



EXAMENSARBETEN I MATEMATIK

MATEMATISKA INSTITUTIONEN, STOCKHOLMS UNIVERSITET

Heltalsoptimering

av

Mårten Knutsson

2007 - No 17

Heltalsoptimering

Mårten Knutsson

Examensarbete i matematik 20 poäng, fördjupningskurs

Handledare: Yishao Zhou

2007

Sammanfattning

Min uppsats behandlar heltalsoptimering. Uppsatsen består av två teoretiska huvuddelar som bägge också innehåller exempel på hur teorin kan användas. Den första behandlar linjära heltalsproblem, där jag går igenom Gomorys plansnittningsmetod. I den andra teoridelen undersöker jag icke-linjär optimeringsteori, som man kan använda sig av för att lösa heltalsproblem. Framför allt undersöks egenskaper hos den duala funktionen, speciellt hur man kan använda sig av subgradientoptimering för att hitta den duala funktionens optimumvärde.

Uppsatsen avslutas med en modellering av ett verkligt schemalägningsproblem.

Tack

Jag skulle här vilja passa på och tacka flera personer. Först min mor för att hon har korrekturläst flera versioner av min uppsats, utan att klaga. Erik Hermelin ska ha ett stort tack för all datorsupport och Magnus Åhl för hjälp med \LaTeX .

Sist men inte minst vill jag tacka min handledare Yishao Zhao för att hon kommit med kreativa förslag samt påpekat matematiska fel som jag har missat.

Innehåll

1 Inledning	5
1.1 Introduktion	5
1.1.1 Historik	7
1.1.2 Heltalsoptimering	7
1.1.3 Matematisk programmering	7
2 Grundläggande teori	9
2.1 Konvexitet	9
2.2 Simplexmetoden	10
2.3 Duala problemet	13
3 Linjär heltalsprogramering	15
3.1 Rena heltalsproblem	18
3.2 Blandade heltalsproblem	24
4 Ickelinjär optimering	27
4.1 Den duala funktionen	27
4.2 Subgradient	34
4.3 En algoritm	42

5 Schemaläggning	44
5.1 Målfunktionen	44
5.2 Bivillkor	45
5.3 Uppdelningen av tiden	46
5.4 Matematisk modellering	47
5.4.1 Parametrar	47
5.4.2 Variabler	48
5.4.3 Målfunktion och bivillkor	48
5.5 Resultat	50
A Arbetsdata	54
B AMPL-kod för städproblemet	62
C Sorteringsalgoritmen	71
D AMPL-kod till exempel 1 och exempel 2	78

Kapitel 1

Inledning

1.1 Introduktion

Optimeringslära används för att finna det bästa handlingsalternativet i olika beslutssituationer. För att kunna tillämpa optimeringslära på ett konkret problem, gäller det att översätta det verkliga problemet till ett matematiskt problem.

En utgångspunkt för att problemet ska gå att optimera är att det finns något som kan varieras. Det som kan varieras kallas för problemets *variabler*. Målet är att hitta de värden på variablerna som ger det bästa resultatet, vilket givetvis beror på vad man vill optimera, något som inte alltid är uppenbart. Det man vill optimera uttrycks med hjälp av en *målfunktion* som beror av variablerna. Problemet blir triviale, om det inte finns några begränsningar på variablerna, eftersom målfunktionen då kan bli hur stor(eller liten) som helst. Variablernas begränsningar regleras via bivillkor. Optimeringsproblemet skrivs matematiskt på följande sätt

$$\min f(\mathbf{x}) \tag{1.1}$$

$$\text{då } \mathbf{x} \in \mathbf{X} \tag{1.2}$$

där (1.1) beskriver om det är ett maximerings- eller minimeringsproblem samt visar målfunktionen. (1.2) visar optimeringsproblemets bivillkor. Detta område beskrivs oftast med ett ekvationssystem av olikheter och/eller likheter, vilka visar de olika variablernas begränsningar.

Exempel 1 Vi ska nu titta på dietproblemet, som är ett klassiskt optimeringsproblem. Problemställningen är att man ska välja mellan fyra olika varor: kött, fisk, bröd och frukt. De olika varorna har olika pris och innehåller olika procenthalter av det dagliga vitaminbehovet enligt följande tabell

	Pris per förpackning	Dagsbehovet av Vitamin A	Dagsbehovet av Vitamin B1	Dagsbehovet av Vitamin C
kött	31.9 kr	60 %	10 %	20 %
fisk	22.9 kr	40 %	40 %	10 %
bröd	16.2 kr	20 %	60 %	30 %
frukt	15.2 kr	10 %	35 %	80 %

Vi är nu intresserade av hur mycket man ska köpa av de olika varorna för att täcka en veckas vitaminbehov. Målet är att det ska kosta så lite som möjligt. Låt $X_{\text{kött}}$, X_{fisk} , $X_{\text{bröd}}$, X_{frukt} vara mängden av de olika varorna. Detta problem kan man formulera matematiskt så här

$$\text{Minimera } 3.19X_{\text{kött}} + 2.29X_{\text{fisk}} + 1.62X_{\text{bröd}} + 1.52X_{\text{frukt}} \quad (1.3)$$

då

$$\begin{aligned} 60X_{\text{kött}} + 40X_{\text{fisk}} + 20X_{\text{bröd}} + 10X_{\text{frukt}} &\geq 700 \\ 10X_{\text{kött}} + 40X_{\text{fisk}} + 60X_{\text{bröd}} + 35X_{\text{frukt}} &\geq 700 \\ 20X_{\text{kött}} + 10X_{\text{fisk}} + 30X_{\text{bröd}} + 80X_{\text{frukt}} &\geq 700 \end{aligned} \quad (1.4)$$

Där (1.3) visar totalkostnaden som vi vill minimera och (1.4) visar, att under en veckas tid måste vi få minst 700% av dagsbehovet av de olika vitaminerna, alltså minst 100 % per dag. Lösningen till detta problem är att köpa en varukorg som består av: 8,34 förpackningar kött, 0 förpackningar fisk, 8,17 förpackningar bröd och 3,59 förpackningar frukt. Detta ger som lägsta kostnad 453,25 kr.

Bivillkoren kan också ha formen av likheter som i aktieportföljsoptimering, där de olika aktiernas vikter måste summeras till ett. I denna uppsats kommer vi att diskutera heltalsoptimering och då måste bivillkoren dels uppfylla vissa olikheter/likheter, dels vara heltal.

1.1.1 Historik

Optimeringslära är en förhållandevis ny gren inom matematiken. Den började användas under andra världskriget. Anledningen till att utvecklingen till att börja med gick så långsamt, är att konkreta optimeringsproblem blir mycket omfattande och därför praktiskt taget olösbare utan hjälp av en dator. Optimeringsproblem som uppkom under kriget var bla: optimering av utbildningstiden för soldater, optimering av tillförsel av materiel och förnödenheter åt soldater ute i fält samt att optimera användandet av begränsade resurser inom krigsindustrin tex stål till ubåtar, flygplan och kanoner.

1.1.2 Heltalsoptimering

Heltalsoptimering uppkommer dels när problemen är naturligt heltaliga, dels när man vill använda logiska 0/1-variabler.

Exempel 2 Ett naturligt heltalsproblem är när man i exempel 1 enbart får köpa hela förpackningar av de olika varorna. Hur ska man avrunda? Om man ställer upp problemet som i exempel 1 men också kräver att $X_{\text{kött}}$, X_{fisk} , $X_{\text{bröd}}$, X_{frukt} ska vara positiva heltal och löser det, så får man följande svar. Varukorgen kostar 457,3 kr alltså lite dyrare, vilket är rimligt, eftersom vi inte har lika många valalternativ längre. Däremot består varukorgen nu av: 4 paket kött, 9 paket fisk, 2 paket bröd och 6 paket frukt. Detta är svårt att gissa sig till när man tittar på hur varukorgen såg ut i exempel 1.

Om man använder logiska variabler får man utökade modelleringsmöjligheter. Man kan beskriva ja/nej-beslut, som tex kan uppkomma om man ska välja mellan olika investeringsbeslut, där 1 står för ja och 0 för nej. Man kan modellera fasta kostnader genom att använda sig av en 0/1-variabel. Man lägger på en term $\textit{konstant} * x_j$ där x_j antar värdet 1 om aktivitet j genomförs, annars antar x_j värdet 0.

