av flera olika material. Således måste man ofta räkna fram koefficienterna till målfunktionen.

Skattningar av LP-problem

Ofta behöver man inte räkna ut optimum, utan det viktiga är ibland att man får en uppfattning om vad som är rimligt och ett ungefärligt svar. Då gör man s.k. optimistiska- och pessimistiska skattningar, för att "stänga in" det optimala målfunktionsvärdet. Dessa är, för mindre LP-problem, förhållandevis enkla att göra – men det är viktigt att man säkerställer att skattningarna är antingen optimistiska eller pessimistiska. En optimistisk skattning får inte kunna visa sig vara pessimistisk och vice versa (dessutom bör det kanske klargöras att intervallet $-\infty \leq z^* \leq \infty$ inte är ett okej svar... ☺).

Noterbart är att visa algoritmer använder optimistiska och pessimistiska skattningar för att ta sig fram till optimum. Det ter sig ganska logiskt att när den optimistiska skattningen är den samma som den pessimistiska (om inga misstag har gjorts) så har vi hittat optimum. Således är begreppen centrala för såväl ämnet som kursen.

Optimistisk skattning

En optimistisk skattning är en uppskattning utav målfunktionsvärdet som är *för bra*. Det är viktigt att den alltid, med säkerhet, är bättre (eller lika bra) som det optimala målfunktionsvärdet. Notera att *för bra* har olika betydelser i maximi- och minimiproblem.

En optimistisk skattning kan göras på flera sätt, men det enklaste är ofta att ta bort bivillkor (att ta bort begränsningar kan aldrig ha en negativ inverkan på målfunktionen).

Pessimistisk skattning

En pessimistisk skattning är en skattning av målfunktionsvärdet som ger ett *för dåligt* (eller *lika dåligt*) värde. Notera att även här gäller det att *för dåligt* har olika betydelser i maximi- resp. minimiproblem.

Det enklaste sättet att göra en pessimistisk skattning på är oftast att bara hitta en punkt som uppfyller alla bivillkoren och sedan räkna ut målfunktionsvärdet för detta. Om en punkt uppfyller alla kraven, så kan det inte vara en bättre punkt än den optimala (den kan dock vara lika bra, men det är ju okej!).

Simplexmetoden

Simplexmetoden, som utvecklades av den amerikanska matematikern George Dantzig, är den överlägset mest använda algoritmen för att lösa LP-problem. Simplexmetoden utnyttjar det faktum att optimum för linjära problem alltid ligger i (minst) en hörnpunkt (se: *Linjärprogrammeringens fundamentalsats*). Algoritmen söker den bästa lösningen genom att systematiskt förflytta sig från en baslösning till en annan närliggande baslösning på ett sådant sätt att målfunktionsvärdet förbättras. Eftersom LP-problem är konvexa kommer den hörnpunkt som saknar intilliggande hörnen med bättre målfunktionsvärde att vara problemets optimallösning.

Simplexmetoden utgår ifrån att problemet är formulerat på *standardform*, så vi börjar med att gå igenom det:

Standardform

1. Se till att *alla variabler* $x_i \geq 0$.

Om $x_i \leq 0$: Gör variabelbytet $y_i = -x_i$, med $y_i \geq 0$.

Om x_i är "fri" (inget teckenkrav): Låt $x_i = y_i^+ - y_i^-$ där $y_i^+, y_i^- \geq 0$ ($y_i^+ = 0$ då $x_i \leq 0$ och $y_i^- = 0$ då $x_i \geq 0$).

2. Se till att *högerleden* ≥ 0 .

Högerleden brukar vara positiva, men skulle de inte vara det kan man alltid multiplicera vänster- och högerledet med -1.

3. Se till att få *likhet* för alla bivillkor.

För \leq -villkor: Fyll ut vänsterledet med en slackvariabel $+s_i$, ($s_i \geq 0$) för att få likhet.

För \geq -villkor: Reducera vänsterledet genom en slackvariabel $-s_i$, ($s_i \geq 0$).

Exempel: Standardform

Följande exempel ska skrivas om på standardform:

$$\begin{aligned} \max \quad z &= 3x_1 + 2x_2 + 4x_3 \\ \text{då} \quad 2x_1 + 3x_2 &\leq 12 \\ x_1 - 2x_2 + x_3 &\geq 7 \\ x_1 \geq 0, x_2 \leq 0, x_3 &\text{ fri} \end{aligned}$$

Vi börjar med att kolla på variablernas teckenkrav och ser att $x_2 \leq 0$ och att x_3 är "fri". Således ersätter vi x_2 med $-y_2$ och x_3 med $y_3^+ - y_3^-$.

Därefter kollar vi på högerleden och konstaterar att alla redan är positiva – inget behöver göras här.

Figur 5: Från FÖ3.

Sist ser vi till att skapa likhet. Vi ser då att vi har ett \leq -villkor och ett \geq -villkor. Alltså inför vi i det första bivillkoret en $+s_1$ och i det andra en $-s_2$. Exemplet på standardform ses nedan:

$$\begin{aligned} \max \quad z &= 3x_1 - 2y_2 + 4y_3^+ - 4y_3^- \\ \text{då} \quad 2x_1 - 3y_2 &+ s_1 = 12 \\ x_1 + 2y_2 + y_3^+ - y_3^- &- s_2 = 7 \\ x_1, y_2, y_3^+, y_3^-, s_1, s_2 &\geq 0 \end{aligned}$$

Figur 6: Exempel på standardform. Från FÖ3.

Kommentar: En anledning till att vi vill ha likhet i vårt problem är just att optimum kommer att finnas i en hörnpunkt, d.v.s. där det på något sätt är likhet. Slackvariablerna som vi inför berättar för oss "hur långt" det är till respektive bivillkor, d.v.s. att när vi är i en hörnpunkt som ligger på bivillkor 1 (det kallas då ett "aktivt" bivillkor) så kommer $s_1 = 0$. Låt säga att vi befinner oss i origo. Då kommer samtliga "vanliga" variabler att vara noll. I det fallet kommer $s_1 = 12$ och $s_2 = 7$ för att uppnå likhet. Om vi befinner oss i det hörn där bivillkor 1 skärs utav bivillkor 2 så kommer $s_1 = 0$ och $s_2 = 0$.

Nu har vi alltså problemet på *standardform*, och A är en $m \times n$ -matris, där $n > m$ (d.v.s. att antalet variabler är större än antalet bivillkor). Härifrån kan vi välja en baslösning att starta i.

Baslösning

Förutsättningar: problemet är skrivet på standardform och A är en $m \times n$ -matris, där $n > m$. Vi kan då välja ut $(n - m)$ variabler som får värdet 0. Dessa kommer att kallas icke-basvariabler. I exemplet nedan ser vi att vi har 3 bivillkor (m) och 5 variabler (n). Således har vi att $(n - m) = (5 - 3) = 2$ variabler ska sättas till 0 för att utgöra icke-basvariabler. Oftast är det enklast att börja med att "nolla" de vanliga variablerna (d.v.s. att låta slackvariablerna utgöra *baslösningen* och därmed börja i origo):

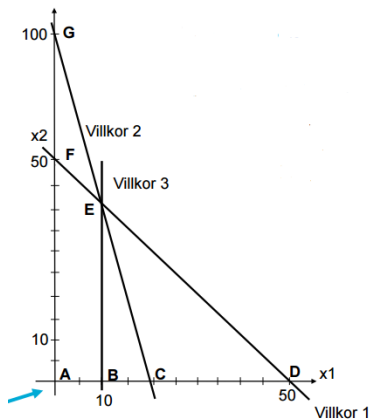
$$\begin{aligned} 2x_1 + 2x_2 + s_1 &= 100 \\ 6x_1 + x_2 + s_2 &= 100 \\ 10x_1 + s_3 &= 100 \end{aligned} \Rightarrow \text{Låter } x_1 = x_2 = 0$$

$$\begin{aligned} 2x_1 + 2x_2 + s_1 &= 100 \\ 6x_1 + x_2 + s_2 &= 100 \\ 10x_1 + s_3 &= 100 \end{aligned}$$

Detta ger oss:

Icke-basvariabler: $x_1 = x_2 = 0$

Basvariabler: $s_1, s_2, s_3 \Rightarrow s_1 = s_2 = s_3 = 100 \Rightarrow$ Tillåten baslösning, ty alla variabler ≥ 0
 $\Rightarrow x = (0 \ 0 \ 100 \ 100 \ 100)^T \Rightarrow$ origo (A)



Detta illustreras i bilden, där vi alltså befinner oss i punkt A. I tabellen nedan kan vi se vart andra baslösningar hade tagit oss.

Figur 7: Grafisk illustration av exemplet. Från FÖ3.

Icke-basvariabler	Basvariabler	Lösning	Hörnpunkt
$x_1, x_2 = 0$	s_1, s_2, s_3	$x=(0, 0, 100, 100, 100)^T$	A
$x_1, s_1 = 0$	x_2, s_2, s_3	$x=(0, 50, 0, 50, 100)^T$	F
$x_1, s_2 = 0$	x_2, s_1, s_3	$x=(0, 100, -100, 0, 100)^T$	G
$x_1, s_3 = 0$	x_2, s_1, s_2	-	-
$x_2, s_1 = 0$	x_1, s_2, s_3	$x=(50, 0, 0, -200, -400)^T$	D
$x_2, s_2 = 0$	x_1, s_1, s_3	$x=(100/6, 0, 200/3, 0, -200/3)^T$	C
$x_2, s_3 = 0$	x_1, s_1, s_2	$x=(10, 0, 80, 40, 0)^T$	B
$s_1, s_2 = 0$	x_1, x_2, s_3	$x=(10, 40, 0, 0, 0)^T$	E
$s_1, s_3 = 0$	x_1, x_2, s_2	$x=(10, 40, 0, 0, 0)^T$	E
$s_2, s_3 = 0$	x_1, x_2, s_1	$x=(10, 40, 0, 0, 0)^T$	E

Tabell 1: Vart andra baslösningar hade tagit oss. Jämför gärna med figuren ovan. Från FÖ3.

Kommentar: Om en baslösning uppfyller alla icke-negativitetsvillkor kallas den för en *tillåten baslösning*. I annat fall har vi en *otillåten baslösning*. Varje tillåten baslösning motsvaras av en entydig hörnpunkt i det tillåtna området (p.g.a. att *minst* $(n - m)$ variabler = 0), men en hörnpunkt kan åt det andra hållet representeras av flera tillåtna hörnpunkter. I vårt exempel ovan händer detta i hörnpunkten E, därför att samtliga slackvariabler i den punkten = 0, men bara två utav dem = 0 p.g.a. att de är "icke-basvariabler" (se tabellen). En sådan lösning, där en (eller flera) *basvariabler* har värdet noll, kallas för en *degenererad lösning*.

Överkurs: Eftersom det bara finns ett ändligt antal baslösningar (närmare bestämt: $\binom{n}{m} = \frac{n!}{m!(n-m)!}$) och det för varje baslösning finns unika värden på basvariablerna (och därmed även målfunktionen) så kommer målfunktionsvärdet strikt förbättras för varje iteration. Det betyder också, p.g.a. att problemen är linjära, att simplexmetoden aldrig återkommer till en tidigare genererad baslösning. Alltså kommer det krävas ett ändligt antal iterationer innan optimum hittas. Simplexmetoden kommer således att lösa alla LP-problem, under förutsättning att målfunktionsvärdet strikt förbättras i varje iteration. Här uppkommer bekymret med degenererade baslösningar, då simplexmetoden kan "fastna" (det blir ungefär som en "loop"). I en degenererad baslösning kan den reducerade kostanden indikera en förbättring i någon riktning, men steglängden i den valda sökriktningen blir noll. Vi byter således baslösning, men står kvar i samma hörnpunkt (och lösning). Det finns sätt att kringgå detta, men när man gör det för hand så kan man ofta se att så är fallet – och då får man välja en annan väg.

Vidare kan vi konstatera att vår baslösning kommer att vara en $m \times m$ -matris, kallad **B**. Om $\det(\mathbf{B}) \neq 0$ så har vi en baslösning. I vårt exempel har vi att s_1, s_2, s_3 är basvariablerna, och därmed är det deras kolumner som utgör matrisen **B**. Vi ser att deras värden i respektive rad = 1 och att dessa kolumner utgör en enhetsmatris. När vi byter bas gäller det fortfarande att den nya basens kolumner ska utgöra en enhetsmatris (typ, för ordningen på kolumnerna är egentligen irrelevant), vilket vi åstadkommer med vanliga radoperationer. Genom dessa radoperationer (bytet av baslösning) förflyttar vi oss mellan hörnpunkter i det tillåtna området, och tanken är att vi går i den riktning som vårt målfunktionsvärde förbättras mest.

Simplexalgoritmens sökmetod

I simplexmetoden går vi från baslösning till baslösning, i den riktning som målfunktionsvärdet förbättras mest. För att avgöra detta använder vi oss av s.k. *reducerad kostnad*, vilken tydligt framgår i simplextablån (koefficienten framför icke-basvariabeln i översta raden, med omvänt tecken). Den *bästa* (beror på om det är ett maximi- eller minimiproblem) reducerade kostnaden blir den *sökriktning* vi väljer, d.v.s. att vi väljer att göra den icke-basvariabeln med bäst reducerad kostnad till en ny basvariabel.

Steglängden, vilket i tablåform motsvaras av vilken basvariabel vi ska byta bort till förmån för den inkommande variabeln, avgörs av deras begränsning. Vi måste ju fortfarande hålla oss inom det tillåtna området, så vi kollar vilken variabel som begränsar oss mest.

Exempel: Simplexalgoritmen

Obs! För enkelhetens skull har problemet redan skrivits om till standardform. Vi kommer dock att konvertera det till tablåform.

$$\begin{array}{rcll} \max z & = & 240x_1 + 60x_2 & \\ \text{då} & & 2x_1 + 2x_2 + s_1 & = 100 \\ & & 6x_1 + x_2 + s_2 & = 100 \\ & & 10x_1 & + s_3 = 100 \\ & & x_1, x_2, s_1, s_2, s_3 & \geq 0 \end{array}$$

Exemplet är här skrivet på standardform. För en enklare lösningsgång skriver vi om det till tablåform.

När vi skriver det på tablåform så ändrar vi i den översta raden, så att vi får en likhet som blir noll.

Figur 8: Standardform. Från FÖ4.

Simplextablå:

bas	z	x1	x2	s1	s2	s3	b
z	1	-240	-60	0	0	0	0
s1	0	2	2	1	0	0	100
s2	0	6	1	0	1	0	100
s3	0	10	0	0	0	1	100

I tablån ser vi den *reducerade kostnaden* för respektive icke-basvariabel markerat med gul överstrykningspenna. Det är alltså omvänt tecken, så x_1 är ger alltså bäst effekt (blå färg) och därmed vill vi byta in den (x_1 blir vår "sökriktning").

Figur 9: Motsvarande simplextablå.