1.1.3 Matematisk programmering

Matematisk programmering används för att lösa de olika problem som uppkommer i optimeringslära. Ordet programmering har inget med datorprogrammering att göra utan syftar på planering. Linjära optimeringsproblem löses med linjär programmering(LP), icke linjära problem löses med icke linjär pro-

gramering(NP) och heltalsproblem löses med heltalsprogramering.

Kapitel 2

Grundläggande teori

Denna uppsats handlar om heltalsoptimering. Innan vi går in på hur man löser dessa problem, ska vi titta på grundläggande optimeringsteori, speciellt linjär optimering. Denna teori kommer vi att använda oss av när vi i de följande avsnitten tittar på heltalsoptimering. Teorin i detta avsnitt är främst tagen från [1] och [2] i litteraturförteckningen.

2.1 Konvexitet

Det viktigaste begreppet inom optimeringslära är utan tvekan begreppet konvexitet.

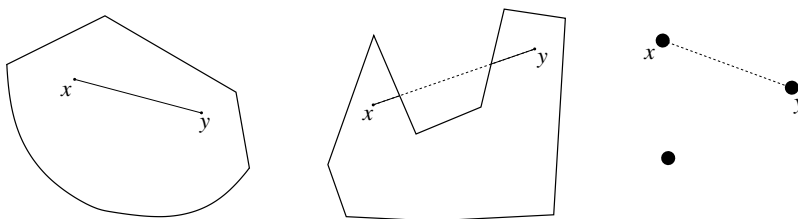
Definition 1. *En mängd $\mathbf{K} \subset \mathbb{R}^n$ är konvex om*

$$\lambda \mathbf{x} + (1 - \lambda) \mathbf{y} \in \mathbf{K} \quad \forall \mathbf{x}, \mathbf{y} \in \mathbf{K} \text{ och } 0 \leq \lambda \leq 1$$

Man kan se det som att om man tar två punkter som ligger i \mathbf{K} , så ska linjen som sammanbinder de två punkterna ligga i \mathbf{K} (fig 2.1).

Definition 2. *En funktion f är konvex över $M \in \mathbb{R}^n$ om*

$$f(\lambda \mathbf{x} + (1 - \lambda) \mathbf{y}) \leq \lambda f(\mathbf{x}) + (1 - \lambda) f(\mathbf{y}) \quad \forall \mathbf{x}, \mathbf{y} \in M \text{ och } 0 \leq \lambda \leq 1$$



Figur 2.1: Den enda konvexa mängden är mängden längst till höger. Mängden längst till vänster är en heltalsmängd och den är inte konvex.

Konvexa funktioner, som är definierade över en konvex mängd, har flera trevliga egenskaper. Man kan visa att de är kontinuerliga samt att varje lokalt optimum också är ett globalt optimum.

Vi kan här notera att linjära funktioner är konvexa funktioner. Förutom konvexa funktioner finns det konkava funktioner. För en konkav funktion f gäller det att $-f$ är konvex. All teori för konvexa funktioner kan alltså tillämpas på konkava funktioner f genom att multiplicera med (-1) .

2.2 Simplexmetoden

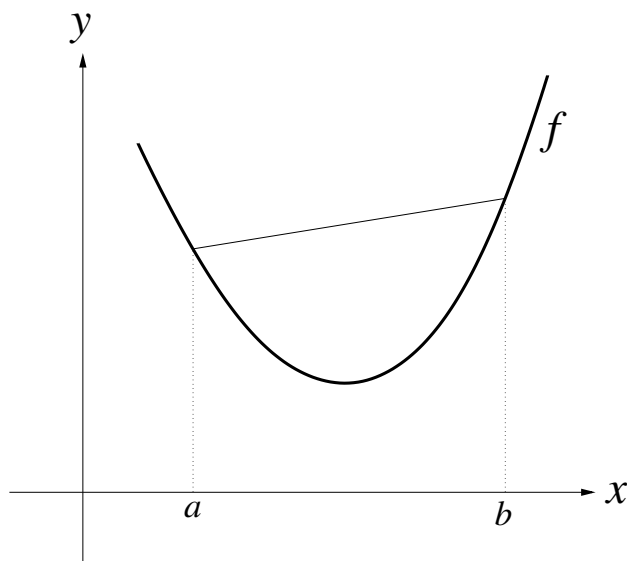
Simplexmetoden¹ är en effektiv metod för att lösa ett linjärt optimeringsproblem, där variablerna får ta vilka värden som helst. Vi kommer att ha användning av simplexmetoden längre fram i texten. Här kommer därför en kort introduktionstext till simplexmetoden.

För att använda simplexmetoden vill vi ha ett optimeringsproblem på följande form.

Problem Minimera v då $y_j \geq 0$ samt att följande ska gälla

$$\begin{aligned}
 c_1 y_1 + c_2 y_2 + \dots + c_n y_n &= v \\
 a_{11} y_1 + a_{12} y_2 + \dots + a_{1n} y_n &= b_1 \\
 a_{21} y_1 + a_{22} y_2 + \dots + a_{2n} y_n &= b_2 \\
 \vdots & \quad \quad \quad \vdots & \quad \quad \quad \vdots \\
 a_{m1} y_1 + a_{m2} y_2 + \dots + a_{mn} y_n &= b_m
 \end{aligned} \tag{2.1}$$

¹Se exempelvis boken Linear Programming and Extensions av Dantzig



Figur 2.2: Geometriskt kan man se det som att kordan ligger ovanför funktionen.

Ekvationer av typen $a_{i1}y_1 + a_{i2}y_2 + \dots + a_{in}y_n = b_i$ definierar *hyperplan* som bestämmer problemets definitionsmängd, en *konvex polytop*.

Definition 3. *Ett hyperplan H definieras av mängden av alla punkter $\mathbf{x} \in \mathbb{R}^n$ som satisfierar ekvationen $\sum_{i=1}^n a_i x_i = k$ eller i vektornotation $H = \{\mathbf{x} : \mathbf{a}^t \mathbf{x} = k\}$*

I \mathbb{R}^2 och \mathbb{R}^3 brukar man kalla hyperplanen för linje respektive plan. Ett hyperplan i \mathbb{R}^n utgör ett slutet halvrum i \mathbb{R}^{n-1} .

Definition 4. *En mängd som är skärningen av ett ändligt antal slutna halvrum kallas en konvex polytop. De hyperplan som utgör halvrummens ränder kallas polytopens begränsande hyperplan.*

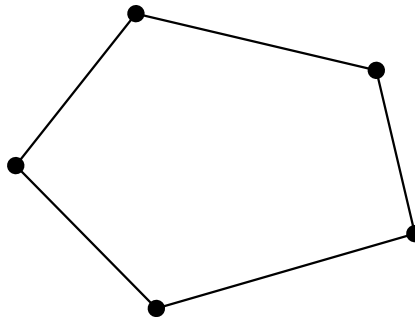
Man kan visa att alla konvexa polytoper² är konvexa mängder. Alla linjära optimeringsproblem på formen (2.1) kommer alltså att ha minst en optimumpunkt. Detta gör att alla linjära optimeringsproblem går att lösa med simplexmetoden.

²I fortsättningen kommer vi att säga polytop istället för konvex polytop.

Man kommer alltid att kunna presentera våra linjära optimeringsproblem på samma form som ovan genom att införa extra variabler s.k slackvariabler.

Exempel 3 Om y får anta både positiva och negativa värden så kan vi ersätta y med $y' - y''$ där $y', y'' \geq 0$. På liknade sätt kan man göra om man har ett villkor som säger $y \leq b$, nämligen att ersätta y med $y' + y''$ där $y' + y'' = b, y', y'' \geq 0$. Om uppgiften vore att maximera v så är det samma sak som att minimera $-v$.

Simplexmetoden använder sig av, att man vet att optimumpunkten eller en av optimumpunkterna, kommer att vara i en hörnpunkt³ till den polytop som spänns upp av hyperplanen $a_{i1}y_1 + a_{i2}y_2 + \dots + a_{in}y_n = b_i$ för $i = 1, 2, \dots, m$. Eftersom man vet var man ska leta efter optimumpunk-



Figur 2.3: Så här kan en konvex polytop se ut i \mathbb{R}^2 . Punkterna visar var en optimumpunkt kan finnas.

ten/punkterna, så går det ganska fort att hitta ett optimum. Man börjar i en hörnpunkt. Om inte den hörnpunkten är ett optimum, går man mot nästa hörnpunkt i den riktning som får v att minska mest och så håller man på tills man är färdig.

Vi ser här, att anledningen till att heltalsproblemen inte kan lösas med hjälp av simplexmetoden, är att optimumpunkterna inte ligger i hörnpunkterna. I försettningen kommer vi inte att bry oss om hur man rent praktiskt använder simplexmetoden, utan vi får vårt problem (2.1) löst och presenterat på

³Det kan finnas flera optimumpunkter. De ligger då längs en av polytopens kanter som sammanbinder två hörnpunkter.

följande form

$$\begin{array}{rcl}
 \bar{c}_1 y_1 + \bar{c}_2 y_2 + \dots + \bar{c}_{n-m} y_{n-m} & = & v - v_0 \\
 \bar{a}_{1,1} y_1 + \bar{a}_{1,2} y_2 + \dots + \bar{a}_{1,n-m} y_{n-m} + y_{n-m+1} & = & \bar{b}_1 \\
 \bar{a}_{2,1} y_1 + \bar{a}_{2,2} y_2 + \dots + \bar{a}_{2,n-m} y_{n-m} + y_{n-m+2} & = & \bar{b}_2 \\
 \vdots & & \vdots \\
 \bar{a}_{m,1} y_1 + \bar{a}_{m,2} y_2 + \dots + \bar{a}_{m,n-m} y_{n-m} + y_n & = & \bar{b}_m
 \end{array} \quad (2.2)$$

där $\bar{c}_j \geq 0$ och alltså fås det minsta värdet på v , nämligen v_0 , då man sätter variablerna y_1, y_2, \dots, y_{n-m} till noll. Vi ser då också att $\bar{b}_i \geq 0$ för annars skulle inte villkoret $y_i \geq 0$ vara uppfyllt för $i = n - m + 1, n - m + 2, \dots, n$. $y_{n-m+1}, y_{n-m+2}, \dots, y_n$ kallas för våra *basvariabler* och y_1, y_2, \dots, y_{n-m} kallas våra *icke basvariabler*.

2.3 Duala problemet

Till varje linjärt problem kan man formulera ett *dualt problem*⁴ med samma indata som definierar det ursprungliga *primala problemet*. Här kommer ett exempel i vektornotation på hur det kan se ut.

$$\begin{array}{ll}
 \text{Primalt problem} & \text{Dualt problem} \\
 \min \quad z = \mathbf{c}^t \mathbf{x} & \max \quad w = \mathbf{b}^t \mathbf{v} \\
 \text{b.v} \quad \mathbf{A} \mathbf{x} \leq \mathbf{b} & \text{b.v} \quad \mathbf{A}^t \mathbf{v} \geq \mathbf{c} \\
 \mathbf{x} \geq \mathbf{0} & \mathbf{v} \geq \mathbf{0}
 \end{array} \quad (2.3)$$

De två olika problemen förhåller sig på följande sätt. Till varje primalt villkor i hör en dual variabel v_i och till varje variabel x_j hör ett dualt villkor j . Målfunktionkoefficienterna i det ena problemet utgör högerledet i det andra problemet och vice versa. Om det ena problemet är ett minproblem så är det andra problemet ett maxproblem. Vi sammanfattar i följande tabell hur det primala problemets variabler påverkar det duala problemets bivillkor samt hur det primala problemets bivillkor påverkar det duala problemets variabler. Vi utgår från att det primala och det duala problemet står i normalform⁵.

⁴För mer grundläggande dualitetsteori hänvisas till Linear Programming and Extensions av Dantzig

⁵Det primala problemet står på normalform i (2.3)

Primalt problem		Dualt problem
Om en variabel $x_j \leq 0$	\implies	Ett bivillkor av omvänd typ
En fri variabel	\implies	Ett bivillkor med likhet
Ett bivillkor av omvänd typ	\implies	En variabel $v_i \leq 0$
Ett bivillkor med likhet	\implies	Ger en fri variabel

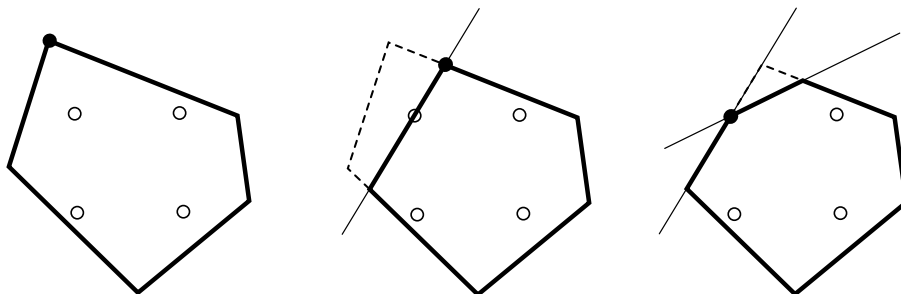
Om man tittar på det duala problemet till det duala problemet så kommer man tillbaka till det ursprungliga problemet, nämligen det primala problemet.

Anledningen till att man är intresserad av det duala problemet är att det kan vara ett enklare problem att lösa. Detta kommer vi att se längre fram i texten.

$$\begin{array}{rcl}
\bar{c}_1 y_1 + \bar{c}_2 y_2 + \dots + \bar{c}_{n-m} y_{n-m} & & = v - \bar{v}_0 \\
\bar{a}_{1,1} y_1 + \bar{a}_{1,2} y_2 + \dots + \bar{a}_{1,n-m} y_{n-m} + y_{n-m+1} & & = \bar{b}_1 \\
\bar{a}_{2,1} y_1 + \bar{a}_{2,2} y_2 + \dots + \bar{a}_{2,n-m} y_{n-m} + y_{n-m+2} & & = \bar{b}_2 \\
\vdots & \vdots & \ddots \\
\bar{a}_{m,1} y_1 + \bar{a}_{m,2} y_2 + \dots + \bar{a}_{m,n-m} y_{n-m} + y_n & & = \bar{b}_m
\end{array} \quad (3.2)$$

där $\bar{c}_j \geq 0$ och alltså fås det minsta värdet på v , nämligen \bar{v}_0 , då man sätter variablerna y_1, y_2, \dots, y_{n-m} till noll.

Poängen här är att vi vill använda simplexmetoden för att lösa ett heltalsproblem, men som vi tidigare sett ligger optimumpunkterna till simplexmetoden i polytopens hörnpunkter. Dessa behöver dock inte vara heltal, så vad gör vi. Jo, vi använder Gomorys metod, som går ut på att konstruera nya hyperplan som skär i vår polytop. Vi vill skära bort våra gamla optimumpunkter, men aldrig skära bort en tillåten heltalspunkt. Vi kommer alltså att skära bort optimumlösningar och därefter köra en ny vända med simplexmetoden, där vi ignorerar kravet på heltalsvariablerna ända tills optimumvärdet består av våra heltalsvariabler.



Figur 3.1: De mörka punkterna är optimumpunkter som vi får när vi kör simplexmetoden och de ljusa punkterna är tillåtna heltalspunkter. Vi konstruerar nya hyperplan som skär i vår polytop tills en optimumpunkt sammanfaller med en heltalspunkt.

I resten av detta avsnitt kommer vi att genomföra vissa omskrivningar och variabelsubstitutioner, som till att börja med kan verka lite omständliga, men som kommer att underlätta lösningen av våra heltalsproblem.