Härnäst måste vi avgöra vilken basvariabel (inringade med röd färg) som ska bytas ut ("steglängd"). Detta gör vi genom att kolla vilken basvariabel som begränsar oss först. Till vår hjälp har vi x_1 :s koefficienter (inringade i lila). Notera här att x_2 fortfarande är en icke-basvariabel, och således är satt till 0, vilket gör att vi inte behöver ta hänsyn till dess koefficienter. Notera också att den basvariabel vi vill byta ut ska bli satt till noll, eftersom den ska bli en icke-basvariabel. Kom ihåg att när en slackvariabel = 0 så är vi på ett villkor. Således har vi att *det villkor som aktiveras först*, d.v.s. den slackvariabel som blir noll vid minsta möjliga x_1 kommer att vara den vi vill byta ut! Om vi tolkar tablån får vi då att:

$$\begin{aligned}
 2x_1 + s_1 &= 100 & 100 - 2x_1 &= 0 & x_1 &= 50 \\
 6x_1 + s_2 &= 100 & \Rightarrow /bas_i = 0/ \Rightarrow & 100 - 6x_1 &= 0 & \Leftrightarrow x_1 &= \frac{100}{6} \\
 10x_1 + s_3 &= 100 & & 100 - 10x_1 &= 0 & x_1 &= 10
 \end{aligned}$$

Alltså har vi att bivillkor 3 aktiveras först ($\min_{x_1} (50 \quad \frac{100}{6} \quad 10) = 10$), så s_3 blir vår utgående variabel. För att uppdatera tablån ska alltså kolumnen för x_1 bli $(0 \ 0 \ 0 \ 1)^T$. För att åstadkomma detta använder vi nuvarande *rad 3* (som tillhör s_3 men ska komma att tillhöra x_1) och gör radoperationer. Vi gör följande operationer (raderna är indexerade från 0 till 3):

$$\begin{aligned}
 (0) &= (0) + 24(3) \\
 (1) &= (1) - \frac{2}{10}(3) \\
 (2) &= (2) - \frac{6}{10}(3) \\
 (3) &= \frac{1}{10}(3)
 \end{aligned}$$

Därmed får vi den uppdaterade tablån:

bas	z	x1	x2	s1	s2	s3	b
z	1	0	-60	0	0	24	2400
s1	0	0	2	1	0	-0,2	80
s2	0	0	1	0	1	-0,6	40
x1	0	1	0	0	0	0,1	10

Här ser vi att målfunktionsvärdet har uppdaterats till 2400, vilket ju är en förbättring. Vi ser också att x_2 är den enda icke-basvariabeln med en förbättrande reducerad kostnad. Denna ska alltså bytas in. Vi kollar vilken basvariabel som ska substitueras ut ur den nuvarande baslösningen:

Figur 10: Tablån efter en iteration.

$$\begin{aligned}
 2x_2 + s_1 &= 80 & 80 - 2x_2 &= 0 & x_2 &= 40 \\
 x_2 + s_2 &= 40 & \Rightarrow /bas_i = 0/ \Rightarrow & 40 - x_2 &= 0 & \Leftrightarrow x_2 &= 40 \\
 10x_1 &= 10 & & \text{Beror ej av } x_2 & & -
 \end{aligned}$$

Här ser vi att det är likvärdigt att byta ut s_1 eller s_2 , så vi kan välja fritt. Här väljer vi att byta ut s_1 . x_2 :s kolumn ska nu bli $(0 \ 1 \ 0 \ 0)^T$, så vi gör radoperationer med utgångspunkt i rad (1):

$$(0) = (0) + 30(1)$$

$$(2) = (2) - \frac{1}{2}(1)$$

$$(3) = (3) \quad , \text{ vilket ger:}$$

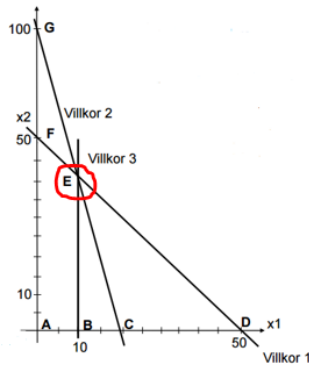
$$(1) = \frac{1}{2}(1)$$

bas	z	x1	x2	s1	s2	s3	b
z	1	0	0	30	0	24	4800
x2	0	0	1	0,5	0	-0,2	40
s2	0	0	0	-0,5	1	-0,6	0
x1	0	1	0	0	0	0,1	10

Figur 11: Tablån efter två iterationer.

Här ser vi att det inte längre finns någon reducerad kostnad som ger en förbättring, och på så sätt vet vi att vi har funnit optimum. Hur tolkar vi då den slutgiltiga tablån?

- Det gulmarkerade värdet är det maximala målfunktionsvärdet ($z^* = 4800$).
- Koordinaterna för detta blir $x^* = (10 \ 40)^T$, vilket utläses i högerledet för respektive basvariabels rad. Notera att slackvariablerna, oavsett deras värden, inte hör till koordinaterna man svarar i. Om en x -variabel inte är med i baslösningen så är den som bekant satt till noll.



Figur 12: Grafisk framställning av optimallösningen. Från FÖ4.

- Att s_2 är en basvariabel, men med värdet 0, innebär att bivillkor 2 är aktivt. Eftersom s_1 och s_3 är icke-basvariabler är de också satta till 0. Detta innebär att bivillkor 1 och 3 också är aktiva. Notera också att detta innebär att vi har en degenererad lösning.

Grafiskt kan vi se att lösningen motsvaras av hörnpunkten E, vilket vi också skulle ha märkt om vi löste problemet grafiskt med hjälp av målfunktionens gradient $\nabla f = \begin{pmatrix} 240 \\ 60 \end{pmatrix} = 60 \begin{pmatrix} 4 \\ 1 \end{pmatrix}$.

Tittar man på de använda simplextablåerna (titta efter vilka bivillkor som varit aktiva, = 0) kan man se att vi har gått ifrån punkt A (origo) till punkt B ($s_3 = 0$) och slutligen till punkt E ($s_1 = s_2 = s_3 = 0$).

Algebraisk version av simplexmetoden

Simplexmetoden fungerar även med ren algebraisk matrissräkning, men detta går jag inte in på här. För mer info kring detta, se powerpointen från föreläsning 4 och/eller ss. 97-99 i läroboken).

Fas I

Den del av simplexmetoden som beskrevs ovan kallas vanligtvis *fas II* och är den man går direkt till om man har en enhetsmatris att starta i (t.ex. om origo är en tillåten lösning). Det kan dock hända att man först måste "hitta till det tillåtna området", varpå man börjar med *fas I*. Om man inte har en enhetsmatris att starta i så måste man lägga till de extra kolumner som saknas och sedan försöka pivotera bort dessa (och på så sätt hitta till området). Därefter kan man lösa problemet med *fas II*.

De variabler man lägger till för att få fler kolumner kallas för *artificiella variabler*. I fas 1 löser man problemet min w istället för z , vilket syns nedan:

Ursprungligt problem

$$\begin{aligned} \min \quad & 3x_1 + x_2 + 2x_3 \\ \text{då} \quad & 2x_1 + 3x_2 + x_3 \geq 5 \\ & 2x_1 + x_2 + 2x_3 = 6 \\ & x_1, x_2, x_3 \geq 0 \end{aligned}$$

Vilket skrivs om:

På standardform

$$\begin{aligned} \min \quad & 3x_1 + x_2 + 2x_3 \\ \text{då} \quad & 2x_1 + 3x_2 + x_3 - s_1 = 5 \\ & 2x_1 + x_2 + 2x_3 = 6 \\ & x_1, x_2, x_3, s_1 \geq 0 \end{aligned}$$

Figur 14: Fas I-exempel. Från FÖ4.

Figur 13: Fas I-exempel (standardform). Från FÖ4.

Detta konverteras sedan till en simplextablå, men denna kommer att sakna en enhetsmatrix:

Tablå

bv	z	x_1	x_2	x_3	s_1	\bar{b}	
z	1	-3	-1	-2	0	0	(0)
?	0	2	3	1	-1	5	(1)
?	0	2	1	2	0	6	(2)

Som synes finns det för närvarande ingen enhetsmatrix, och således ingen baslösning att börja i. Vi löser detta genom att köra fas I och införa två artificiella variabler. Notera att vi då får problemet min w istället. Vi skriver alltså om bivillkoren först och använder att $w = a_1 + a_2$ för att få fram fas I-målfunktionen:

Figur 15: Fas I-exempel (tablå). Från FÖ4.

Lägger till två artificiella variabler och ta fram den nya målfunktionen:

$$\begin{aligned} 2x_1 + 3x_2 + x_3 - s_1 + a_1 &= 5 \quad (1) \\ 2x_1 + x_2 + 2x_3 &+ a_2 = 6 \quad (2) \end{aligned}$$

$$\Leftrightarrow / \min w = a_1 + a_2, \text{ så lös ut } a_1 \text{ \& } a_2 / \Leftrightarrow \begin{aligned} a_1 &= 5 - 2x_1 - 3x_2 - x_3 + s_1 \\ a_2 &= 6 - 2x_1 - x_2 - 2x_3 \end{aligned} \Rightarrow$$

$$w = a_1 + a_2 = 5 - 2x_1 - 3x_2 - x_3 + s_1 + 6 - 2x_1 - x_2 - 2x_3 = 11 - 4x_1 - 4x_2 - 3x_3 + s_1$$

Vi inför nu detta i en simplextablå m.a.p. w :

Starttablå, fas I (min w):

bv	w	x_1	x_2	x_3	s_1	a_1	a_2	\bar{b}	
!	w	1	4	4	3	-1	0	0	11 (0)
	a_1	0	2	3	1	-1	1	0	5 (1)
	a_2	0	2	1	2	0	0	1	6 (2)

Figur 16: Fas I-exempel (fas I-tablå). Från FÖ4.

Nu har vi en tablå som vi faktiskt kan räkna med. De artificiella variablerna kan på ett sätt tolkas som avståndet till det tillåtna området, ungefär på samma sätt som slackvariablerna anger avståndet till bivillkoren. Vi vill således *minimera* w , för när de artificiella variablerna tillhör icke-

basvariablerna (och är lika med noll, vilket är deras minsta möjliga värde p.g.a. teckenkrav) så har vi hittat det tillåtna området och har en tillåten baslösning i de vanliga fas II-variablerna. Då kan vi stryka de artificiella och räkna vidare. Notera att vi fortfarande har "omvänt tecken" i målfunktionsraden, så i detta minimiproblem väljer vi den icke-basvariabel som har den högsta (positiva) reducerade kostnaden. I det här fallet blir det x_1 eller x_2 . Vi väljer x_1 och räknar, precis som i fas II, ut vilken basvariabel som begränsar oss först:

$$\begin{aligned} 2x_1 + a_1 = 5 \\ 2x_1 + a_2 = 6 \end{aligned} \Rightarrow /bas_i = 0 / \Rightarrow \begin{aligned} a_1 = 5 - 2x_1 \\ a_2 = 6 - 2x_1 \end{aligned} \Leftrightarrow \begin{aligned} x_1 = \frac{5}{2} \\ x_1 = 3 \end{aligned} \Rightarrow a_1 \text{ ska ut!}$$

Vi byter baslösning genom radoperationer, precis som i fas II:

$$\begin{aligned} (1) &= \frac{1}{2}(1) \\ (2) &= (2) - (1) \quad \text{vilket ger:} \\ (0) &= (0) - 2(1) \end{aligned}$$

Tablå, fas I, efter en iteration:

bv	w	x_1	x_2	x_3	s_1	a_1	a_2	\bar{b}	
w	1	0	-2	1	1	-2	0	1	(0)
x_1	0	1	3/2	1/2	-1/2	1/2	0	5/2	(1)
a_2	0	0	-2	1	1	-1	1	1	(2)

Figur 17: Fas I-exempel (1 iteration). Från FÖ4.

Ny inkommande variabel blir x_3 (eller s_1), då dessa har bäst reducerad kostnad. För att se vilken basvariabel som ska bytas ut kollar vi vart vi begränsas först:

$$\begin{aligned} \frac{1}{2}x_3 + x_1 = \frac{5}{2} \\ x_3 + a_2 = 1 \end{aligned} \Rightarrow /bas_i = 0 / \Rightarrow \begin{aligned} x_1 = \frac{5}{2} - \frac{1}{2}x_3 \\ a_2 = 1 - x_3 \end{aligned} \Leftrightarrow \begin{aligned} x_3 = 5 \\ x_3 = 1 \end{aligned} \Rightarrow a_2 \text{ ska ut!}$$

Vi gör följande radoperationer för att åstadkomma basbytet:

$$\begin{aligned} (2) &= (2) \\ (1) &= (1) - \frac{1}{2}(2) \quad \text{vilket ger:} \\ (0) &= (0) - (2) \end{aligned}$$

Tablå, fas I, efter två iterationer:

bv	w	x_1	x_2	x_3	s_1	a_1	a_2	\bar{b}	
w	1	0	0	0	0	-1	-1	0	(0)
x_1	0	1	5/2	0	-1	1	-1/2	2	(1)
x_3	0	0	-2	1	1	-1	1	1	(2)

Här ser vi att vi har hittat optimum för fas I, då vi inte längre har någon fördelaktig

Figur 18: Fas I-exempel (2 iterationer). Från FÖ4.

reducerad kostnad att basera ett basbyte på. Enhetsmatrisen hör ni dessutom till de "ursprungliga variablerna", vilket var vad vi var ute efter. De artificiella variablerna är nu icke-basvariabler och därmed lika med noll, varpå vi kan ta bort dem. Detta görs efter samma princip som när vi bytte till w från början. Vi utgår från den ursprungliga målfunktionen:

$$z = 3x_1 + x_2 + 2x_3$$

Nu vill vi uttrycka z i *enbart icke-basvariabler*. Från tablå ser vi att x_1 och x_3 är basvariabler och dessa kan skrivas om med hjälp av deras respektive rader i tablå:

$$\begin{aligned} x_1 &= 2 - \frac{5}{2}x_2 + s_1 \\ x_3 &= 1 + 2x_2 - s_1 \end{aligned}$$

Slår vi ihop detta får vi att:

$$\Rightarrow z = 3x_1 + x_2 + 2x_3 = 3\left(2 - \frac{5}{2}x_2 + s_1\right) + x_2 + 2(1 + 2x_2 - s_1) = \dots = 8 - \frac{5}{2}x_2 + s_1$$

Sorterar i över detta till den vanliga tablåformen så får vi följande:

Starttablå, fas II (min z):

bv	z	x_1	x_2	x_3	s_1	\bar{b}	
!	z	1	0	5/2	0	-1	8 (0)
	x_1	0	1	5/2	0	-1	2 (1)
	x_3	0	0	-2	1	1	1 (2)

Härifrån löser vi det på samma sätt som vi gjorde tidigare, men det går jag inte igenom här.

Figur 19: Fas I-exempel (fas II). Från FÖ4.

För att sammanfatta fas I kan vi säga att de artificiella variablerna tillåter oss att starta vår simplexalgoritm i en otillåten punkt, så att vi åtminstone får en startbaslösning. Härifrån pivoterar vi för att "nolla" de artificiella variablerna (och därmed även w). När detta sker så har vi en tillåten lösning i det ursprungliga problemet, så vi formulerar om tablå och räknar vidare. Om w inte går att "nolla" så har det ursprungliga problemet ingen lösning.

Att tolka simplextablån

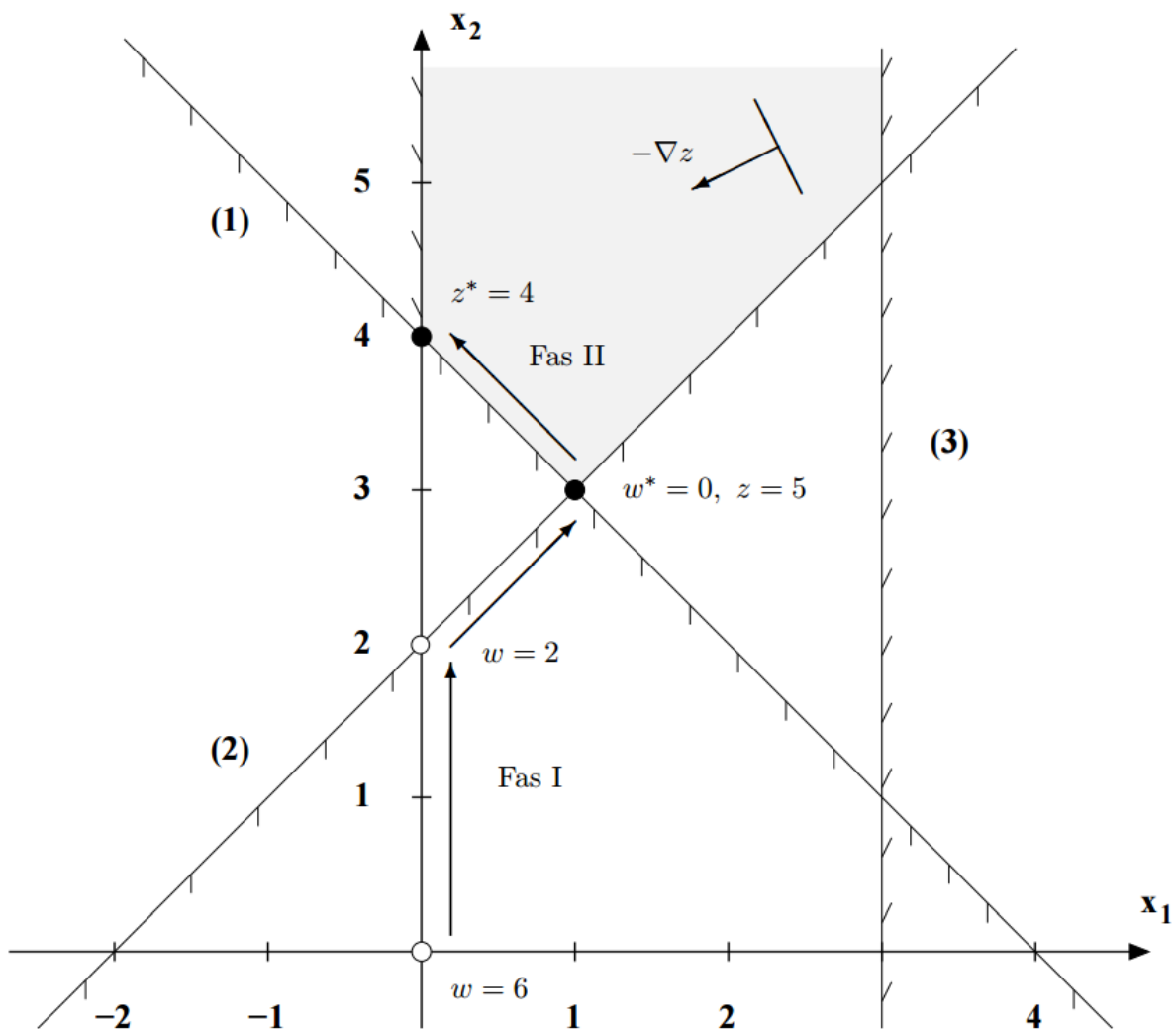
Ett antal saker kan utläsas ur simplextablån. Det här korta delavsnittet ger några exempel på vad.