Om vi utgår från att vi har fått fram vår lösning (3.2)

$$\begin{array}{rcl}
 \bar{c}_1 y_1 + \bar{c}_2 y_2 + \dots + \bar{c}_{n-m} y_{n-m} & & = v - \bar{v}_0 \\
 \bar{a}_{1,1} y_1 + \bar{a}_{1,2} y_2 + \dots + \bar{a}_{1,n-m} y_{n-m} & + y_{n-m+1} & = \bar{b}_1 \\
 \bar{a}_{2,1} y_1 + \bar{a}_{2,2} y_2 + \dots + \bar{a}_{2,n-m} y_{n-m} & + y_{n-m+2} & = \bar{b}_2 \\
 \vdots & \vdots & \vdots \\
 \bar{a}_{m,1} y_1 + \bar{a}_{m,2} y_2 + \dots + \bar{a}_{m,n-m} y_{n-m} & + y_n & = \bar{b}_m
 \end{array}$$

så gör vi följande variabelsubstitution för våra icke-basvariabler

$$y_i = \pi_i \quad i = 1, 2, \dots, n - m \quad (3.3)$$

och våra basvariabler uttrycker vi med hjälp av π_i

$$v = \bar{v}_0 + \sum_{i=1}^{n-m} \bar{c}_i \pi_i \quad (3.4)$$

$$y_{n-m+k} = \bar{b}_k - \sum_{i=1}^{n-m} \bar{a}_{k,i} \pi_i \quad k = 1, 2, \dots, m \quad (3.5)$$

Nu är alla variablerna y_1, y_2, \dots, y_n parametiserade med π_i .

Vi låter $\alpha_{i,j}$ beteckna koefficienten framför π_i , $\alpha_{0,j}$ får beteckna konstant-termen och $\alpha_{i,0}$ får beteckna koefficienterna framför v . Vårt ursprungliga problem blir nu följande i de nya variablerna

$$\begin{array}{r}
 v = \alpha_{0,0} + \alpha_{1,0} \pi_1 + \dots + \alpha_{\bar{m},0} \pi_{\bar{m}} \\
 y_1 = \alpha_{0,1} + \alpha_{1,1} \pi_1 + \dots + \alpha_{\bar{m},1} \pi_{\bar{m}} \\
 y_2 = \alpha_{0,2} + \alpha_{1,2} \pi_1 + \dots + \alpha_{\bar{m},2} \pi_{\bar{m}} \\
 \vdots \\
 y_n = \alpha_{0,n} + \alpha_{1,n} \pi_1 + \dots + \alpha_{\bar{m},n} \pi_{\bar{m}}
 \end{array} \quad (3.6)$$

Här ovan har vi satt $\bar{m} = n - m$. Vi kan nu notera, att om alla våra α_i -värden, som är kopplade till de y_j där $j \in \mathbf{J}$, är heltal, så är vi färdiga, dvs optimumpunkten består av heltal, då vi sätter $\pi_i = 0$. Om inte, kommer vi att skapa nya hyperplan tills vi är färdiga. Återigen, om vi struntar i heltalsvillkoret och vi vet att $y_j \geq 0$ så kan vi se problemet (3.6) på följande sätt

$$\alpha_{0,0} + \sum_{i=1}^{\bar{m}} \alpha_{i,0} \pi_i = v \quad (\min) \quad (3.7)$$

$$\alpha_{0,j} + \sum_{i=1}^{\bar{m}} \alpha_{i,j} \pi_i \geq 0 \quad j = 1, 2, \dots, n \quad (3.8)$$

Det duala problemet till (3.7)-(3.8) blir att maximera x_0 där $x_j \geq 0$ för $j \neq 0$ under följande villkor

$$\begin{aligned} x_0 + \alpha_{0,1}x_1 + \alpha_{0,2}x_2 + \dots + \alpha_{0,n}x_n &= \alpha_{0,0} \\ \alpha_{1,1}x_1 + \alpha_{1,2}x_2 + \dots + \alpha_{1,n}x_n &= \alpha_{1,0} \\ &\vdots \\ \alpha_{\bar{m},1}x_1 + \alpha_{\bar{m},2}x_2 + \dots + \alpha_{\bar{m},n}x_n &= \alpha_{\bar{m},0} \end{aligned} \quad (3.9)$$

Eftersom vi bara har parametriserat y_i i systemet (3.2), så är det ovan duala systemet ekvivalent med det duala systemet till (3.2) som vi vet har optimumvärdet \bar{v}_0 som är samma som α_0 i våra nya variabler. Detta värde får vi i (3.9) då vi sätter $x_j = 0$ för alla $j \neq 0$. Alltså har vi systemet i optimal form.

Vi avslutar detta uppbyggnadsavsnitt med att titta på hur vi kan beskriva heltalsdelen respektive decimaldelen av ett tal. Låt $[\alpha_j]^*$ vara det största heltalet $\leq \alpha_j$ och vi definerar nu f_j

$$f_j = \alpha_j - [\alpha_j]^*$$

som är den positiva decimaldelen av α_j . Vi definerar också komplementet, \bar{f}_j , till f_j som

$$\bar{f}_j = \begin{cases} 1 - f_j & \text{om } f_j > 0 \\ 0 & \text{om } f_j = 0 \end{cases}$$

Denna notation kommer vi att använda oss av i följande kapitel när vi ska konstruera nya hyperplan.

3.1 Rena heltalsproblem

Vi ska nu börja med att skapa nya hyperplan till problem, där alla våra y_i -variabler är heltal.

Sats 1. *Om y är en heltalsvariabel och $\pi_i \geq 0$ är parametrar relaterade på följande vis*

$$y = \alpha_0 + \alpha_1\pi_1 + \dots + \alpha_{\bar{m}}\pi_{\bar{m}} \quad (3.10)$$

där $\alpha_i \geq 0$, för $i \neq 0$. Då gäller följande olikhet

$$1 \leq f_0 + \alpha_1\pi_1 + \dots + \alpha_{\bar{m}}\pi_{\bar{m}} \quad (3.11)$$

för alla π_i som genererar heltalet y , men som inte satisfierar den optimala lösningen $\pi_i = 0$ för $i = 1, 2, \dots, \bar{m}$

Bevis. Eftersom $\alpha_i \geq 0$ och $\pi_i \geq 0$ så är $\min y \geq \alpha_0$. Men eftersom y :s möjliga värden är heltal, så gäller följande

$$y \geq [\alpha_0]^* + 1 \quad (3.12)$$

Om vi stoppar in denna olikhet i uttrycket för y (3.10) så får vi följande

$$\alpha_0 + \alpha_1\pi_1 + \dots + \alpha_{\bar{m}}\pi_{\bar{m}} \geq [\alpha_0]^* + 1$$

och vi utnyttjar definitionen $f_0 = \alpha_0 - [\alpha_0]^*$ och får olikheten (3.11) och beviset är färdigt. \square

Om vi i (3.12) skaffar en likhet med hjälp av y^* genom att skriva $y - y^* = [\alpha_0]^* + 1$ där $y^* \geq 0$ är ett heltal, så får vi en starkare variant av olikheten (3.11), nämligen följande

$$y^* = -\bar{f}_0 + \alpha_1\pi_1 + \dots + \alpha_{\bar{m}}\pi_{\bar{m}}$$

Vi har alltså skapat ett nytt hyperplan y^* som skär bort tidigare tillåtna optimumpunkter, men som inte tar bort en tillåten heltalspunkt.

Sats 2. Om y är en heltalsvariabel och om $\pi_i \geq 0$ också är heltalsvariabler som uppfyller följande villkor

$$y = \alpha_0 + \alpha_1\pi_1 + \dots + \alpha_{\bar{m}}\pi_{\bar{m}} \quad (3.13)$$

så gäller följande olikhet

$$1 \leq f_0 + f_1\pi_1 + \dots + f_{\bar{m}}\pi_{\bar{m}} \quad (3.14)$$

för alla π_i som genererar heltalet y men som inte uppfyller den optimala lösningen $\pi_i = 0$

Bevis. Substituera $\alpha_j = f_j + [\alpha_j]^*$ i uttrycket (3.13) vilket ger

$$y = f_0 + [\alpha_0]^* + (f_1 + [\alpha_1]^*)\pi_1 + \dots + (f_{\bar{m}} + [\alpha_{\bar{m}}]^*)\pi_{\bar{m}}$$

$$\iff$$

$$y - [\alpha_0]^* - \sum_{i=1}^{\bar{m}} [\alpha_i]^* \pi_i = f_0 + f_1 \pi_1 + \dots + f_{\bar{m}} \pi_{\bar{m}}$$

Vänsterledet är ett heltal och eftersom $f_j \geq 0$ kan vi tillämpa sats 1, varvid vi får olikheten (3.14) och vi är klara. \square

På samma sätt som ovan kan man använda sats 2 till att skapa ett nytt hyperplan som skär i vår polytop, men som inte skär bort tillåtna heltalspunkter

$$y^{**} = -\bar{f}_0 + f_1 \pi_1 + \dots + f_{\bar{m}} \pi_{\bar{m}}$$

där y^{**} är ett positivt heltal.

I sats 2 får α_i anta både positiva och negativa värden. Eftersom alla våra $y_i \geq 0$ är heltal, kommer alltid $\pi_i \geq 0$ vara ett heltal, eftersom $\pi_i = y_i$ för $i = 1, \dots, \bar{m}$. Alltså kommer vi alltid att kunna använda sats 2 på rena heltalsproblem för att skapa nya hyperplan.

När vi stoppar in ett nytt hyperplan, säg y^{**} , i vårt system (3.6) får vi följande

$$\begin{aligned} v &= \alpha_{0,0} + \alpha_{1,0}\pi_1 + \dots + \alpha_{\bar{m},0}\pi_{\bar{m}} \\ y_1 &= \alpha_{0,1} + \alpha_{1,1}\pi_1 + \dots + \alpha_{\bar{m},1}\pi_{\bar{m}} \\ y_2 &= \alpha_{0,2} + \alpha_{1,2}\pi_1 + \dots + \alpha_{\bar{m},2}\pi_{\bar{m}} \\ &\vdots \\ y_n &= \alpha_{0,n} + \alpha_{1,n}\pi_1 + \dots + \alpha_{\bar{m},n}\pi_{\bar{m}} \\ y^{**} &= -\bar{f}_0 + f_1 \pi_1 + \dots + f_{\bar{m}} \pi_{\bar{m}} \end{aligned} \tag{3.15}$$

Här ser vi, att $\pi_i = 0$ för alla i , inte längre är en tillåten lösning, eftersom y^{**} då blir negativ. Om vi istället tittar på det duala systemet till (3.15), så kommer det att innehålla en extra variabel jämfört med vårt gamla duala system (3.9), eftersom vi får ta hänsyn till det extra bivillkoret y^{**} . Om vi sätter $x_i = 0$ för alla $i \neq 0$ i det duala systemet, ser vi att vi får en tillåten lösning, men inte längre optimal pga $-\bar{f}_0$

$$\begin{aligned}
x_0 + \alpha_{0,1}x_1 + \alpha_{0,2}x_2 + \dots + \alpha_{0,n}x_n + (-\bar{f}_0)x_{n+1} &= \alpha_{0,0} \\
\alpha_{1,1}x_1 + \alpha_{1,2}x_2 + \dots + \alpha_{1,n}x_n + f_1x_{n+1} &= \alpha_{1,0} \\
&\vdots \\
\alpha_{\bar{m},1}x_1 + \alpha_{\bar{m},2}x_2 + \dots + \alpha_{\bar{m},n}x_n + f_{\bar{m}}x_{n+1} &= \alpha_{\bar{m},0}
\end{aligned} \tag{3.16}$$

Nu justerar vi det duala problemet så att det hamnar i optimum. Det var detta som var poängen med våra parametiseringar och variabelsubstitutioner. Det blir nämligen relativt enkelt att hitta ett optimum i systemet (3.16). Det som samtidigt händer i det primala systemet, är att ett av våra π_i elimineras, samtidigt som vi parametiserar y^{**} med $\pi_{\bar{m}+1}$. Detta kommer att bli tydligare när vi tittar på följande konkreta exempel.

Exempel 4 Vi ska nu använda metoden för att lösa följande problem. Hitta $y_i \geq 0$ som minimerar v och som uppfyller följande

$$\begin{aligned}
\frac{1}{4}y_1 &+ \frac{4}{3}y_3 &= v + \frac{7}{12} \\
2y_1 + y_2 + \frac{4}{3}y_3 &= \frac{13}{3} \\
\frac{1}{4}y_1 &+ \frac{3}{4}y_3 + y_4 &= \frac{9}{4}
\end{aligned}$$

där v och y_i är heltal.

Om vi relaxerar heltalsvillkoret så är problemet i optimal form. y_1 och y_3 är våra icke-basvariabler. Vi följer nu steg (3.3)-(3.5) och sätter $y_1 = \pi_1$ och $y_3 = \pi_3$ och uttrycker v , y_2 och y_4 med hjälp av π_1 och π_3 . Då fås

$$\begin{aligned}
v &= -\frac{7}{12} + \frac{1}{4}\pi_1 + \frac{4}{3}\pi_3 \\
y_1 &= \pi_1 \\
y_2 &= \frac{13}{3} - 2\pi_1 - \frac{4}{3}\pi_3 \\
y_3 &= \pi_3 \\
y_4 &= \frac{9}{4} - \frac{1}{2}\pi_1 - \frac{3}{4}\pi_3
\end{aligned} \tag{3.17}$$

Nu använder vi sats 2 och skapar ett nytt hyperplan på följande form

$$y^{**} = -\bar{f}_0 + f_1\pi_1 + \dots + f_{\bar{m}}\pi_{\bar{m}}$$

Om vi använder oss av den översta ekvationen i (3.17) så får vi

$$y_5 = -\frac{7}{12} + \frac{1}{4}\pi_1 + \frac{1}{3}\pi_3$$

Vårt ekvationsystem (3.17) får alltså ett nytt villkor y_5

$$\begin{aligned}
 v &= -\frac{7}{12} + \frac{1}{4}\pi_1 + \frac{4}{3}\pi_3 \\
 y_1 &= \pi_1 \\
 y_2 &= \frac{13}{3} - 2\pi_1 - \frac{4}{3}\pi_3 \\
 y_3 &= \pi_3 \\
 y_4 &= \frac{9}{4} - \frac{1}{2}\pi_1 - \frac{3}{4}\pi_3 \\
 y_5 &= -\frac{7}{12} + \frac{1}{4}\pi_1 + \frac{1}{3}\pi_3
 \end{aligned} \tag{3.18}$$

Nu tar vi fram det duala problemet till (3.18)

$$\begin{aligned}
 x_0 &+ \frac{13}{3}x_2 + \frac{9}{4}x_4 - \frac{7}{12}\mathbf{x}_5 = -\frac{7}{12} \\
 x_1 &- 2x_2 - \frac{1}{2}x_4 + \frac{1}{4}\mathbf{x}_5 = \frac{1}{4} \\
 &-\frac{4}{3}x_2 + x_3 - \frac{3}{4}x_4 + \frac{1}{3}\mathbf{x}_5 = \frac{4}{3}
 \end{aligned} \tag{3.19}$$

Vi ser här att y_5 -villkoret översätts till x_5 -variabeln i det duala systemet. Nu ser vi att $x_2, x_4, x_5 = 0$ är en tillåten punkt, men inte optimal pga $-\frac{7}{12}\mathbf{x}_5$. Det är alltså lätt att få det duala problemet i optimal form. Vi tar bara och eliminerar bort $-\frac{7}{12}\mathbf{x}_5$ genom att pivotera på $\frac{1}{4}\mathbf{x}_5$ i (3.19). Då fås

$$\begin{aligned}
 x_0 + \frac{7}{3}x_1 - \frac{1}{3}x_2 &+ \frac{13}{12}x_4 = 0 \\
 4x_1 - 8x_2 &- 2x_4 + x_5 = 1 \\
 -\frac{4}{3}x_1 + \frac{4}{3}\mathbf{x}_2 + x_3 - \frac{1}{12}x_4 &= 1
 \end{aligned} \tag{3.20}$$

Nu är det inte x_1 och x_3 som har ledande ettor som i (3.19), utan det är x_3 och x_5 . Detta motsvaras av att vi eliminerar bort π_1 från vårt primala system, vilket sker genom att sätta $\pi_5 = y_5 = -\frac{7}{12} + \frac{\pi_1}{4} + \frac{\pi_3}{3}$ i vårt primala problem, jämför med det som står med fet stil i (3.19), och låta $y_5 = \pi_5$ vara vår nya parameter. Vi utnyttjar att vi har $y_5 = \pi_5$ och skriver om (3.18)

$$\begin{aligned}
 v &= -\frac{7}{12} + \frac{1}{4}\pi_1 + \frac{4}{3}\pi_3 \\
 y_1 &= \pi_1 \\
 y_2 &= \frac{13}{3} - 2\pi_1 - \frac{4}{3}\pi_3 \\
 y_3 &= \pi_3 \\
 y_4 &= \frac{9}{4} - \frac{1}{2}\pi_1 - \frac{3}{4}\pi_3 \\
 0 &= -\frac{7}{12} + \frac{1}{4}\pi_1 + \frac{1}{3}\pi_3 - \pi_5
 \end{aligned} \tag{3.21}$$

och pivoterar $\frac{1}{4}\pi_1$. Sedan flyttar vi tillbaka π_5 och kallar den återigen för y_5 .