- Reducerade kostnaden = 0
Om den reducerade kostnaden för en variabel är 0 så kommer man inte att tjäna något på att låta denna variabels värde öka.
- Obegränsad lösning
Om den inkommande variabeln har en kolumn som innebär att ingen basvariabels värde minskar när den inkommande variabelns värde ökar så finns det i den riktningen en obegränsad lösning. Man kommer alltså inte att slå i något bivillkor, hur långt man än går.
- Degenererad lösning
Om någon basvariabel har värdet noll så är lösningen degenererad. Detta är alltså en punkt där "för många" bivillkor skär varandra. Problematiken med detta diskuteras under rubriken *baslösning* (längst ned, under "överkurs") i början av detta avsnitt.

Sammanfattning

Sammanfattningsvis kan vi säga att simplexalgoritmen är en sökmetod som går från hörnpunkt till hörnpunkt (eftersom optimum i ett LP-problem finns i (minst) en hörnpunkt) i den riktning som målfunktionsvärdet ökar mest. Från ett hörn går man alltid till ett angränsande hörn. Sökningen anses färdig när det inte längre finns någon *närliggande* (granne) med ett bättre målfunktionsvärde. Sökningen kommer att konvergera mot optimum eftersom det finns ett ändligt antal hörnpunkter och vi kommer aldrig (förutsatt att vi inte fastnar i en degenererad lösning) att söka igenom samma baslösning två gånger.

Simplexalgoritmen kan delas in i "fas I" och "fas II", vilka i runda drag fungerar på samma sätt. Under fas I använder vi artificiella variabler för att leta upp det tillåtna området och under fas II letar vi upp optimum inom det tillåtna området. En bra bild som förklarar hur simplexalgoritmen (fas I och fas II) söker finns här nedan:



Figur 20: Ett exempel på hur simplexalgoritmen söker. Sökningen började i origo och gjorde två iterationer i fas I för att ta sig till området. Därifrån krävdes det en iteration i fas II för att hitta optimum. Från "utförliga exempel" på Lisam.

Dualitet

Ytterligare ett centralt begrepp inom optimeringsläran är dualitet. Till varje LP-problem kan man formulera ett *dualt problem* som är definierat med samma indata som det ursprungliga *primala problemet*. Det finns flera viktiga och intressanta kopplingar mellan det primala- och det duala problemet, vilket kommer till användning i flera sammanhang.

Utan att gå in för djupt på hur det faktiskt hänger ihop (en djupare härledning finns i läroboken, kapitel 6.1) så kan vi konstatera att om man tittar på ett problems bivillkor, så ger ju de en sorts "tak" för hur bra målfunktionsvärde vi kan få. Vi kan ha flera bivillkor, och alla behöver inte ens vara aktiva, men vi vet att optimallösningen *inte bryter mot någon av dem*. På så sätt vet vi också att varje bivillkor kan användas för att göra en *optimistisk skattning* utav problemet. För att få ett komplett problem ger man varje bivillkor i det primala problemet en "vikt", d.v.s. en variabel v_i i det duala problemet.

Det duala problemets målfunktion, kallad w , är alltså *alltid* en optimistisk skattning. Om vårt primala problem är ett maximiproblem så kommer således $w \geq z^* \geq z$. Utav detta följer det att det minsta tillåtna w kommer att ha samma värde som det optimala värdet på z . Således betraktar vi det duala problemet som ett minimiproblem när det primala är ett maximiproblem och konstaterar lite finurligt att $w^* = z^*$. Om det primala problemet är ett minimiproblem så följer det med motsvarande resonemang att det duala problemet är ett maximiproblem, men det gäller fortfarande att $w^* = z^*$.

I det här avsnittet kikar vi på hur man tar fram det duala problemet, vad man kan använda det till och en del annat.

Att ta fram det duala problemet

Vi börjar med att kika på det rent algebraiskt:

(P) $\max z = \mathbf{c}^T \mathbf{x}$ Det primala problemet skrivs på matrisform som t.v., där c representerar
 då $\mathbf{Ax} \leq \mathbf{b}$ koefficienterna framför variablerna i målfunktionen, A är koefficienterna i
 $\mathbf{x} \geq \mathbf{0}$ bivillkoren och b är högerledet.

Figur 21: Primalt problem.
Från FÖ5.

(D) $\min w = \mathbf{b}^T \mathbf{v}$ Det duala problemet använder ju som sagt samma indata och som det syns i
 då $\mathbf{A}^T \mathbf{v} \geq \mathbf{c}$ figuren till vänster så är det i grund och botten en algebraisk rockad (och
 $\mathbf{v} \geq \mathbf{0}$ viss transponering) som skiljer dem åt. Notera också att primalen är ett
 maximiproblem, medan dualen är ett minimiproblem. Dessutom är
 bivillkoren lite annorlunda, då \leq har blivit \geq i det duala problemet. Detta
 återkommer vi till.

Figur 22: Dualt problem.
Från FÖ5.

Om man ska ta fram dualen i ett mer detaljerat exempel, med utskrivna variabler och koefficienter så går man tillväga som följer. Betrakta följande problem:

(P) $\min z = 3x_1 - x_2 + 2x_3 + x_4$
 då $4x_1 + 2x_2 + 2x_3 - x_4 \geq 14 \quad |v_1$
 $2x_1 + 2x_2 + x_3 + x_4 \leq 10 \quad |v_2$
 $x_1 + 3x_3 \leq 8 \quad |v_3$
 $x_1, x_2, x_3, x_4 \geq 0$

Figur 23: Exempel. Från FÖ5.

Vi kan lägga märke till ett antal saker:

- Det primala problemet är ett minimiproblem, vilket gör att det duala problemet ska vara ett *maximiproblem*.
- Det är varierande tecken hos bivillkoren. Detta är i sig inget problem, men dessa tecken är kopplade till teckenkravet på de variabler v som de är kopplade till. Tumregeln är att *standardform ger standardform*. Standardform för ett minimiproblem är \geq -villkor, så endast v_1 kommer att vara positiv i det duala problemet. Skulle ett bivillkor kräva likhet så kommer motsvarande dualvariabel att vara "fri".

- Vi ser också att samtliga variabler x_i är icke-negativa. Detta innebär, med samma resonemang som ovan, att samtliga bivillkor i det duala problemet kommer att vara på standardform. För vårt problem innebär det att alla kommer att vara \leq -villkor (ty det duala problemet är i vårt fall ett maximiproblem).

Vill vi koefficienterna på matrisform så har vi att:

$$c^T = (3 \quad -1 \quad 2 \quad 1), A = \begin{pmatrix} 4 & 2 & 2 & -1 \\ 2 & 2 & 1 & 1 \\ 1 & 0 & 3 & 0 \end{pmatrix}, b = \begin{pmatrix} 14 \\ 10 \\ 8 \end{pmatrix}$$

Dessa kan transponeras och sättas direkt in i de algebraiska formlerna på samma sätt. Vi får i vilket fall att det duala problemet blir:

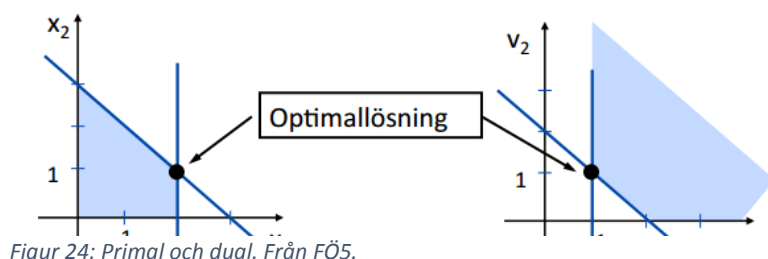
$$\begin{aligned} \text{(D) } \max \quad w &= 14v_1 + 10v_2 + 8v_3 \\ \text{då} \quad 4v_1 + 2v_2 + v_3 &\leq 3 \\ 2v_1 + 2v_2 &\leq -1 \\ 2v_1 + v_2 + 3v_3 &\leq 2 \\ -v_1 + v_2 &\leq 1 \\ v_1 \geq 0, \quad v_2, v_3 &\leq 0 \end{aligned}$$

Vi ser alltså att vår vektor b har transponerats och nu utgör koefficienterna i den duala målfunktionen, att A bara har transponerats och att c har blivit högerledet. Vi ser också att teckenkraven blev som vi konstaterade tidigare, både för bivillkoren och variablerna v .

Vidare ska vi kolla vad vi kan använda detta till.

Relationen mellan primal och dual

Ett sätt att grafiskt illustrera relationen dem emellan är med följande bild:

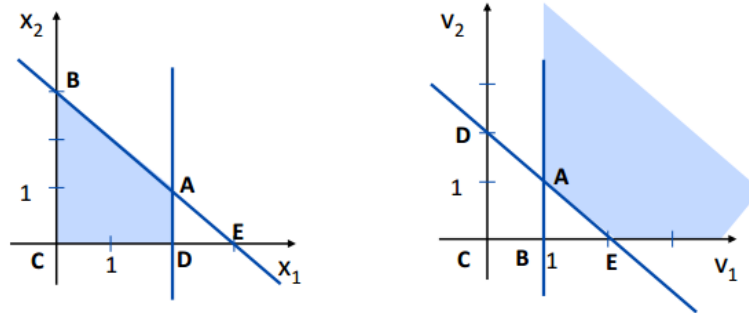


Figur 24: Primal och dual. Från FÖ5.

Vi ser här att det primala problemet är ett maximiproblem, medan det duala ska minimeras. Vi ser också att koordinaterna i de olika systemen inte är desamma, men när dessa "översätts" så kommer de att vara det.

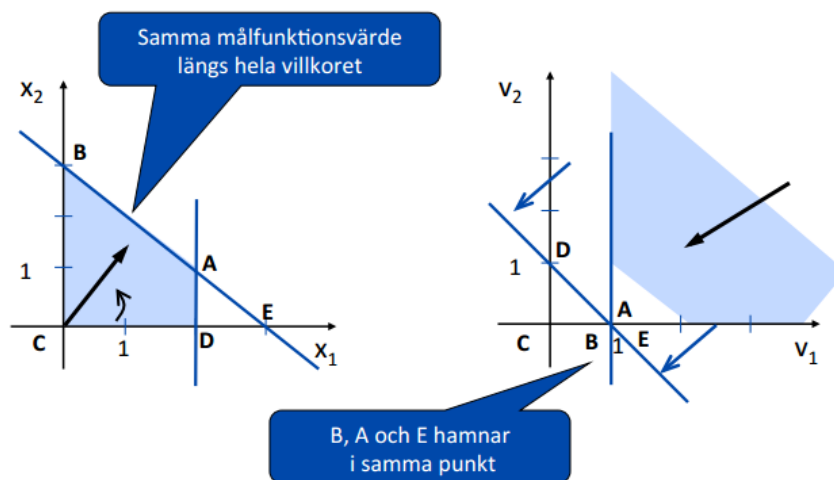
Vi ser också att en tillåten lösning i primalen kommer att ha ett lägre målfunktionsvärde än en tillåten lösning i det duala problemet. Vi har alltså att $c^T \hat{x} \leq b^T \hat{v}$ (se "svaga dualsatsen"). Likheter kommer att uppstå då $c^T \hat{x}$ är så stort som möjligt och $b^T \hat{v}$ är så litet som möjligt, d.v.s. att $z^* = w^*$. Detta är alltså det enda gemensamma målfunktionsvärdet och det ter sig dessutom så att om man översätter x^* så får man v^* . Detta är den enda tillåtna punkten i X som, när den översätts, kommer att vara tillåten i V . Detta sammanfaller med de s.k. komplementsvillkoren. Det är nämligen så att om \hat{x} och \hat{v} är optimallösningar i respektive problem så måste det vara så att:

$\hat{v}^T (b - A\hat{x}) = 0$ och $\hat{x}^T (A^T \hat{v} - c) = 0$. Detta hör ihop med att varje extrempunkt i primalen har en motsvarande extrempunkt i dualen, men att bara en extrempunkt (optimum) är "gemensam", vilket illustreras i figuren:



Figur 25: Koppling mellan extrempunkter i primalen och dualen. Endast optimum är tillåten i båda problemen. Från FÖ5.

En annan intressant egenskap som komplementariteten medför är att om gradienten för den primala målfunktionen är parallell med ett bivillkor, låt säga det bivillkoret som innehåller punkterna B, A och E i det primala problemet ovan, så kommer motsvarande punkter i det duala problemet att hamna i samma koordinat i det duala problemet:



Figur 26: Från FÖ5.

Summerar vi detta så ser vi att vi har ett antal optimalitetsvillkor som vi kan använda för att bekräfta att vi faktiskt har hittat optimum:

- Primal tillåtenhet: $Ax \leq b, x \geq 0$
- Dual tillåtenhet: $A^t v \geq c, v \geq 0$
- Komplementaritet: $v^T(Ax - b) = 0$
 $x^T(A^t v - c) = 0$

Exempel

Låt säga att vi har samma exempel som tidigare (så dualen är redan uträknad och klar). Vi vill undersöka huruvida punkten $x = (2 \ 3 \ 0 \ 0)^T$ är optimal. Vi kollar således primal tillåtenhet (är x tillåten?), komplementaritet (vilket v hör till x ?) och dual tillåtenhet (är v tillåten?).

Primalen och dualen är sedan tidigare:

$$\begin{array}{ll}
 \text{(P)} \quad \min z = 3x_1 - x_2 + 2x_3 + x_4 & \text{(D)} \quad \max w = 14v_1 + 10v_2 + 8v_3 \\
 \text{då} \quad 4x_1 + 2x_2 + 2x_3 - x_4 \geq 14 & \text{då} \quad 4v_1 + 2v_2 + v_3 \leq 3 \\
 \quad 2x_1 + 2x_2 + x_3 + x_4 \leq 10 & \quad 2v_1 + 2v_2 \leq -1 \\
 \quad x_1 + 3x_3 \leq 8 & \quad 2v_1 + v_2 + 3v_3 \leq 2 \\
 \quad x_1, x_2, x_3, x_4 \geq 0 & \quad -v_1 + v_2 \leq 1 \\
 & \quad v_1 \geq 0, v_2, v_3 \leq 0
 \end{array}$$

Figur 28: Primalen. Från FÖ5.

Figur 27: Dualen. Från FÖ5.