Då får vi

$$\begin{aligned}
 v &= \pi_5 + \pi_3 \\
 y_1 &= \frac{7}{3} + 4\pi_5 - \frac{4}{3}\pi_3 \\
 y_2 &= -\frac{1}{3} - 8\pi_5 + \frac{4}{3}\pi_3 \\
 y_3 &= \pi_3 \\
 y_4 &= \frac{13}{12} - 2\pi_5 - \frac{1}{12}\pi_3 \\
 y_5 &= \pi_5
 \end{aligned} \tag{3.22}$$

Nu har vi sett vad som händer i det primala systemet, så vi återgår till det duala systemet (3.20). Vi ser att vi fortfarande inte har en optimal lösning pga $-\frac{1}{3}x_2$. Vi pivoterar på $\frac{4}{3}\mathbf{x}_2$ och får

$$\begin{aligned}
 x_0 + 2x_1 &+ \frac{1}{4}x_3 + \frac{17}{16}x_4 &= \frac{1}{4} \\
 -4x_1 &+ 6x_3 - \frac{5}{2}x_4 + x_5 &= 7 \\
 -x_1 + x_2 + \frac{3}{4}x_3 - \frac{1}{16}x_4 &&= \frac{3}{4}
 \end{aligned} \tag{3.23}$$

På samma sätt som tidigare svarar detta mot att eliminera π_3 och ersätta med $\pi_2 = y_2 = -\frac{1}{3} - 8\pi_5 + \frac{4}{3}\pi_3$ i det primala systemet. Det primala systemet blir följande

$$\begin{aligned}
 v &= \frac{1}{4} + 7\pi_5 + \frac{3}{4}\pi_2 \\
 y_1 &= 2 - 4\pi_5 - \pi_2 \\
 y_2 &= \pi_2 \\
 y_3 &= \frac{1}{4} + 6\pi_5 + \frac{3}{4}\pi_2 \\
 y_4 &= \frac{17}{16} - \frac{5}{2}\pi_5 - \frac{1}{16}\pi_2 \\
 y_5 &= \pi_5
 \end{aligned} \tag{3.24}$$

Vårt duala problem (3.23) är i optimum, men vi ser att vårt primala system (3.24) inte uppfyller heltalsvillkoret. Vi använder sats 2 igen på $y_3 = \frac{1}{4} + 6\pi_5 + \frac{3}{4}\pi_2$ och får ett nytt hyperplan

$$y_6 = \frac{3}{4} + \frac{3}{4}\pi_2$$

Vi stoppar in detta villkor i (3.24) och tar fram det duala problemet

$$\begin{aligned}
 x_0 + 2x_1 &+ \frac{1}{4}x_3 + \frac{17}{16}x_4 - \frac{3}{4}x_6 &= \frac{1}{4} \\
 -4x_1 &+ 6x_3 - \frac{5}{2}x_4 + x_5 &= 7 \\
 -x_1 + x_2 + \frac{3}{4}x_3 - \frac{1}{16}x_4 + \frac{3}{4}x_6 &&= \frac{3}{4}
 \end{aligned}$$

Vi pivoterar på $\frac{3}{4}\mathbf{x}_6$ och får

$$\begin{aligned}
 x_0 &+ x_1 &+ x_2 &+ x_3 &+ x_4 &&= 1 \\
 -4x_1 &&&+ 6x_3 &- \frac{5}{2}x_4 &+ x_5 &= 7 \\
 -\frac{4}{3}x_1 &+ \frac{4}{3}x_2 &+ x_3 &- \frac{1}{12}x_4 &&+ x_6 &= 1
 \end{aligned}$$

där vi har eliminerat π_2 och ersatt med $\pi_6 = y_6 = \frac{3}{4} + \frac{3}{4}\pi_2$ i det primala systemet. Nu är vi färdiga och vårt primala system ser ut så här

$$\begin{aligned} v &= 1 + 7\pi_5 + \pi_6 \\ y_1 &= 1 - 4\pi_5 - \frac{4}{3}\pi_6 \\ y_2 &= 1 + \frac{4}{3}\pi_6 \\ y_3 &= 1 + 6\pi_5 + \pi_6 \\ y_4 &= 1 - \frac{5}{2}\pi_5 - \frac{1}{12}\pi_6 \\ y_5 &= \pi_5 \\ y_6 &= \pi_6 \end{aligned}$$

där den optimala lösningen fås om $\pi_5 = \pi_6 = 0$.

3.2 Blandade heltalsproblem

För problem som innehåller både heltal och decimaltal måste vi kunna skapa nya olikheter, då koefficienten framför decimaltalet kan anta både positiva och negativa värden. Vi delar upp y på följande sätt

$$y = \alpha_0 + \sum_{i=1}^{\bar{m}} \alpha_i \pi_i = \alpha_0 + P - N \quad (3.25)$$

Där y är ett heltal, P och $-N$ är summorna av de positiva respektive negativa termerna och där α_0 inte är ett heltal¹. För vissa värden på π_i kommer $P - N$ vara antingen positivt eller negativt. Vi delar in det i två fall.

Fall 1 $P - N \geq 0$ och eftersom $y = \alpha_0 + P - N$ är ett heltal, så måste $f_0 + P - N$ också vara ett heltal och $f_0 + P - N > 0$. Alltså måste följande gälla

$$\begin{aligned} 1 &\leq f_0 + P - N \Leftrightarrow \bar{f}_0 \leq P - N \\ 1 &\leq \frac{1}{f_0}P - \frac{1}{f_0}N \leq \frac{1}{f_0}P + \frac{1}{f_0}N \end{aligned}$$

Fall 2 inträffar för de värden på π_i som uppfyller $N - P \geq 0$. Då är $-y = \alpha_0 + N - P$ ett heltal och följande gäller

$$1 \leq \bar{f}_0 + N - P \Leftrightarrow f_0 \leq N - P$$

¹Om det vore ett heltal så vore vi färdiga.

$$1 \leq \frac{1}{f_0}N - \frac{1}{f_0}P \leq \frac{1}{f_0}N + \frac{1}{f_0}P$$

Vi ser alltså att oberoende av värdena på koefficienterna framför π_i så gäller

$$1 \leq \frac{1}{f_0}P + \frac{1}{f_0}N \quad (3.26)$$

Sats 3. Om y är ett heltal och $\pi_i \geq 0$ uppfyller följande

$$y = \alpha_0 + \sum_{i=1}^{\bar{m}} \alpha_i \pi_i \quad (3.27)$$

så gäller följande olikhet

$$1 \leq \frac{1}{f_0} \left(\sum_{i \in I_1} f_i \pi_i + \sum_{i \in I_2} \alpha_i \pi_i \right) + \frac{1}{f_0} \left(\sum_{i \in I_3} \bar{f}_i \pi_i - \sum_{i \in I_4} \alpha_i \pi_i \right) \quad (3.28)$$

för alla π_i som genererar heltalet y , men som inte uppfyller den optimala lösningen som fås då $\pi_i = 0$ och där

$i \in I_1$ om $f_i \leq \bar{f}_0$ och π_i heltal

$i \in I_2$ om $\alpha_i > 0$ och π_i decimaltal

$i \in I_3$ om $f_i < f_0$ och π_i heltal

$i \in I_4$ om $\alpha_i < 0$ och π_i decimaltal

Bevis. Om π_i är ett heltal, ersätt α_i i (3.27) med $[\alpha_i]^* + f_i$ eller med $[\alpha_i]^* + 1 - \bar{f}_i$ beroende på om $f_i < \bar{f}_0$ eller $\bar{f}_i < f_0$. Då fås

$$y = \alpha_0 + \left(\sum_{i \in I_1} ([\alpha_i]^* + f_i) \pi_i + \sum_{i \in I_2} \alpha_i \pi_i \right) + \left(\sum_{i \in I_3} ([\alpha_i]^* + 1 - \bar{f}_i) \pi_i + \sum_{i \in I_4} \alpha_i \pi_i \right)$$

där mängderna I_i är samma som ovan. Nu flyttar vi över alla heltal till vänster sida och kallar detta nya heltal för y'

$$\begin{aligned} y' &= \alpha_0 + \left(\sum_{i \in I_1} f_i \pi_i + \sum_{i \in I_2} \alpha_i \pi_i \right) + \left(\sum_{i \in I_3} -\bar{f}_i \pi_i + \sum_{i \in I_4} \alpha_i \pi_i \right) \Leftrightarrow \\ y' &= \alpha_0 + \underbrace{\left(\sum_{i \in I_1} f_i \pi_i + \sum_{i \in I_2} \alpha_i \pi_i \right)}_P - \underbrace{\left(\sum_{i \in I_3} \bar{f}_i \pi_i - \sum_{i \in I_4} \alpha_i \pi_i \right)}_N \end{aligned}$$

Vi ser nu att detta uttryck är på samma form som (3.25) och där har vi noterat att olikheten (3.26) gäller. Stoppa in uttrycken för P och N i olikheten

$$1 \leq \frac{1}{f_0}P + \frac{1}{f_0}N$$

varvid vi får vår sökta olikhet (3.28) och beviset är klart. \square

Vi gör nu som tidigare och tar fram ett nytt hyperplan $y^{***} \geq 0$.

$$y^{***} = -1 \frac{1}{\bar{f}_0} \left(\sum_{i \in I_1} f_i \pi_i + \sum_{i \in I_2} \alpha_i \pi_i \right) + \frac{1}{f_0} \left(\sum_{i \in I_3} \bar{f}_i \pi_i - \sum_{i \in I_4} \alpha_i \pi_i \right)$$

Tyvärr är inte y^{***} ett heltal, vilket gör det besvärligt för oss när vi ska lösa problem som i exemplet ovan. Vi ska nu se vad som krävs för att få y^{***} till ett heltal.

Ordna om i uttrycket $y = \alpha_0 + \sum_{i=1}^{\bar{m}} \alpha_i \pi_i$ så att $i = 1, \dots, k$ är index på heltalsvariablerna, och låt $i > k$ vara index för decimalvariablerna. Låt nu

$$\bar{y} = \sum_{i>k} \alpha_i \pi_i - \pi^* \quad \text{där} \quad \pi^* \in [0, 1]$$

Här är \bar{y} en heltalsvariabel och π^* är \bar{y} :s positiva decimaldel. Det nya villkoret som vi vill ha är då

$$y^{****} = -\bar{f}_0 + \sum_{i=1}^k f_i \pi_i + \pi^* \quad \text{där} \quad \pi^* \in [0, 1]$$

Nu har vi fått problemet på vår önskade heltalsform. Tyvärr har vi också fått en ny begränsad variabel π^* . Detta gör att det inte längre blir lika effektivt att gå över till det duala problemet, utan man kanske ska stanna i det primala.

Kapitel 4

Ickelinjär optimering

Teorin för den icke-linjära optimeringen, som jag presenterar här, är främst tagen från [4]. Ett generellt icke-linjärt programmeringsproblem ser ut på följande sätt och kallas för det *primala* problemet

$$\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{då} & g_i(\mathbf{x}) \leq 0 \quad \text{för } i = 1, \dots, m \\ & h_i(\mathbf{x}) = 0 \quad \text{för } i = 1, \dots, l \\ & \mathbf{x} \in \mathbf{X} \end{array}$$

där funktionerna f, g och h kan vara icke-linjära. I fortsättningen kommer vi att använda oss av vektornotation, men vi kan ha i minnet, att vi har m stycken g -funktioner och l stycken h -funktioner. Vi kommer strax att se att vi kan använda satsen för den icke-linjära optimeringen för att lösa heltalsproblem.

4.1 Den duala funktionen

Som vi sett tidigare, när vi ägnade oss åt linjära problem, så fanns det ett dualt problem kopplat till vårt primala problem. Det finns liknande samband inom den icke-linjära optimeringen. Om vi tittar på vårt icke-linjära problem ovan och kallar det för vårt primala problem, så finns det ett dualt problem relaterat till vårt primala problem på följande vis

$$\begin{array}{ll}
\min & f(\mathbf{x}) \\
\text{(P) då} & \mathbf{g}(\mathbf{x}) \leq \mathbf{0} \\
& \mathbf{h}(\mathbf{x}) = \mathbf{0} \\
& \mathbf{x} \in \mathbf{X}
\end{array}
\qquad
\begin{array}{ll}
\max & \theta(\mathbf{u}, \mathbf{v}) \\
\text{(D) då} & \mathbf{u} \geq \mathbf{0}
\end{array}$$

där $\theta(\mathbf{u}, \mathbf{v}) = \inf\{f(\mathbf{x}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}) + \mathbf{v}^t \mathbf{h}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$

Vi säger att vi har *Lagrangerelaxerat* bivillkoren $\mathbf{g}(\mathbf{x})$ och $\mathbf{h}(\mathbf{x})$. \mathbf{u} och \mathbf{v} kallas för *Lagrangemultiplikatorer*.

Exempel 5 Vi ska nu se hur vi kan Lagrangerelaxera bivillkoren i ett linjärt optimeringsproblem för att få fram ett dualt problem på samma form som i avsnitt 2.3. Anta att vi har ett primalt problem som ser ut så här

$$\begin{array}{ll}
\min & c_1 x_1 + c_2 x_2 \\
\text{då} & b_1 + a_{1,1} x_1 + a_{1,2} x_2 \leq 0 \\
& b_2 + a_{2,1} x_1 + a_{2,2} x_2 = 0 \\
& x_1, x_2 \geq 0
\end{array}$$

Det duala problemet blir då att

$$\max_{u \geq 0} \theta(u, v)$$

där

$$\begin{aligned}
\theta(u, v) = \inf_{x_1, x_2} & \left\{ (c_1 x_1 + c_2 x_2) + u(b_1 + a_{1,1} x_1 + a_{1,2} x_2) \right. \\
& \left. + v(b_2 + a_{2,1} x_1 + a_{2,2} x_2) : x_1, x_2 \geq 0 \right\}
\end{aligned}$$

θ funktionen kan nu skrivas

$$\begin{aligned}
\theta(u, v) = b_1 u + b_2 v + \inf_{x_1} & \{x_1(c_1 + a_{1,1} u + a_{2,1} v) : x_1 \geq 0\} \\
& + \inf_{x_2} \{x_2(c_2 + a_{1,2} u + a_{2,2} v) : x_2 \geq 0\}
\end{aligned}$$

Vi ser hur θ separeras i två subproblem

$$\inf_{x_1} \{x_1(c_1 + a_{1,1} u + a_{2,1} v) : x_1 \geq 0\}$$

och

$$\inf_{x_2} \{x_2(c_2 + a_{1,2} u + a_{2,2} v) : x_2 \geq 0\}$$

Det är dessa två som kommer att bli våra bivillkor, när vi ska framställa det duala problemet. Om vi tittar på

$$h(x_2) = \inf_{x_2} \{(x_2(c_2 + a_{1,2}u + a_{2,2}v) : x_2 \geq 0)\}$$

så ser vi att $h(x_2)$ kommer att anta värdet 0 eller $-\infty$.

$$h(x_2) = \begin{cases} 0 & \text{om } c_2 + a_{1,2}u + a_{2,2}v > 0 \\ 0 & \text{om } c_2 + a_{1,2}u + a_{2,2}v = 0 \\ -\infty & \text{om } c_2 + a_{1,2}u + a_{2,2}v < 0 \end{cases}$$

Vi ser att om vi vill maximera θ så ska vi inte välja Lagrangemultiplikatorerna u och v på så sätt att uttrycket $c_2 + a_{1,2}u + a_{2,2}v$ blir negativt. På samma sätt gäller det förstås att $c_1 + a_{1,1}u + a_{1,2}v$ inte får vara negativt, eftersom då skulle $h(x_1)$ bli obegränsat negativt. Alltså kan vi ställa upp problemet med att maximera $\theta(u, v)$ på följande sätt

$$\begin{aligned} \max \quad & b_1u + b_2v \\ \text{då} \quad & c_1 + a_{1,1}u + a_{2,1}v \geq 0 \\ & c_2 + a_{1,2}u + a_{2,2}v \geq 0 \\ & u \geq 0 \end{aligned}$$

Här ovan har vi ett problem på samma form som det duala problemet i avsnitt 2.3.