Vi börjar alltså med att kolla primal tillåtenhet. Vi börjar med att konstatera att $x_1, x_2, x_3, x_4 \geq 0$. Sedan kollar vi att x -värdena uppfyller primalens bivillkor.

$$\begin{aligned}
 4 \times 2 + 2 \times 3 + 0 - 0 &= 14 \geq 14 \text{ ok!} \\
 2 \times 2 + 2 \times 3 + 0 + 0 &= 10 \leq 10 \text{ ok!} \\
 1 \times 2 + 0 + 0 + 0 &= 2 \leq 8 \text{ ok!}
 \end{aligned}$$

Därefter kollar vi komplementaritetsvillkoren:

$$\begin{aligned}
 v_1(4x_1 + 2x_2 + 2x_3 - x_4 - 14) &= 0 & v_1(14 - 14) &= 0 & v_1 &=? \\
 v_2(2x_1 + 2x_2 + x_3 + x_4 - 10) &= 0 & \Leftrightarrow v_2(10 - 10) &= 0 & \Rightarrow v_2 &=? \\
 v_3(x_1 + 0x_2 + 3x_3 - 0x_4 - 8) &= 0 & v_3(2 - 8) &= 0 & v_3 &= 0
 \end{aligned}$$

Som vi ser får vi i detta fall bara information om v_3 p.g.a. att det är likhet i primalens första och andra bivillkor. Därmed går vi vidare och kollar på det åt andra hållet.

$$\begin{aligned}
 x_1(4v_1 + 2v_2 + v_3 - 3) &= 0 & 4v_1 + 2v_2 &= 3 \\
 x_2(2v_1 + 2v_2 + 0v_3 + 1) &= 0 & \Leftrightarrow /x_3, x_4, v_3 = 0 / \Leftrightarrow 2v_1 + 2v_2 &= -1 \Rightarrow \\
 x_3(2v_1 + v_2 + 2v_3 - 2) &= 0 & &?? \\
 x_4(-v_1 + v_2 + 0v_3 - 1) &= 0 & &?? \\
 4v_1 + 2v_2 = 3 & v_1 = 2 \\
 2v_1 + 2v_2 = -1 & \Leftrightarrow v_2 = -\frac{5}{2} \Rightarrow v = \begin{pmatrix} 2 \\ -\frac{5}{2} \\ 0 \end{pmatrix} \\
 v_3 &= 0
 \end{aligned}$$

Nu har vi fått fram den motsvarande punkten i det duala problemet. Då fattas det bara att undersöka den duala tillåtenheten. Vi konstaterar att teckenkraven är uppfyllda och kollar därefter om bivillkoren stämmer:

$$\begin{aligned}
 4 \times 2 + 2 \times \left(-\frac{5}{2}\right) + 0 &= 3 \leq 3 \text{ ok!} \\
 2 \times 2 + 2 \times \left(-\frac{5}{2}\right) + 0 &= -1 \leq -1 \text{ ok!} \\
 4 \times 2 + 1 \times \left(-\frac{5}{2}\right) + 0 &= \frac{3}{2} \leq 2 \text{ ok!} \\
 (-1) \times 2 + 1 \times \left(-\frac{5}{2}\right) + 0 &= -\frac{9}{2} \leq 1 \text{ ok!}
 \end{aligned}$$

Vi har alltså att samtliga villkor är uppfyllda och kan därför konstatera att punkten $x = (2 \ 3 \ 0 \ 0)^T$ är optimal.

Känslighetsanalys

I känslighetsanalysen undersöker vi vad som händer med målfunktionsvärdet när olika förutsättningar ändras.

Relaxation och Restriktion

Först måste man känna till två viktiga begrepp: relaxation och restriktion. Dessa finns beskrivna i definitionsavsnittet, men om man snabbt ska beskriva det så är en relaxation en förändring av problemet som innebär att det tillåtna området blir större och en restriktion är en förändring som gör området mindre. Storleken på området avgör ju hur många lösningar vi kan hitta, så att förstora området *kan inte göra optimallösningen sämre*. Får vi fler lösningar att välja på, men samtliga är sämre, så är ju vårt befintliga optimum fortfarande kvar. Samma motivering fungerar omvänt för en restriktion.

Skuggpriset och dess intervall

Skuggpris är ett annat centralt begrepp som man bör känna till. Detta finns väl beskrivet i definitionsavsnittet, men lite snabbt kan man säga att *skuggpriset för ett villkor ges av förändringen av det ursprungliga målfunktionsvärdet vid en enhetsökning av högerledet*. Skuggpriset för ett bivillkor går att utläsa i simplextablån. Det är det koefficientvärde som hör till bivillkorets slackvariabel, d.v.s. slackvariablernas reducerade kostnad. Det är dock viktigt att komma ihåg att skuggpriset bara gäller inom vissa intervall. Intervallet beror på *när ett annat bivillkor "tar över"* eller blir mer restriktivt än det befintliga bivillkoret, d.v.s. att det utanför intervallet inte spelar någon roll hur vi flyttar bivillkoret, eftersom vi då är utanför det tillåtna området till följd utav andra bivillkor.

Skuggpriset kan också, om man har räknat med ett dualt problem, ges av det optimala värdet på den dualvariabel som är kopplad till respektive bivillkor.

Intervallet som nämndes ovan kan utläsas i en datorutskrift som "_con.up" och "_con.down", men kan också räknas fram algebraiskt:

Intervallet är giltigt *så länge vi befinner oss i samma baslösning*, d.v.s. att när vi behöver byta baslösning så är skuggpriset inte längre giltigt. Kravet för att en baslösning ska vara tillåten är:

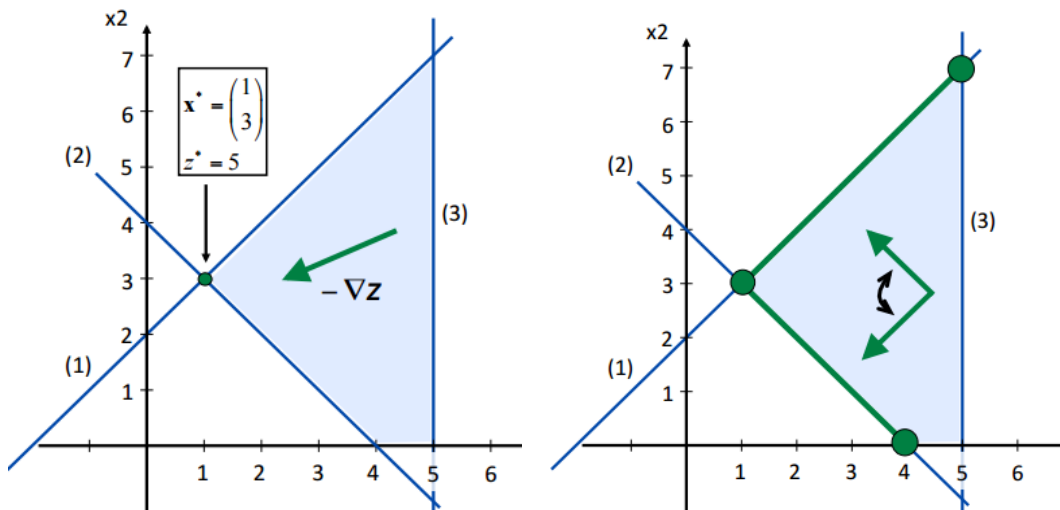
$$x_B = B^{-1}b \geq 0$$

Det giltiga intervallet för en komponent i **b**-vektorn (högerledet) ges således av att ovanstående olikhet ska vara uppfylld.

Förändring av en målfunktionskoefficient

En annan intressant del av känslighetsanalysen är en förändring av en målfunktionskoefficient. Hur kommer målfunktionsvärdet att reagera på det? Hur mycket kan vi förändra?

För detta gäller i grund och botten samma regel som för skuggpriset – vi har ett fall när vi håller oss inom samma baslösning och ett annat när vi går utanför det. Intervallet beror här på gradienten, eftersom den förändras när vi ändrar i målfunktionskoefficienten c_j . Låt säga att optimum ligger i en hörnpunkt som skapas av att två bivillkor skär varandra. Då kan vi "vrida" på målfunktionsgradienten (genom att ändra koefficienterna) på ett sådant sätt att den i ena ytterligheten blir parallell med det ena bivillkorets gradient och i den andra ytterligheten blir parallell med det andra bivillkorets gradient. Bilderna nedan beskriver det ganska bra:



Figur 29: Vänster: gradienten. Höger: Hur mycket gradienten kan vridas och fortfarande vara kvar i samma baslösning.

Förutsatt att vi är kvar i samma baslösning kan vi räkna på som vanligt, d.v.s. att vi kan ta vårt befintliga optimalvärde på variabeln som är berörd och multiplicera med förändringen i koefficienten Δc .

Utanför intervallet fungerar det på samma sätt som för skuggpriset. Behöver vi byta bas så kommer vi inte att få en större förändring än vad vi har inom intervallet och den kommer heller inte vara mindre än 0. Således får vi ett intervall som svar.

Beräkningarna går också att göra algebraiskt, vilket finns beskrivet i powerpointen för föreläsning 6. Utgångspunkten är det algebraiska kravet för optimalitet.

Införandet av en ny variabel

Inför man en ny variabel så vill man inte behöva lösa om hela problemet från början. Istället kan man beräkna dess reducerade kostnad (förutsatt att man fått alla koefficienter). Om man har en *gynnsam* reducerad kostnad så vill man pivotera in variabeln i sin simplexlösning. I annat fall kan man direkt förkasta den (om den reducerade kostnaden inte är gynnsam så vill vi "nolla" variabeln ändå).

För att få fram den reducerade kostnaden för den nya variabeln kan man använda följande formel:

$$\hat{c}_{ny} = c_{ny} - c_B^T B^{-1} a_{ny}$$

Observera att $c_B^T B^{-1}$ kan plockas direkt från simplextablån, men att man kan komma behöva ändra tecken på siffrorna innanför beroende på om den tillhörande slackvariabeln införts som "minus" eller "plus".

För att försäkra sig om att den nya variabeln är lönsam behöver man sedan pivotera in den. Mer om det finns i powerpointen.

Nytt bivillkor

Kolla i boken!

Icke-linjär programmering (ILP)

Icke-linjär programmering (d.v.s. optimering) ställer till med en del problem som vi inte stöter på när vi bara behandlar vanliga linjära problem. Exempelvis kan vi inte ta konvexitet för givet, vilket gör det svårare att avgöra om vi har funnit ett globalt optimum. Konvexitetsbegreppet visar sig därför vara av yttersta vikt. Ett problem kallas icke-linjärt om minst en av $f(x)$, $g_i(x)$ är icke-linjär.

Konvexitet

Om ett problem är konvext så vet vi att ett lokalt optimum också är ett globalt optimum, vilket gör problemen *mycket* enklare att lösa. Således är det viktigt att vi kan avgöra huruvida våra problem är konvexa eller inte.

Ett problem kallas ett [konvext problem](#) om $f(x)$ är en konvex funktion och X är en konvex mängd, d.v.s. att samtliga bivillkor är konvexa. Observera dock att en konvex funktion är konkav för maximiproblem. Mängden ska dock alltid vara konvex, oavsett om det är ett minimi- eller maximiproblem.

Kolla in vad som definierar en konvex funktion och en konvex mängd i definitionsavsnittet (genväg: använd hyperlänken i stycket ovan). Vissa av definitionerna är dessutom kommenterade för extra tydlighet.

Konvex funktion

I ett minimeringsproblem vill vi ha en konvex målfunktion, men i ett maximeringsproblem ska det vara konvext för att funktionen ska kunna kallas konvex (rörligt värre!).

Vidare bör man göra skillnad på funktioner som är *konvexa* och funktioner som är *strikt konvexa*. Detta spelar viss roll då strikt konvexa funktioner maximalt kan ha ett optimum, medan konvexa funktioner kan ha flera (angränsande, d.v.s. ett område).

För att undersöka konvexiteten hos en funktion kan man använda sig av den s.k. *hessianen* (se: "[övrigt](#)" i definitionsavsnittet). Hessianens *teckenkaraktär* (positivt definit, positivt semidefinit, et cetera) avslöjar huruvida funktionen är (strikt) konvex, (strikt) konkav eller ingetdera. Med hjälp utav hessianens egenvärden kan man komma fram till följande:

Antag att $f(x)$ är en två gånger deriverbar funktion definierad på en konvex mängd X . Då gäller att:

*$f(x)$ är en **konvex** funktion på X om \mathcal{H} är **positivt semidefinit** för alla $x \in X$. Detta gäller då $\lambda_1 \geq 0, \dots, \lambda_n \geq 0$.*

*$f(x)$ är en **strikt konvex** funktion på X om \mathcal{H} är **positivt definit** för alla $x \in X$. Detta gäller då $\lambda_1 > 0, \dots, \lambda_n > 0$.*

*$f(x)$ är en **konkav** funktion på X om \mathcal{H} är **negativt semidefinit** för alla $x \in X$. Detta gäller då $\lambda_1 \leq 0, \dots, \lambda_n \leq 0$.*

*$f(x)$ är en **strikt konkav** funktion på X om \mathcal{H} är **negativt definit** för alla $x \in X$. Detta gäller då $\lambda_1 < 0, \dots, \lambda_n < 0$.*

*$f(x)$ är ingetdera om \mathcal{H} är **indefinit** (varierande tecken på λ).*

Man kan också använda sig utav s.k. "ledande underdeterminanter" för att avgöra hessianens teckenkaraktär, men dessa säger bara något om karaktären om den är strikt konvex eller strikt konkav. Det är dock oftast ett smidigare sätt att räkna på. För en detaljerad beskrivning, kolla avsnittet med satser i slutet på sammanfattningen.

I kursen kan man också stöta på *sammansatta funktioner*. För att avgöra om en sådan är konvex måste man titta på de olika delarna. Om $h(\mathbf{y})$ och $g(\mathbf{x})$ är sammansatta, så att $f(\mathbf{x}) = h(g(\mathbf{x}))$, så är funktionen konvex ifall *båda funktionerna är konvexa och $h(\mathbf{y})$ är icke-avtagande*. För en närmare beskrivning, se satsen om sammansatta funktioner i avsnittet med satser.

För att bedöma om *linjärkombinationer*, d.v.s. summor av termer, är konvexa så kan man analysera dem var för sig. Om samtliga termer är konvexa så är den sammansatta funktionen konvex.

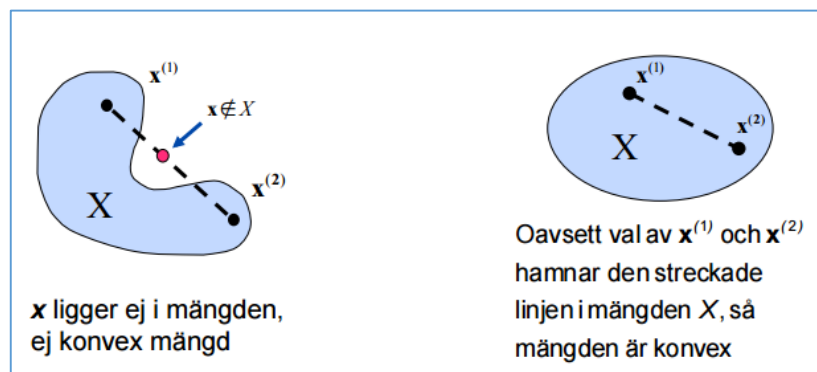
Minnesregel: *Summan av konvexa funktioner är konvex*. För mer detalj, se satsen om linjärkombinationer i satsavsnittet.

Konvexa mängder

En mängd är kallad *konvex* om det för alla punkter i mängden gäller att samtliga punkter emellan dessa två godtyckligt valda punkter också tillhör mängden. Således är kravet att man ska kunna ta två punkter i området och dra ett rakt sträck emellan dessa, utan att någonsin lämna mängden. Detta ska alltså vara sant oavsett val av punkter. Mer formellt säger man att:

En mängd $X \subset \mathbb{R}^n$ är en konvex mängd om det för varje val av punkter $x^{(1)}, x^{(2)} \in X$ och $0 \leq \lambda \leq 1$ gäller att $x = \lambda x^{(1)} + (1 - \lambda)x^{(2)} \in X$.

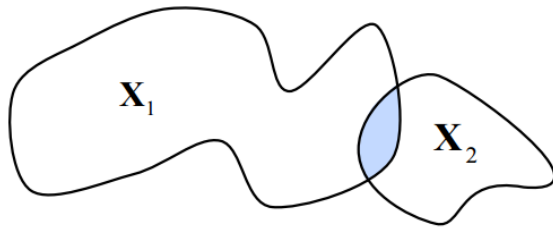
Detta kan tydligt illustreras med två figurer:



Figur 30: T.v. en icke-konvex mängd. T.h. en konvex mängd. Från FÖ7.

Mängden, eller det tillåtna området, spänns ju som bekant upp utav problemets bivillkor. Således kan man använda de olika funktionerna $g_i(x)$, d.v.s. respektive bivillkors V.L., för att avgöra om mängden är konvex. Man säger att mängden är konvex om *samtliga $g_i(x)$ är konvexa funktioner*. Notera dock att detta enbart fungerar åt ett håll. Att ett utav bivillkoren *inte* är konvext betyder inte per automatik att mängden är icke-konvex. Detta beror på att icke-konvexa funktioner kan vara konvexa inom vissa intervall, och hela funktionerna är inte nödvändigtvis intressanta i ett optimeringsproblem. Vi spinner vidare på detta när vi pratar om *snittet av konvexa mängder*.

Snittet av ett antal mängder är konvext om samtliga delmängder är konvexa. D.v.s. att om samtliga bivillkor är konvexa, så kommer *snittet* (det tillåtna området) att vara konvext. Däremot kan det te sig så att snittet blir konvext, trots att det spänns upp av ett antal icke-konvexa delmängder. Detta beror på att icke-konvexa mängder *inte är icke-konvexa överallt*. Detta demonstreras tydligt i bilden nedan:



Här ser vi alltså snittet (i blått) av två icke-konvexa delmängder. Snittet är fortfarande konvext. I sådana här situationer kan man behöva rita upp områdena grafiskt för att på så sätt avgöra om snittet är konvext eller ej.

Figur 31: Snittet av icke-konvexa delmängder. Från FÖ7.

För att avgöra om delmängderna, d.v.s. bivillkorensfunktionerna, är konvexa kan man använda hessianen, precis som för målfunktionen. Eftersom funktionerna $g_i(x)$ är just *funktioner* så kan man behandla dem precis som man behandlar funktionerna $f(x)$. Således gäller t.ex. reglerna om sammansatta funktioner och linjärkombinationer även här.

Alltså, se till att kolla in de [satser](#) och [definitioner](#) som behandlar konvexitetsbegreppet, då dessa är väldigt centrala.

Obegränsad optimering & sökmetoder

Inom obegränsad optimering behandlar vi problem som saknar bivillkor, vilket innebär att vi söker över hela målfunktionens definitionsmängd och att målfunktionsvärdet kommer att ligga någonstans inom funktionens värdemängd. Satserna om optimalitet vid [obegränsad optimering](#) är mycket relevanta för avsnittet.

Sökmetoder behöver generellt tre saker: en *startpunkt* (brukar vara given), en *sökriktning* och en *steglängd*. De två senare varierar en del mellan olika sökmetoder, vilket vi kommer att titta på. Generellt kan en sökmetod beskrivas som:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t\mathbf{d}^{(k)}$$

där $\mathbf{x}^{(k+1)}$ är den nya punkten vi går till, $\mathbf{x}^{(k)}$ är punkten vi står i t är steglängden och $\mathbf{d}^{(k)}$ är sökriktningen.

Vid obegränsad optimering, och vid sökmetoder för densamma, hittar vi enbart lokala optima. Således är det mycket viktigt att veta ifall funktionen är konvex, eftersom vi då vet att ett lokalt optima är detsamma som ett globalt optima.

Eftersom vi inte har några bivillkor i den obegränsade optimeringen vet vi att optima måste ligga i en *stationär punkt*, d.v.s. en punkt där $\nabla f(\mathbf{x}) = 0$. Vet vi dessutom att funktionen är konvex, så kommer en stationär punkt att vara en optimallösning till problemet. Dessa två delar (stationär punkt och konvex funktion) är tillräckliga för att avgöra om vi funnit en optimallösning.

Skulle vi inte ha en konvex funktion så kan vi fortfarande leta upp stationära punkter och sedan studera hessianen i denna punkt för att avgöra om det är en lokal maximi- eller minimipunkt.

Sökriktning

När vi väljer sökriktning, $\mathbf{d}^{(k)}$, studerar vi funktionen i den punkten vi står i, genom att göra en *approximation* (se: [Taylorapproximation](#)). Vi väljer sedan en riktning mot optimum för den approximerade funktionen och på så sätt tar vi oss till ett bättre målfunktionsvärde.

I den här kursen delar vi in sökriktningarna i två olika metoder: *Brantaste lutningsriktning* som används i brantaste lutningsmetoden (B.L.) och *Newtonriktning* som används i olika newtonmetoder (NM).

För att bestämma $\mathbf{d}^{(k)}$ i *brantaste lutning* (BL) använder vi helt enkelt funktionens gradient rakt av:

$$\text{Minimiproblem: } \mathbf{d} = -\nabla f(\mathbf{x})$$

$$\text{Maximiproblem: } \mathbf{d} = \nabla f(\mathbf{x})$$

Exempel

Bestäm sökriktningen för $f(x) = -2x_1 - x_1x_2 + \frac{1}{2}x_1^2 + x_2^2$ i punkten $\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ i ett minimeringsproblem. Bestäm $\mathbf{d}_{BL}^{(0)}$ (sökriktningen med BL-metoden).

Enligt formeln ovan så söker vi efter $-\nabla f(\mathbf{x})$ (ty minimiproblem). Alltså behöver vi bara räkna ut $\nabla f(\mathbf{x})$, sätta in värdena $\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ och gå i motsatt riktning.

$$\nabla f(\mathbf{x}) = \begin{pmatrix} x_1 - x_2 - 2 \\ -x_1 + 2x_2 \end{pmatrix} \Rightarrow / \mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} / \Rightarrow \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} -2 \\ 0 \end{pmatrix} \Rightarrow$$

$$\Rightarrow -\nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} 2 \\ 0 \end{pmatrix}$$

$$\text{Således har vi att } \mathbf{d}_{BL}^{(0)} = \begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

Newtons metod kan å andra sidan delas upp i två underfall, kallade den "vanliga" och den "modifierade" (det finns ännu fler varianter). Skillnaden är att steglängden i den "vanliga" metoden alltid är 1, medan den i den modifierade metoden tas fram genom en linjesökning. Sökriktningen ges i vilket fall av:

$$\mathbf{d} = -H^{-1}(\mathbf{x}) \times \nabla f(\mathbf{x})$$

Härledningen av detta gick vi igenom under föreläsning 7, men det känns inte relevant att ta upp här just nu. Istället tar vi ett exempel, med samma funktion och startpunkt som i föregående exempel.

Exempel

Bestäm sökriktningen för $f(x) = -2x_1 - x_1x_2 + \frac{1}{2}x_1^2 + x_2^2$ i punkten $\mathbf{x}^{(0)} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ i ett minimeringsproblem. Bestäm $\mathbf{d}_{NM}^{(0)}$ (sökriktningen med Newtons vanliga metod).

Enligt formeln $\mathbf{d} = -H^{-1}(\mathbf{x}) \times \nabla f(\mathbf{x})$ behöver vi ta fram gradienten och hessianen (samt invertera denna):

$$\nabla f(\mathbf{x}) = \begin{pmatrix} x_1 - x_2 - 2 \\ -x_1 + 2x_2 \end{pmatrix} \Rightarrow \nabla f(\mathbf{x}^{(0)}) = \begin{pmatrix} -2 \\ 0 \end{pmatrix}$$

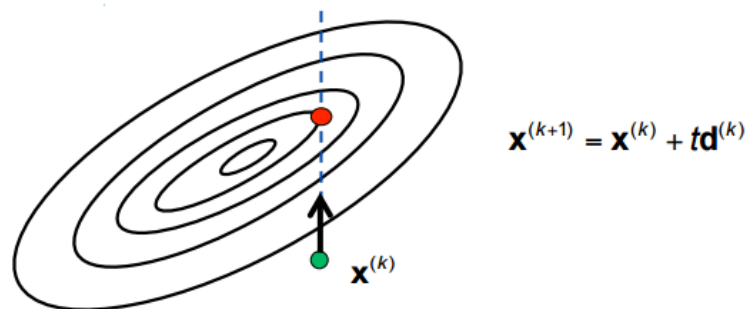
$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{pmatrix} x_1 - x_2 - 2 \\ -x_1 + 2x_2 \end{pmatrix} \Rightarrow H(\mathbf{x}) = \begin{pmatrix} 1 & -1 \\ -1 & 2 \end{pmatrix} \Rightarrow \\ \Rightarrow H^{-1}(\mathbf{x}) &= \frac{1}{1} \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \end{aligned}$$

$$\Rightarrow \mathbf{d}_{NM}^{(0)} = - \begin{pmatrix} 2 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} -2 \\ 0 \end{pmatrix} = - \begin{pmatrix} -4 \\ -2 \end{pmatrix} = \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

Notera alltså att vi får en annorlunda sökriktning i jämförelse med BL-metoden. Observera också att Newtons metod söker efter *stationära punkter*, så \mathbf{d}_{NM} kan faktiskt gå åt fel håll...

Steglängd

När vi bestämt sökriktningen måste vi också bestämma i vilken riktning vi ska gå. Generellt kan man säga att vi vill ta det största möjliga steg vi kan utan att funktionen (i sökriktningen) börjar bli "sämre" igen. Vi studerar således funktionen i riktningen $\mathbf{d}^{(k)}$ och använder den temporära variabeln t för att se vad som händer med $f(t)$ när t ökar. Figuren illustrerar detta:



Figur 32: Vi vill inte "skjuta över". Från FÖ7.

Detta kallas en *linjesökning* och är ett problem i en variabel. Vi vill rent matematiskt försöka hitta är:

$$\min_{t \geq 0} f(t) = \min_{t \geq 0} (f(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}))$$

Detta kan göras på två sätt:

1. Analytiskt: Beräkna för vilket t som $f'(t) = 0$.
2. Intervallhalvering: Antag att vi vet att $f'(t) = 0$ ligger i intervallet $[\alpha, \beta]$. Beräkna derivatan i intervallets mittpunkt m . Om:
 - $f'(m) \geq 0 \mapsto$ Nytt intervall: $[\alpha, m]$.
 - $f'(m) \leq 0 \mapsto$ Nytt intervall: $[m, \beta]$.Avsluta då t.ex. $\beta - \alpha \leq \varepsilon$, där ε är ett godtyckligt litet "fel".

Exempel: Intervallhalvering

Hitta, med hjälp av intervallhalvering, ett intervall $[\alpha, \beta]$, där $\beta - \alpha \leq 0,5$, som innehåller det optimala värdet t när man vill minimera funktionen $f(t) = 2t^2 - 4t$, $0 \leq t \leq 3$.

Vi vill alltså, enkelt instruktionerna för intervallhalvering, börja med att kolla vad derivatans värde är mitt i intervallet. Först tar vi fram derivatan:

$$f(t) = 2t^2 - 4t \Rightarrow f'(t) = 4t - 4$$

Vi börjar med intervallet $[0,3]$. Intervallets mittpunkt $m = \frac{3-0}{2} = \frac{3}{2}$. Vi testar vad derivatan i m blir:

$$f'(m) = 4 \cdot \frac{3}{2} - 4 = 2 > 0$$

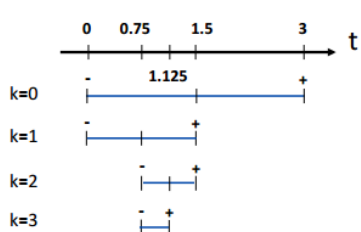
Således får vi ett nytt intervall $[0, 1.5]$. Vi testar avbrottskriteriet: $1,5 - 0 = 1,5 > \varepsilon = 0,5$, så vi måste fortsätta. Det nya intervallets mittpunkt $m = \frac{1,5+0}{2} = 0,75$. Vi kör derivatan:

$$f'(0,75) = 4 \times 0,75 - 4 = -1 < 0$$

*Nytt intervall: $[0,75, 1.5]$. Avbrottskriterium: $1,5 - 0,75 = 0,75 > 0,5$ (fortsätt).
 $m = \frac{1,5+0,75}{2} = 1,125$*

$$f'(1,125) = 4 \times 1,125 - 4 = 0,5 > 0$$

*Nytt intervall: $[0,75, 1.125]$. Avbrottskriterium: $1,125 - 0,75 = 0,375 \leq 0,5$.
Alltså avbryter vi iterationen och konstaterar att $t^* \in [0,75, 1,125]$.*



Iterationerna (halveringarna) vi gjorde i exemplet illustreras i figuren till vänster. Intervallhalvering kan vara effektivt i de fall då det är svårt att identifiera derivatans nollställen.

Figur 33: Intervallhalvering. Från FÖ7.

Brantaste lutningsmetoden

Brantaste lutningsmetoden ser stegvis ut som följer. De flesta stegen har redan beskrivits ovan.

0. Välj startpunkt $\mathbf{x}^{(0)}$, och sätt $k = 0$.
1. Bestäm sökriktning $\mathbf{d}^{(k)}$.
 Minimiproblem: $\mathbf{d}^{(k)} = -\nabla f(\mathbf{x}^{(k)})$
 Maximiproblem: $\mathbf{d}^{(k)} = \nabla f(\mathbf{x}^{(k)})$
2. Kolla avbrottskriterium.
 Punkten $\mathbf{x}^{(k)}$ är en optimallösning om normen av gradienten är tillräckligt liten, d.v.s. om $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$, där ε är en godtyckligt liten "felmarginal" (oftast given).
3. Bestäm steglängd $t^{(k)}$ genom linjesökning.
4. Identifiera den nya punkten:
 $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)} \mathbf{d}^{(k)}$
5. Uppdatera $k := k + 1$ och starta nästa iteration från steg 1.

Newtonmetoder

Newtonmetodernas algoritm ser ut som följer. Notera skillnaderna i steglängd mellan de olika varianterna.

0. Välj startpunkt $\mathbf{x}^{(0)}$, och sätt $k = 0$.
1. Bestäm $\nabla f(\mathbf{x}^{(k)})$ och $\mathbf{H}^{-1}(\mathbf{x}^{(k)})$. Beräkna sedan sökriktningen:
$$\mathbf{d}^{(k)} = -\mathbf{H}^{-1}(\mathbf{x}^{(k)})\nabla f(\mathbf{x}^{(k)})$$
2. Kolla avbrottskriterium.
Punkten $\mathbf{x}^{(k)}$ är en optimallösning om normen av gradienten är tillräckligt liten, d.v.s. om $\|\nabla f(\mathbf{x}^{(k)})\| < \varepsilon$, där ε är en godtyckligt liten "felmarginal" (oftast given).
3. Bestäm steglängd $t^{(k)}$:
Newtons "vanliga" metod \rightarrow Fix steglängd ($t^{(k)} = 1$)
Newtons modifierade metod \rightarrow Utför linjesökning.
4. Identifiera den nya punkten: $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}$
5. Uppdatera $k := k + 1$ och starta nästa iteration från steg 1.

KKT-villkoren

KKT står för *Karush, Kahn och Tucker*, vilka utvecklade villkoren. KKT-villkoren hjälper oss att avgöra om en punkt är optimal. Observera att vi i detta avsnitt använder maximiproblem med \leq -villkor, då detta blir tydligare och enklare att visa i figurer. I övrigt är det helt vanliga icke-linjära problem.

Inom den icke-begränsade optimeringen så vi att optimum alltid fanns i stationära punkter. I den begränsade optimeringen, å andra sidan, kan vi ju hitta optimum längst det tillåtna området rand, och detta behöver inte vara en stationär punkt. Där kommer KKT-villkoren in. Vi börjar med ett konstaterande:

Antag att det finns en punkt \mathbf{x}^* som uppfyller KKT-villkoren. Om problemet är konvext så är \mathbf{x}^* både ett lokalt och globalt optimum.

För konvexa problem kan vi således använda KKT-villkoren för att avgöra om en given punkt är optimal. Grundidén är att en inre punkt är ett globalt optimum om $\nabla f = 0$ och *en randpunkt är "intressant" om det inte finns någon tillåten ascentriktning* (för maximiproblem, annars blir det descentriktning enligt samma princip). Detta avgör man genom att jämföra målfunktionens gradient i en randpunkt med de aktiva bivillkorens respektive gradienter. Dessa spänner nämligen upp en "kon" och om ∇f ligger innanför denna (d.v.s. om den kan beskrivas som en icke-negativ linjärfunktion av bivillkorens gradienter) så finns det ingen tillåten ascentriktning i vilken vi kan gå vidare. Detta *måste* (förutsatt att problemet är konvext) innebära att en sådan randpunkt är ett lokalt och globalt optimum.

Det vi måste ha koll på är alltså *vilka bivillkor som är aktiva* (uppfyllda med likhet) och *om lösningen är tillåten*.

För att kolla detta, och sammanfatta KKT-villkoren i mer konkret form, kommer vi att komma tillbaka till ett tidigare tema: primalproblem och dualproblem. Det finns ett tydligt samband här emellan!

KKT-villkoren

1. Primal tillåtenhet
 $g_i(\mathbf{x}) \leq b_i, \quad i = 1, \dots, m$
2. Komplementaritet
 $v_i(b_i - g_i(\mathbf{x})) = 0, \quad i = 1, \dots, m$
3. Dual tillåtenhet
 $\nabla f(\mathbf{x}) = \sum_{i=1}^m v_i \nabla g_i(\mathbf{x}), \quad v_i \geq 0; \quad i = 1, \dots, m$

OBS! Om villkoret inte är aktivt är $v_i = 0$, d.v.s. villkoret är inte intressant!

"Konen"

Formuleringen ovan gäller som sagt för maximiproblem med \leq -villkor. För andra kombinationer erhåller man andra teckenkrav på v_i , enligt samma princip som för dualen till ett LP-problem. Kom ihåg att "*normalform ger normalform*", vilket var tumregeln för LP-problemen. En tabell över teckenkraven finns bifogad nedan:

Problemtyp	\leq -villkor	\geq -villkor	=-villkor
Minproblem	$v_i \leq 0$	$v_i \geq 0$	v_i fri
Maxproblem	$v_i \geq 0$	$v_i \leq 0$	v_i fri

Tabell 2: Teckenkrav beroende på problemtyp och villkorskaraktär. Från FÖ8.

Exempel: KKT-villkoren

Studera problemet:

$$\begin{aligned} \max_{f(\mathbf{x})} f(\mathbf{x}) &= -(x_1 + 2)^2 - 2(x_2 - 1)^2 - 2x_1 + 2x_1x_2 \\ &3x_1 + 2x_2 \geq 4 \\ \text{då} \quad &4x_1 + x_2^2 \leq 9 \\ &x_1 \geq 0 \end{aligned}$$

- a) Teckna KKT-villkoren
- b) Avgör med hjälp av KKT-villkoren ifall punkten $\mathbf{x} = (0 \ 3)^T$ utgör en KKT-punkt.
- c) Kontrollera svaret i b grafiskt.

I deluppgift a ska vi teckna KKT-villkoren för exemplet, vilket vi gör lite "generellt". Vi börjar med att namnge bivillkoren och att tilldela dem sina respektive dualvariabler:

$$\begin{aligned} g_1(\mathbf{x}) &= 3x_1 + 2x_2 \rightarrow \text{Kopplad till } v_1 \leq 0 \\ g_2(\mathbf{x}) &= 4x_1 + x_2^2 \rightarrow \text{Kopplad till } v_2 \geq 0 \\ g_3(\mathbf{x}) &= x_1 \rightarrow \text{Kopplad till } v_3 \leq 0 \end{aligned}$$

Vi tar också fram uttryck för ∇g_i och ∇f då dessa behövs:

$$\begin{aligned} \nabla f(\mathbf{x}) &= \begin{pmatrix} -2(x_1 + 2) - 2 + 2x_2 \\ -4(x_2 - 1) + 2x_1 \end{pmatrix} = \begin{pmatrix} -2x_1 + 2x_2 - 6 \\ 2x_1 - 4x_2 + 4 \end{pmatrix} \\ \nabla g_1 &= \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad \nabla g_2 = \begin{pmatrix} 4 \\ 2x_2 \end{pmatrix}, \quad \nabla g_3 = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \end{aligned}$$

Därmed kan vi sätta upp KKT-villkoren, vilket är svaret på deluppgift (a):

1. Primal tillåtenhet

$$g_1(x) \geq 4$$

$$g_2(x) \leq 9$$

$$g_3(x) \geq 0$$

2. Komplementaritet

$$v_1(4 - g_1(x)) = 0$$

$$v_2(9 - g_2(x)) = 0$$

$$v_3(0 - g_3(x)) = 0$$

3. Dual tillåtenhet

$$\nabla f(x) = v_1 \nabla g_1(x) + v_2 \nabla g_2(x) + v_3 \nabla g_3(x)$$

$$v_1, v_3 \leq 0; v_2 \geq 0$$

I deluppgift (b) testar vi om KKT-villkoren är uppfyllda för punkten $x = (0 \ 3)^T$:

1. Primal tillåtenhet

$$g_1(x) = 3 * 0 + 2 * 3 = 6 \geq 4 \text{ ok!}$$

$$g_2(x) = 4 * 0 + 3^2 = 9 \leq 9 \text{ ok!}$$

$$g_3(x) = 0 \geq 0 \text{ ok!}$$

2. Komplementaritet

$$v_1(4 - 6) = 0 \quad v_1 = 0$$

$$v_2(9 - 9) = 0 \Rightarrow v_2 = ?$$

$$v_3(0 - 0) = 0 \quad v_3 = ?$$

3. Dual tillåtenhet

$$\nabla f(x) = v_1 \nabla g_1(x) + v_2 \nabla g_2(x)$$

$$+ v_3 \nabla g_3(x)$$

$$= v_2 \nabla g_2(x) + v_3 \nabla g_3(x)$$

$$\Rightarrow \nabla f(x) = \begin{pmatrix} 0 \\ -8 \end{pmatrix} = v_2 \begin{pmatrix} 4 \\ 6 \end{pmatrix} + v_3 \begin{pmatrix} 1 \\ 0 \end{pmatrix} \Rightarrow$$

$$\Rightarrow \begin{cases} 0 = 4v_2 + v_3 \\ -8 = 6v_2 \end{cases} \Rightarrow \begin{cases} v_2 = -\frac{4}{3} \\ v_3 = \frac{16}{3} \end{cases}$$

Kontrollerar teckenkrav:

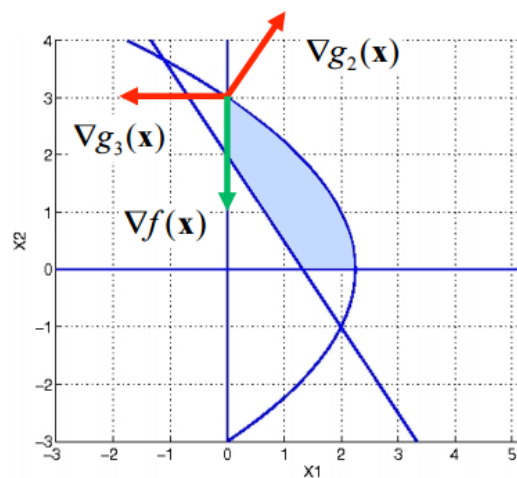
$$v_1 = 0 \leq 0 \text{ ok!}$$

$$v_2 = -\frac{4}{3} \not\leq 0 \text{ inte ok ...}$$

$$v_3 = \frac{16}{3} \not\leq 0 \text{ inte ok ...}$$

Slutsats: Punkten x är inte en KKT-punkt...

Det finns alltså en tillåten riktning som förbättrar målfunktionsvärdet. En grafisk tolkning (som efterfrågas i uppgift c) ges nedan. Notera att gradienten *inte* ligger innanför konen.



Figur 34: Punkten är INTE en KKT-punkt. Från FÖ8.

KKT-villkoren i icke-konvexa problem och i LP-problem

Om problemet inte är konvext kan det finnas punkter som uppfyller KKT-villkoren, men som inte nödvändigtvis är varken lokala eller globala maximipunkter. Det kan t.o.m. bli så tokigt så att ett globalt maximum i ett icke-linjärt problem *inte uppfyller KKT-villkoren*. Med det sagt så är det alltså oerhört viktigt att fastställa att problemet faktiskt är konvext!

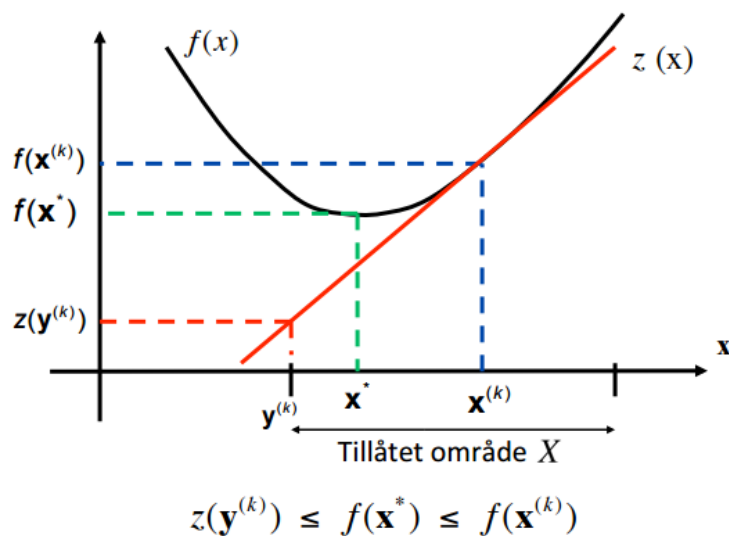
För LP-problem gäller fortfarande KKT-villkoren. De vanliga optimalitetsvillkoren för LP-problem är specialfall av KKT-villkoren.

Frank-Wolfe metoden

Frank-Wolfe (FW) används vid problem med linjära bivillkor, men en icke-linjär målfunktion. Precis som i KKT-fallet kan vi här få två scenarion, p.g.a. att målfunktionen är icke-linjär. Antingen så har vi ett optimum inuti det tillåtna området eller så har vi ett optimum längst det tillåtna områdets rand.

Grundprincipen bakom FW-metoden är att man approximerar den icke-linjära målfunktionen och därmed får en linjär funktion istället. Kravet är att funktionen är konvex, vilket gör att lösningen till det linjära problemet (vår approx.) kommer att vara en *optimistisk skattning*. Dessutom vet vi att varje tillåten punkt till det "riktiga" problemet är en *pessimistisk skattning*, varpå metoden går ut på att minska gapet mellan skattningarna och på så sätt nå fram till optimum.

En bra illustration av detta, i en variabel, finns bifogad nedan. I exemplet står y för lösningen på det approximerade exemplet, d.v.s. att y blir den optimistiska skattningen:



Figur 35: Frank-Wolfe i sin enklaste form. Från FÖ9.

Frank-Wolfes algoritm

LBD = Lower bound = Nedre gräns
UBD = Upper bound = Övre gräns

0. Startpunkt $\mathbf{x}^{(0)}$. Sätt $k = 0$, LBD = $-\infty$ och UBD = $f(\mathbf{x}^{(0)})$.

1. Formulera första ordningens Taylor-approx. av $f(\mathbf{x})$ i punkten $\mathbf{x}^{(k)}$ och lös:

$$\begin{aligned} \min z(\mathbf{x}) &= f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)}) \\ \text{då } \mathbf{Ax} &= \mathbf{b} \\ \mathbf{x} &\geq \mathbf{0} \end{aligned}$$

Beteckna LP-lösningen $\mathbf{y}^{(k)}$. Uppdatera LBD = $\max\{\text{LBD}, z(\mathbf{y}^{(k)})\}$.
Kontrollera avbrottskriterium, avbryt om UBD-LBD $< \varepsilon$

2. Definiera sökriktningen $\mathbf{d}^{(k)} = \mathbf{y}^{(k)} - \mathbf{x}^{(k)}$ och sätt $\mathbf{x}^{(k+1)}(t) = \mathbf{x}^{(k)} + t\mathbf{d}^{(k)}$

3. Utför linjesökning mellan $\mathbf{x}^{(k)}$ och $\mathbf{y}^{(k)}$, $t^{(k)}$ ges av $\min_{0 \leq t \leq 1} f(\mathbf{x}^{(k+1)}(t))$

4. $\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + t^{(k)}\mathbf{d}^{(k)}$. Uppdatera UBD = $f(\mathbf{x}^{(k+1)})$.
Kontrollera avbrottskriterium, avbryt om UBD-LBD $< \varepsilon$

5. Uppdatera $k := k + 1$, gå till steg 1.

Notera att algoritmen ovan är för ett minimiproblem. För ett maximiproblem blir det vissa skillnader. T.ex. sätter vi att LBD = $f(\mathbf{x}^{(0)})$ och UBD = ∞ när vi börjar, och z ska maximeras istället för minimeras. Det blir alltså flera skillnader, så var vaksam på dessa. Notera också att algoritmen kräver att målfunktionen är en konvex funktion (konkav för maximiproblem), för annars stämmer inte skattningarna per automatik.

Lagrangedualitet & Lagrangerelaxation

Lagrangerelaxation är en lösningsstrategi som ofta används för att lösa stora strukturerade optimeringsproblem, bl.a. heltalsproblem (som dyker upp i fortsättningskursen). Idén är att "förenkla" problemet genom att relaxera vissa bivillkor och istället ta hänsyn till dem indirekt genom att inkludera dem i målfunktionen, vilken då kallas för *Lagrangefunktion*. Skulle det relaxerade bivillkoret inte vara uppfyllt så är tanken att funktionen automatiskt ska "straffa" denna otillåtenhet, vilket den gör med hjälp av s.k. Lagrangemultiplikatorer.

Lagrangerelaxation bygger på teori rörande dualitet i icke-linjära problem, vilket brukar kallas *Lagrangedualitet*. Vi börjar därför i den änden. Grundtanken är att vi vid ett icke-linjärt problem:

$$\begin{aligned} \min f(x) \\ \text{då } g_i(x) \geq b_i, \quad i = 1, \dots, m \end{aligned}$$

ska kunna ta hänsyn till ett bivillkor $g_i(x) \geq b_i$ genom att straffa målfunktionen när det inte är uppfyllt. Straffmultiplikatorn kallar vi för v_i , och $v_i \geq 0$. Vi lägger till dessa extra termer i målfunktionen och ser till att de straffar otillåtenhet:

$$L(x, v) = f(x) + \sum_{i=1}^m v_i(b_i - g_i(x))$$

Notera att vi vårt minimiproblem ska: $\begin{cases} v_i(b_i - g_i(x)) > 0 & \text{om villkoret inte är uppfyllt} \\ v_i(b_i - g_i(x)) \leq 0 & \text{om villkoret är uppfyllt} \end{cases}$ För ett maximiproblem blir det självklart tvärt om. Det blir också annorlunda om bivillkoren har en olikhet som inte är på standardform. Tänker man bara att den extra termen ska straffa otillåtenhet så kan man räkna ut vilket tecken som krävs på v_i och hur termen ska se ut.

Det duala problemet, som bara beror på v , ska fortfarande minimera med avseende på x :

$$h(v) = \min_x L(x, v) = \min_x \left\{ f(x) + \sum_{i=1}^m v_i(b_i - g_i(x)) \right\}, v_i \geq 0 \forall i$$

En fin egenskap med detta är att ifall vi har flera variabler (vilket typ alltid är fallet) så kan vi samla dem var för sig och minimera dem separat. Detta gäller dock inte om termerna är sammansatta. Vi kikar på ett exempel där det blir så:

Exempel: Lagrangedualitet

Betrakta det följande problemet. Relaxera bivillkor 1.

$$\begin{aligned} \max z &= 3x_1 - x_2 - 3x_3 + 7x_4 \\ &x_1 - 2x_2 - x_3 + 5x_4 \leq 15 \\ &x_1 + x_2 \leq 5 \\ \text{då} \quad &x_1 \leq 3 \\ &2x_3 + x_4 \geq 4 \\ &x_4 \leq 6 \\ &x_1, x_2, x_3, x_4 \geq 0 \end{aligned}$$

Vi börjar med att formulera *Lagrangefunktionen*:

$$\begin{aligned} L(x, v) &= 3x_1 - x_2 - 3x_3 + 7x_4 + v(15 - x_1 + 2x_2 + x_3 - 5x_4) \\ &= 15v + (3 - v)x_1 + (2v - 1)x_2 + (v - 3)x_3 + (7 - 5v)x_4 \end{aligned}$$

Om vi nu tittar på de andra bivillkoren (de som kvarstår) så ser vi att de *antingen* innehåller x_1 och x_2 eller x_3 och x_4 . Inget bivillkor innehåller variabler från *båda* grupperna. Således kan vi ta det i beaktning när vi formulerar vår *duala funktion* och gruppera dem därefter, så att vi får relativt enkla problem att lösa:

$$\begin{aligned} h(v) = \min_x L(x, v) &= 15v + \max_{x_1, x_2} \{ (3 - v)x_1 + (2v - 1)x_2 \} + \max_{x_3, x_4} \{ (v - 3)x_3 + (7 - 5v)x_4 \} \\ \text{då} \quad &x_1 + x_2 \leq 5 && 2x_3 + x_4 \geq 4 \\ &x_1 \leq 3 && x_4 \leq 6 \\ &x_1, x_2 \geq 0 && x_3, x_4 \geq 0 \end{aligned}$$

Vi får således två *subproblem*, vilka enkelt kan lösas t.ex. grafiskt. En typisk uppgift skulle kunna vara att vi utöver detta får två värden på v och ska räkna ut den duala funktionens värde för de multiplikatorerna. Det skulle också kunna vara så att man med hjälp av dessa multiplikatorer ska ge ett intervall för den ursprungliga målfunktionen. Detta löses enkelt genom att sätta in v och sedan räkna ut de optimala värdena på de olika x -variablerna.

När man har de olika x -variablerna måste man kolla om koordinaten är primalt tillåten (med hjälp av det relaxerade bivillkoret). Om lösningen är *primalt tillåten* så får vi en pessimistisk skattning till det ursprungliga problemet $z(x)$ genom att sätta in koordinaterna i den ursprungliga målfunktionen. Värdet på det duala problemet kommer att ge en optimistisk skattning, oavsett primal tillåtenhet eller ej. Således får vi ett intervall att sätta det ursprungliga målfunktionsvärdet i. Den primala

tillåtenheten avslöjar dock huruvida straffet är för hårt eller för klen. Om vi får primal tillåtenhet för de x -variabler vi får fram så är straffet för hårt (om det inte är optimalt, d.v.s. om problemet är konvext och $h(\mathbf{v}) = f(\mathbf{x})$). Om vi inte får primal tillåtenhet så har straffet varit för klen. Denna info kan vi använda när vi förfinar värdet på vår multiplikator \mathbf{v} .

Man kan alltså säga att allt kommer ner till att man vill hitta *det optimala värdet på Lagrangemultiplikatorn* \mathbf{v} för att hitta optimum.

Sammanfattningsvis kan vi säga några ord om relationen mellan $f(\mathbf{x})$ och $h(\mathbf{v})$ i Lagrange-fallet. Givet ett tillåtet \mathbf{x} och ett tillåtet \mathbf{v} så gäller det att:

$$h(\mathbf{v}) \leq h(\mathbf{v}^*) \leq f(\mathbf{x}^*) \leq f(\mathbf{x})$$

Det gäller alltså **alltid** att $h(\mathbf{v})$ ger en optimistisk skattning av $f(\mathbf{x})$ och det gäller **alltid** att en tillåten lösning är en pessimistisk skattning. Sambandet ovan gäller således för ett minimiproblem. Om problemet är konvext gäller dessutom *stark dualitet*, d.v.s. att $h(\mathbf{v}^*) = f(\mathbf{x}^*)$. I det icke-konvexa fallet är det vanligt att det uppstår ett s.k. dualgap, d.v.s. att $h(\mathbf{v}^*) < f(\mathbf{x}^*)$.

Generell lösningsstrategi

0. Bestäm vilka villkor som ska relaxeras och bestäm den duala funktionen. Välj initiala värden $\mathbf{v}^{(0)}$ (oftast givna). Sätt $k = 0$, LBD = $-\infty$ och UBD = ∞
1. Lös Lagrangesubproblemet för $\mathbf{v} = \mathbf{v}^{(k)}$:
 Erhåller lösning $\mathbf{x}^{(k)}$ samt en optimistisk skattning $h(\mathbf{v}^{(k)})$.
 Uppdatera LBD = $\max\{\text{LBD} \quad h(\mathbf{v}^{(k)})\}$.
2. Om $\mathbf{x}^{(k)}$ är tillåten:
 Erhåller en pessimistisk skattning och kan uppdatera UBD = $\min\{\text{UBD} \quad f(\mathbf{x}^{(k)})\}$.

 Om $\mathbf{x}^{(k)}$ är otillåten:
 Försök omvandla till en tillåten lösning.
3. Kontrollera avbrottskriterier.
4. Uppdatera dualvariablerna: $\mathbf{v}^{(k+1)} = \mathbf{v}^{(k)} + t^{(k)} \mathbf{d}^{(k)}$, där t och \mathbf{d} representerar steglängd respektive sökriktning. Detta steg behövs dock inte om de \mathbf{v} som ska användas redan är givna.
 Sätt $k := k + 1$ och gå till steg 1.

Definitioner

Konvexitet

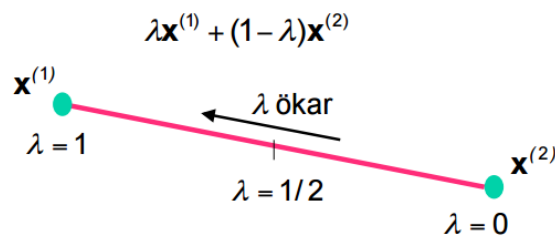
Konvexitet/Konvext problem

Ett problem är konvext om $f(x)$ är en konvex funktion och X är en konvex mängd.

OBS! LP-problem är *alltid* konvexa, medan ILP-problem kan vara antingen konvexa eller icke-konvexa. Anledningen till konvexitetens relevans är att det för konvexa problem per definition gäller att *ett lokalt optimum är ett globalt optimum*. Notera att $f(x)$ är en konvex funktion om den i ett minimiproblem är konvex och i ett maximiproblem är konkav.

Konvexkombination

En punkt y är en konvexkombination av två punkter $x^{(1)}$ och $x^{(2)}$ om $y = \lambda x^{(1)} + (1 - \lambda)x^{(2)}$ där $0 \leq \lambda \leq 1$.

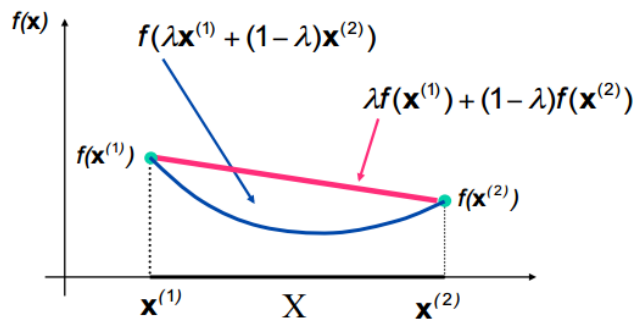


Figur 36: Konvexkombination. Tagen från FÖ1.

Kommentar: Definitionen innebär att punkten y ligger på en rät linje mellan $x^{(1)}$ och $x^{(2)}$, där värdet på λ avgör avståndet till respektive punkt. Konvexkombinationen kan användas för att beskriva samtliga punkter på en linje mellan punkterna $x^{(1)}$ och $x^{(2)}$, vilket är användbart när t.ex. optimum i ett LP-problem utgörs av alla punkter mellan två hörnpunkter (d.v.s. när ∇f är parallell med ∇g).

Konvex funktion

$f(x)$ är en konvex funktion på X om det för varje val av punkter $x^{(1)}, x^{(2)} \in X$ och $0 \leq \lambda \leq 1$ gäller att $f(\lambda x^{(1)} + (1 - \lambda)x^{(2)}) \leq \lambda f(x^{(1)}) + (1 - \lambda)f(x^{(2)})$.



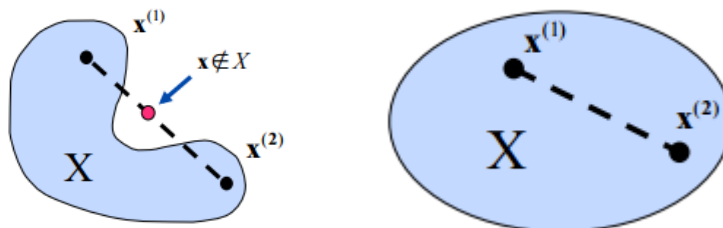
Figur 37: Konvex funktion. Tagen från FÖ1.

Kommentar: Den röda linjen i figuren är alltså konvexkombinationen mellan punkterna $f(x^{(1)})$ och $f(x^{(2)})$, medan den blå linjen är *funktionsvärdet* av alla punkter mellan $x^{(1)}$ och $x^{(2)}$. För att funktionen ska kallas konvex måste detta vara uppfyllt för alla värden på $x^{(1)}$ och $x^{(2)}$.

Observera också att detta endast gäller för ett minimiproblem. För ett maximiproblem kallas funktionen konvex när den egentligen är konkav.

Konvex mängd

En mängd $X \subset \mathbb{R}^n$ är en konvex mängd om det för varje val av punkter $x^{(1)}, x^{(2)} \in X$ och $0 \leq \lambda \leq 1$ gäller att $x = \lambda x^{(1)} + (1 - \lambda)x^{(2)} \in X$.



Figur 38: Vänster: Icke-konvex mängd. Höger: Konvex mängd. Tagen från FÖ1.

Optimalitet

Globalt optimum

En tillåten punkt $x^{(k)}$ är ett globalt optimum om det inte finns någon annan tillåten punkt med bättre funktionsvärde.

Kommentar: För ett minimiproblem innebär detta att $f(x^{(k)}) \leq f(x) \quad \forall x \in X$.

Lokalt optimum

En tillåten punkt $x^{(k)}$ är ett lokalt optimum om det i en omgivning $N(x^{(k)})$ till $x^{(k)}$ inte finns någon annan tillåten punkt med bättre målfunktionsvärde.

Kommentar: För ett minimiproblem innebär detta att $f(x^{(k)}) \leq f(x) \quad \forall x \in X$ och $x \in N(x^{(k)})$.

Sökriktningar

Sökriktningar

En riktning $\mathbf{d}^{(k)}$ sådan att $\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)} < 0$ kallas för en descentriktning och är alltså en riktning i vilken $\nabla f(\mathbf{x}^{(k)})$ minskar (önskvärt för minimiproblem).

En riktning $\mathbf{d}^{(k)}$ sådan att $\nabla f(\mathbf{x}^{(k)})^T \mathbf{d}^{(k)} > 0$ kallas för en ascentriktning och är alltså en riktning i vilken $\nabla f(\mathbf{x}^{(k)})$ ökar (önskvärt för maximiproblem).

Tillåten sökriktning

Det ska vara möjligt att ta ett litet steg i sökriktningen $\mathbf{d}^{(k)}$ från punkten $\mathbf{x}^{(k)}$ och samtidigt stanna kvar i det tillåtna området. Kravet är att det ska existera något litet $\tau > 0$, så att: $\mathbf{x}^{(k)} + t\mathbf{d}^{(k)} \in \mathbf{X}; \forall t, 0 < t < \tau$.

Förbättrande sökriktning

Målfunktionsvärdet i punkten $\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}$ ska vara bättre än målfunktionsvärdet i punkten $\mathbf{x}^{(k)}$ för tillräckligt litet $t > 0$.

Kommentar: För ett minimiproblem får vi alltså att: $f(\mathbf{x}^{(k)} + t\mathbf{d}^{(k)}) < f(\mathbf{x}^{(k)})$.

Övrigt

Degenererad baslösning

En baslösning är degenererad om en (eller flera) basvariabler har värdet noll.

Kommentar: Detta får viss effekt på t.ex. simplexalgoritmen, då flera baslösningar kan dela på samma hörnpunkt. Den reducerade kostnaden kan indikera en förbättring, men steglängden kan bli lika med noll – vi byter alltså baslösning men förblir i samma hörnpunkt. För icke-degenererade baslösningar kommer steglängden i varje iteration att vara strikt positiv och leda till ett strikt bättre målfunktionsvärde.

Relaxation

En relaxation är en förändring av ett problem som innebär att det tillåtna området blir större (får fler tillåtna lösningar) alternativt blir oförändrat.

Restriktion

En restriktion är en förändring av ett problem som innebär att det tillåtna området blir mindre (får färre tillåtna lösningar) alternativt blir oförändrat.

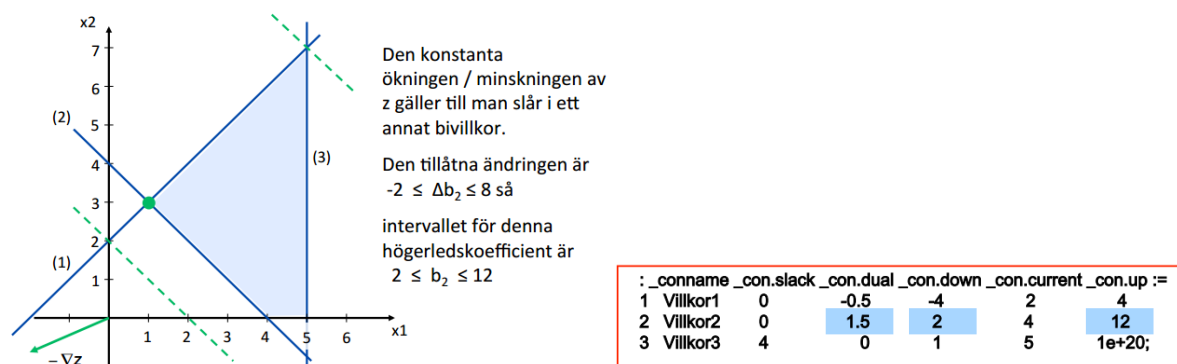
Skuggpris

Skuggpriset för ett problem ges av förändringen av det ursprungliga målfunktionsvärdet vid en enhetsökning av högerledet.

Kommentar: Skuggpris kallas även *dualpris*, *dualvärde* eller *marginalpris*. Till varje LP-problem finns det som bekant ett *dualt* problem (med exakt samma indata som vårt ursprungliga *primala* problem). I det duala problemet finns det en *dualvariabel* v_i för varje bivillkor i , i det primala problemet. Skuggpriset för bivillkoret är det optimala värdet på denna dualvariabel. Skuggpriserna ges alltså av den optimala duallösningen. Eftersom varje dualvariabel har ett teckenkrav (undantaget de fall i vilka vi har s.k. "fria" variabler) kan vi direkt utläsa huruvida en *ökning* av bivillkorets högerled kommer att innebära en relaxation eller restriktion.

Notera också att det optimala värdet på dualvariabeln (d.v.s. skuggpriset) kan tolkas som derivatan av z m.a.p. b_i , d.v.s. $v_i = \frac{\partial z}{\partial b_i}$. Det optimala värdet på dualvariabeln är alltså den *marginella förändringen* av målfunktionsvärdet (z) vid en förändring av högerledet (b_i).

Observera att skuggpriset endast gäller inom vissa intervall (det som i datautskriften benämns "_con.down" respektive "_con.up"). Detta beror på att bivillkoret inte längre *kan* vara aktivt (andra bivillkor blir mer begränsande) utanför detta intervall på högerledet. Således kan vi bara räkna med att skuggpriset gäller inom detta intervall. Ska man beräkna en förändring som går utanför intervallet måste man därför ange svaret som ett intervall, där man utgått från att skuggpriset antingen är 0 eller detsamma utanför intervallet. Det nya skuggpriset (utanför intervallet) *måste* nämligen ligga däremellan.



Figur 39: Vänster: Skuggpriset gäller enbart inom intervallet. Höger: Datautskriften visar skuggpriset och intervallet. Taget från FÖ5.

Gradient

Gradienten $\nabla f(\mathbf{x})$ är en vektor bestående av samtliga partiella derivator till funktionen $f(\mathbf{x})$ och den kan uttryckas som:

$$\nabla f(x_1, x_2, \dots, x_n) = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \vdots \\ \frac{\partial f}{\partial x_n} \end{pmatrix}$$

Kommentar: Funktionen gradient är en vektor som pekar i den riktning där *funktionsvärdet växer snabbast*. Den pekar således ditåt funktionen *ökar* mest, vilket gör att man får använda $-\nabla f(\mathbf{x})$ vid minimiproblem (för att få riktningen ditåt funktionen *minskar* mest). I en geometrisk tolkning kan man säga att gradienten är vinkelrät (normal) mot sina nivåkurvor.

Hessian

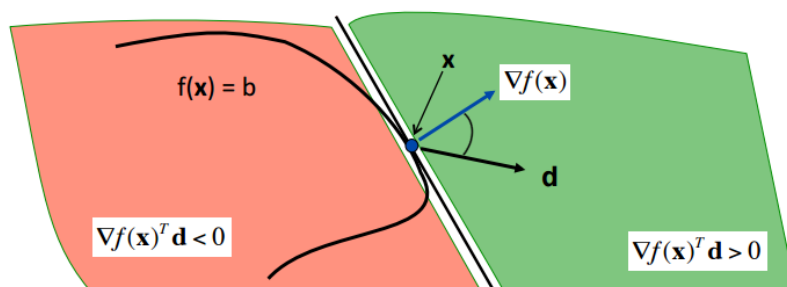
Hessianen \mathcal{H} är en matris bestående av samtliga partiella andraderivator till funktionen $f(\mathbf{x})$ och den kan uttryckas som:

$$\mathcal{H}(x_1, x_2, \dots, x_n) = \nabla^2 f(x) = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{pmatrix}$$

Kommentar: Hessianen kan sägas beskriva funktionens *krökning* och är alltid symmetrisk. Hessianen avslöjar således om funktionen är (strikt) konvex, (strikt) konkav eller ingetdera (se satsen *Hessianen, Egenvärden & Konvexitet*). Information om detta fås genom att antingen räkna ut dess egenvärden eller genom att undersöka dess ledande underdeterminanter.

Riktningderivata

Riktningderivata för funktionen $f(\mathbf{x})$ i riktningen \mathbf{d} skrivs som $\nabla f(\mathbf{x})^T \mathbf{d}$.



Figur 40: Riktningderivata, samt ascent- och descentriktning. Taget från FÖ7.

Kommentar: Riktningderivatan kan ses som en partiell derivata, men i riktningen \mathbf{d} .

Första ordningens Taylorapproximation

En första ordningens Taylorapproximation av funktionen $f(\mathbf{x})$ omkring punkten $\mathbf{x}^{(k)}$ ges av

$$f_1(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)})$$

Andra ordningens Taylorapproximation

En andra ordningens Taylorapproximation av funktionen $f(\mathbf{x})$ omkring punkten $\mathbf{x}^{(k)}$ ges av

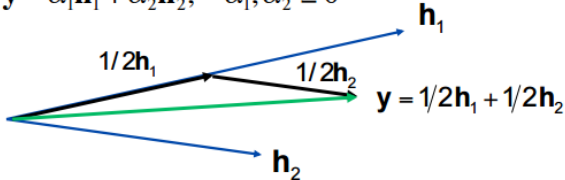
$$f_2(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \nabla f(\mathbf{x}^{(k)})^T (\mathbf{x} - \mathbf{x}^{(k)}) + \frac{1}{2} (\mathbf{x} - \mathbf{x}^{(k)})^T \mathcal{H}(\mathbf{x}^{(k)}) (\mathbf{x} - \mathbf{x}^{(k)})$$

Kon

En kon C som spänns upp av vektorerna $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_s$ består av alla vektorer \mathbf{y} som kan skrivas som en icke-negativ linjärkombination av vektorerna $\mathbf{h}_i, i = 1, \dots, s$
– det vill säga:

$$C = \left\{ \mathbf{y} : \mathbf{y} = \sum_{i=1}^s \alpha_i \mathbf{h}_i, \quad \alpha_i \geq 0, \quad i = 1, \dots, s \right\}$$

Exempel:

$$\mathbf{y} = \alpha_1 \mathbf{h}_1 + \alpha_2 \mathbf{h}_2, \quad \alpha_1, \alpha_2 \geq 0$$


Figur 41: En vektor \mathbf{y} innanför konen som spänns upp av \mathbf{h}_1 och \mathbf{h}_2 . Från FÖ8.

Satser

Konvexitet

Hessianen, Egenvärden & Konvexitet

Antag att $f(\mathbf{x})$ är en två gånger deriverbar funktion definierad på en konvex mängd \mathbf{X} . Då gäller att:

$f(\mathbf{x})$ är en **konvex** funktion på \mathbf{X} om \mathcal{H} är **positivt semidefinit** för alla $\mathbf{x} \in \mathbf{X}$. Detta gäller då $\lambda_1 \geq 0, \dots, \lambda_n \geq 0$.

$f(\mathbf{x})$ är en **strikt konvex** funktion på \mathbf{X} om \mathcal{H} är **positivt definit** för alla $\mathbf{x} \in \mathbf{X}$. Detta gäller då $\lambda_1 > 0, \dots, \lambda_n > 0$.

$f(\mathbf{x})$ är en **konkav** funktion på \mathbf{X} om \mathcal{H} är **negativt semidefinit** för alla $\mathbf{x} \in \mathbf{X}$. Detta gäller då $\lambda_1 \leq 0, \dots, \lambda_n \leq 0$.

$f(\mathbf{x})$ är en **strikt konkav** funktion på \mathbf{X} om \mathcal{H} är **negativt definit** för alla $\mathbf{x} \in \mathbf{X}$. Detta gäller då $\lambda_1 < 0, \dots, \lambda_n < 0$.

$f(\mathbf{x})$ är ingetdera om \mathcal{H} är **indefinit** (varierande tecken på λ).

Kommentar: Hessianens egenvärden kan *alltid* användas för att bevisa att funktionen är konvex eller konkav. Det kan dock vara lite omständigt, så det är ingen dum idé att känna till de s.k. ledande underdeterminanterna som definieras nedan.

Ledande underdeterminanter

\mathcal{H} är positivt definit om och endast om
 $\det(\mathbf{h}_1) > 0, \det(\mathbf{h}_2) > 0, \dots, \det(\mathcal{H}) > 0$.

\mathcal{H} är negativt definit om och endast om
 $\det(\mathbf{h}_1) < 0, \det(\mathbf{h}_2) > 0, \det(\mathbf{h}_3) < 0 \dots$
 (Observera det alternerande teckenkravet, med början på negativt).

$$\mathbf{H} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_1 \partial x_3} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial^2 f}{\partial x_2 \partial x_3} \\ \frac{\partial^2 f}{\partial x_3 \partial x_1} & \frac{\partial^2 f}{\partial x_3 \partial x_2} & \frac{\partial^2 f}{\partial x_3^2} \end{pmatrix}$$

\mathbf{h}_1 \mathbf{h}_2 $\mathbf{h}_3 = \mathbf{H}$

Figur 42: Ledande underdeterminanter. Taget från FÖ6.

Kommentar: Ledande underdeterminanter säger enbart någon om funktionen *ifall den är strikt konvex eller konkav*. I andra fall är det värdelöst. Det är dock en ganska snabb och smidig metod, som oftast är enklare att använda än egenvärdena.

Sammansatta funktioner (konvexitet)

Låt $h(\mathbf{y})$ och $g(\mathbf{x})$ vara konvexa funktioner och låt dessutom $h(\mathbf{y})$ vara en icke-avtagande funktion. Då är den sammansatta funktionen $f(\mathbf{x}) = h(g(\mathbf{x}))$ en konvex funktion.

Exempel: Funktionen $f(x) = e^{2x_1^2 + x_2^2}$ är konvex, ty $g(x) = 2x_1^2 + x_2^2$ är konvex (studera hessianen) och $h(y) = e^y$ är konvex och icke-avtagande.

Linjärkombinationer

Om $f_1(x), f_2(x), \dots, f_p(x)$ är konvexa funktioner och vi har $\lambda_i \geq 0, i = 1, \dots, p$ så är även $f(x) = \sum_{i=1}^p \lambda_i f_i(x)$ en konvex funktion.

Kommentar: Summan av konvexa funktioner är konvex.

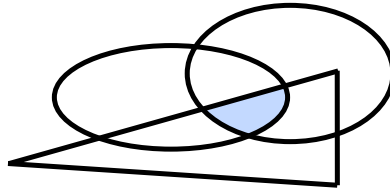
Konvex mängd

Mängden $X = \{x : g(x) \leq b\}$ är en konvex mängd om funktionen $g(x)$ är en konvex funktion.

Kommentar: För att avgöra om ett bivillkor (eller en mängd) är konvext undersöker man funktionen precis som ovan. För denna gäller samma regler som för målfunktionen, d.v.s. i avseende t.ex. sammansatta funktioner o.s.v.). Observera dock att det bara är vissa områden som är intressanta. En icke-konvex funktion kan vara konvex på det intressanta området, t.ex. i $\ln(x) \leq 2$.

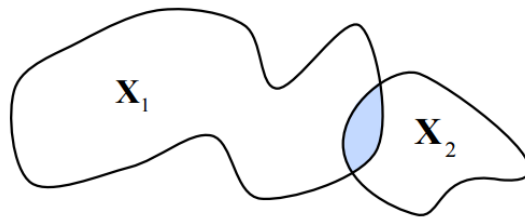
Snittet av konvexa mängder

Låt X_1, X_2, \dots, X_p vara konvexa mängder. Då är även snittet $X = X_1 \cap X_2 \cap \dots \cap X_p$ av mängderna en konvex mängd.



Figur 43: Snittet av konvexa mängder är en konvex mängd. Taget från FÖ7.

Kommentar: Även ett snitt av icke-konvexa mängder (eller en kombination av konvexa och icke-konvexa mängder) kan vara konvex (på vissa områden). Således kan vi inte dra slutsatsen att snittet är icke-konvext bara utifrån att studera de enskilda bivillkoren för sig, utan i det fallet då vi har icke-konvexa delmängder så måste vi analysera vidare. I den här kursen görs det enklast genom en grafisk analys av mängderna.



Figur 44: Snittet av två eller flera icke-konvexa mängder kan vara konvext. Taget från FÖ7.

Linjärprogrammeringens fundamentalsats

Antag att det tillåtna området X till ett LP-problem är begränsat och icke-tomt (det existerar en begränsad optimallösning till LP-problemet). Då gäller att maximum/minimum av målfunktionen $c^T x$ antas i en extrempunkt $x^{(k)}$ i X .

Kommentar: Satsen säger att optimum i ett LP-problem ligger i en av det tillåtna områdets hörnpunkter. Detta kan man utnyttja i en lösningsmetod. Detta beror på att LP-problem alltid är konvexa.

Dualitet

Den duala funktionen

För ett minimiproblem är den duala funktionen en konkav funktion. För ett maximeringsproblem är det duala problemet en konvex funktion.

Svag dualitet (I)

Antag att $\hat{\mathbf{x}}$ är en tillåten lösning till det primala problemet (P) och att $\hat{\mathbf{v}}$ är en tillåten lösning till det duala problemet (D). Då gäller det att $\mathbf{c}^T \hat{\mathbf{x}} \leq \mathbf{b}^T \hat{\mathbf{v}}$.

Kommentar: Detta innebär att en tillåten lösning i det ena problemet alltid ger en optimistisk skattning av det optimala målfunktionsvärdet i det andra problemet.

Bevis: Vi utnyttjar att det för tillåtna lösningar \mathbf{v} gäller att $\mathbf{c} \leq \mathbf{A}^T \mathbf{v}$ och att det för tillåtna lösningar \mathbf{x} gäller att $\mathbf{A}\mathbf{x} \leq \mathbf{b}$. Då fås:

$$\mathbf{c}^T \hat{\mathbf{x}} \leq (\mathbf{A}^T \hat{\mathbf{v}})^T \hat{\mathbf{x}} = \hat{\mathbf{v}}^T \mathbf{A} \hat{\mathbf{x}} \leq \hat{\mathbf{v}}^T \mathbf{b}$$

Optimalitetssats i primal och dual

Om $\hat{\mathbf{x}}$ och $\hat{\mathbf{v}}$ är tillåtna lösningar till (P) respektive (D) och $\mathbf{c}^T \hat{\mathbf{x}} = \mathbf{b}^T \hat{\mathbf{v}}$, så är $\hat{\mathbf{x}}$ och $\hat{\mathbf{v}}$ optimala lösningar till respektive problem.

Stark dualitet

Om endera (P) eller (D) har en begränsad optimallösning så har även det andra problemet en begränsad optimallösning. Dessutom gäller då att: $\mathbf{c}^T \hat{\mathbf{x}} = \mathbf{b}^T \hat{\mathbf{v}}$.

Kommentar: Med hjälp av denna sats räcker det att hitta optimum i ett av problemen (det primala eller det duala) och sedan bara hitta motsvarande punkt i det andra problemet. Det gäller även att de optimala målfunktionsvärdena i respektive problem är lika.

Komplementsatsen

Antag att $\hat{\mathbf{x}}$ och $\hat{\mathbf{v}}$ är tillåtna lösningar i (P) respektive (D). Då är $\hat{\mathbf{x}}$ och $\hat{\mathbf{v}}$ optimala lösningar till respektive problem om och endast om komplementvillkoren $\hat{\mathbf{v}}^T (\mathbf{b} - \mathbf{A}\hat{\mathbf{x}}) = \mathbf{0}$ och $\hat{\mathbf{x}}^T (\mathbf{A}^T \hat{\mathbf{v}} - \mathbf{c}) = \mathbf{0}$ är uppfyllda.

Dualsatsen

Antal att $\mathbf{x}_B = \mathbf{B}^{-1} \mathbf{b}$ är en optimallösning till (P). Då är $\bar{\mathbf{v}}^T = \mathbf{c}_B^T \mathbf{B}^{-1}$ en optimallösning till (D) och $z^ = w^*$.*

Svag dualitet (II, Lagrangedualitet)

Antag att \mathbf{x}^ är en optimallösning till ett ickelinjärt problem (ILP). För varje $\mathbf{v} \geq \mathbf{0}$ gäller det att $h(\mathbf{v}) \leq f(\mathbf{x}^*)$.*

Kommentar: Satsen säger att $h(\mathbf{v})$ alltid är mindre än funktionsvärdet för den ursprungliga funktionens optimallösning. Således ter det sig naturligt att vi vill maximera $h(\mathbf{v})$ för att på så sätt hitta optimallösningen i det ursprungliga problemet. Observera att detta gäller för minimiproblem. $h(\mathbf{v})$ bör i det generella fallet betraktas som en *optimistisk skattning*, d.v.s. att $h(\mathbf{v}) \geq f(\mathbf{x}^*)$ i ett maximeringsproblem, varpå vi istället vill minimera $h(\mathbf{v})$.

Notera att om $f(\mathbf{x})$ är konvex och samtliga bivillkor definierar en konvex mängd så gäller stark dualitet, d.v.s. att $h(\mathbf{v}^*) = f(\mathbf{x}^*)$. I det icke-konvexa fallet är det vanligt med ett s.k. "dualgap", d.v.s. att $h(\mathbf{v}^*) < f(\mathbf{x}^*)$ för ett minimiproblem och tvärt om för ett maximiproblem.

Obegränsad optimering

Optimalitetsvillkor (obegränsad optimering) I

Ett nödvändigt villkor för att det obegränsade problemet $\min f(\mathbf{x})$ ska ha en optimallösning i punkten \mathbf{x}^ är att den är en stationär punkt, d.v.s. att $\nabla f(\mathbf{x}) = \mathbf{0}$.*

Kommentar: Om det inte är en stationär punkt kan det inte vara en optimallösning. Att det är en optimallösning implicerar att det är en stationär punkt. Observera dock att detta endast gäller för obegränsade problem. Begränsade problem kan ha sitt optimum längst områdets rand.

Optimalitetsvillkor (obegränsad optimering) II

Ett tillräckligt villkor för att det obegränsade problemet $\min f(\mathbf{x})$ ska ha en optimallösning i punkten \mathbf{x}^ är att den är en stationär punkt, samt att funktionen är konvex.*

Kommentar: Om man för en konvex funktion hittar en stationär punkt så kan man dra slutsatsen att detta också är en optimallösning till problemet. En stationär punkt till ett konvext problem *måste* vara optimal.

Optimalitetsvillkor (obegränsad optimering) III

Ett tillräckligt villkor för att det obegränsade problemet $\min f(\mathbf{x})$ ska ha ett lokalt optimum i punkten \mathbf{x}^ är att den är en stationär punkt, samt att hessianen i punkten \mathbf{x}^* är positivt definit.*

Kommentar: Om Hessianen är positivt definit för en stationär punkt vet man att punkten är ett *lokalt* optima (för minimiproblem, maximiproblem kräver analogt en negativt definit hessian).

Övrigt

KKT-villkor (optimalitet)

Antag att det finns en punkt \mathbf{x}^ som uppfyller KKT-villkoren. Om problemet är konvext så är \mathbf{x}^* både ett lokalt och globalt optimum.*

Kommentar: För konvexa problem kan vi alltså använda KKT-villkoren för att avgöra om en given punkt är optimal.

Källor

Holmberg, Kaj. 2015. http://courses.mai.liu.se/GU/TAOP86/Fo/h-TAOP86_04_LPdual.pdf

Lundgren, J., Rönnqvist, M. & Värbrand, P. *Optimeringslära*. 2010. Studentlitteratur.

Quttineh, Nils-Hassan. 2015. PowerPoint-presentationer och exempeluppgifter från föreläsningar, hämtat från LISAM.