Frågan är hur det primala problemet och det duala problemet hänger ihop. Svaret på den frågan ges av den *Starka dualitetssatsen*¹. Den visar att om målfunktionen och definitionsmängden är konvexa i det primala problemet, så har det primala och duala problemet samma optimum. Vi kommer inte att visa denna sats eftersom heltalsproblem inte är konvexa. Frågan blir då hur det primala och duala problemet hänger ihop, om vi inte utgår från en konvex mängd. Svaret ges i följande sats.

Sats 4 (Svaga dualitetssatsen). *Låt \mathbf{x} vara en tillåten lösning till det primala problemet, dvs att $\mathbf{x} \in \mathbf{X}$, $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ och $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. Anta också att (\mathbf{u}, \mathbf{v}) är en tillåten lösning till det duala problemet, dvs $\mathbf{u} \geq \mathbf{0}$. I såfall är $f(\mathbf{x}) \geq \theta(\mathbf{u}, \mathbf{v})$.*

Bevis.

$$\theta(\mathbf{u}, \mathbf{v}) = \inf \{f(\mathbf{y}) + \mathbf{u}^t \mathbf{g}(\mathbf{y}) + \mathbf{v}^t \mathbf{h}(\mathbf{y}) : \mathbf{y} \in \mathbf{X}\}$$

¹Se Non Linear Programming

$$\leq f(\mathbf{x}) + \mathbf{u}^t \mathbf{g}(\mathbf{x}) + \mathbf{v}^t \mathbf{h}(\mathbf{x}) \leq f(\mathbf{x})$$

Där vi i sista olikheten har använt oss av, att vi vet att $\mathbf{u} \geq \mathbf{0}$, $\mathbf{g}(\mathbf{x}) \leq \mathbf{0}$ och $\mathbf{h}(\mathbf{x}) = \mathbf{0}$. \square

Alltså vet vi, att om vi lyckas lösa vårt duala problem, får vi en undre begränsning till vårt primala problem. När inte det primala och det duala problemet har samma optimumvärde, säger man att ett *dualitetsgap* har uppstått.

Vi ska nu titta på vissa egenskaper hos den duala funktionen $\theta(\mathbf{u}, \mathbf{v})$. För att att slippa skriva så mycket, kommer vi att för bekvämlighets skull kombinera vektorerna \mathbf{u} och \mathbf{v} som vektorn \mathbf{w} samt kombinera funktionerna \mathbf{g} och \mathbf{h} som $\boldsymbol{\mu}$.

$$\mathbf{w} = \begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \quad \boldsymbol{\mu} = \begin{pmatrix} \mathbf{g} \\ \mathbf{h} \end{pmatrix}$$

Vi antar också att mängden \mathbf{X} är kompakt. Anledningen till att vi är så intresserade av det duala problemet är, att det visar sig att den duala funktionen är konkav².

Sats 5. *Låt \mathbf{X} vara en icke-tom mängd i \mathbf{R}^n , och låt $f : \mathbf{R}^n \rightarrow \mathbf{R}$ och $\boldsymbol{\mu} : \mathbf{R}^n \rightarrow \mathbf{R}^{m+l}$ vara kontinuerliga funktioner. I så fall är*

$$\theta(\mathbf{w}) = \inf\{f(\mathbf{x}) + \mathbf{w}^t \boldsymbol{\mu}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}$$

konkav på \mathbf{R}^{m+l} .

Bevis. Eftersom f och $\boldsymbol{\mu}$ är kontinuerliga funktioner och \mathbf{X} är kompakt, så är θ ändlig på hela \mathbf{R}^{m+l} . Låt $\mathbf{w}_1, \mathbf{w}_2 \in \mathbf{R}^{m+l}$ och låt $\lambda \in (0, 1)$. I så fall gäller följande

$$\begin{aligned} \theta[\lambda \mathbf{w}_1 + (1 - \lambda) \mathbf{w}_2] &= \inf\{f(\mathbf{x}) + [\lambda \mathbf{w}_1 + (1 - \lambda) \mathbf{w}_2]^t \boldsymbol{\mu}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \\ &= \inf\{\lambda [f(\mathbf{x}) + \mathbf{w}_1^t \boldsymbol{\mu}(\mathbf{x})] + (1 - \lambda) [f(\mathbf{x}) + \mathbf{w}_2^t \boldsymbol{\mu}(\mathbf{x})] : \mathbf{x} \in \mathbf{X}\} \\ &\geq \lambda \inf\{f(\mathbf{x}) + \mathbf{w}_1^t \boldsymbol{\mu}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \end{aligned}$$

²Kom ihåg att om en funktion f är konkav så är $-f$ konvex.

$$\begin{aligned}
&+(1-\lambda)\inf\{f(\mathbf{x})+\mathbf{w}_2^t\boldsymbol{\mu}(\mathbf{x}):\mathbf{x}\in\mathbf{X}\} \\
&=\lambda\theta(\mathbf{w}_1)+(1-\lambda)\theta(\mathbf{w}_2)
\end{aligned}$$

I beviset ovan har vi utnyttjat den duala funktionens definition. Vi ser att θ är konkav enligt definition 2. \square

Vi vet nu att θ är konkav, utan att det primala problemet är konvext. Bivillkoret $\mathbf{u}\geq\mathbf{0}$ är uppenbarligen en konvex mängd. Vi vet också att varje lokalt optimum för en konkav funktion också är ett globalt optimum. Konkret betyder det att vår duala funktion har ett optimum.

Eftersom vårt primala problem inte är konvext, så behöver inte det primala problemet och det duala problemet ha samma optimumvärde, utan det kan uppstå ett dualitetsgap. Vår duala funktion är inte explicit given. För att få reda på θ 's värde i olika punkter måste vi för varje punkt lösa ett suboptimeringsproblem. Poängen är ändå att det duala problemet kommer att vara betydligt enklare att lösa än det primala.

Lagrangerelaxering av heltalsproblem

Vi ska nu se hur vi kan använda ickelinjär teori för att lösa linjära heltalsproblem. Man kan nämligen Lagrangerelaxera en del av bivillkoren och på så vis skaffa sig olika subproblem som är enklare att lösa. Detta är fram för allt användbart när man löser stora problem. Vi ska illustrera hur man kan relaxera olika bivillkor samt hur man kan tolka Lagrangemultiplikatorerna i följande exempel.

Exempel 6 Detta problem är hämtat ur [1] och kallas för det *Generaliserande tillordningsproblemet* och formuleras

$$\begin{array}{ll}
\min & \sum_{i=1}^m\sum_{j=1}^nc_{i,j}x_{i,j} \\
\text{då} & \sum_{i=1}^mx_{i,j}=1 \quad j=1,\dots,n \quad (1) \\
& \sum_{j=1}^na_{i,j}x_{i,j}\leq b_i \quad i=1,\dots,m \quad (2) \\
& x_{i,j}\in\{0,1\} \quad i=1,\dots,m;j=1,\dots,n \quad (3)
\end{array}$$

Problemet har följande tolkning. Det ska utföras n stycken jobb på m maskiner. När jobb j utförs på maskin i uppstår en kostnad $c_{i,j}$. Målet är att minimera denna kostnad. Den tid det tar att utföra jobb j på maskin i är

$a_{i,j}$ och den totala tillgängliga tiden för maskin i är b_i

Villkor (1) säger att varje maskin ska tilldelas exakt ett jobb och villkor (2) utgör ett kapacitetsvillkor. Vi ska nu gå igenom tre olika fall på hur man kan relaxera detta problem för att få en undre uppskattning av problemet.

Fall 1 Relaxera heltalsvillkoren (3) och lös problemet med simplexmetoden.

Fall 2 Lagrangerelaxera villkor (1). Vi får följande Lagrangesubproblem

$$\begin{aligned}\theta_1(\mathbf{v}) &= \min \left(\sum_{i=1}^m \sum_{j=1}^n c_{i,j} x_{i,j} + \sum_{j=1}^n v_j \left(1 - \sum_{i=1}^m x_{i,j}\right) \right) \\ &= \min \left(\sum_{i=1}^m \sum_{j=1}^n (c_{i,j} - v_j) x_{i,j} + \sum_{j=1}^n v_j \right) \\ \text{då} \quad &\sum_{j=1}^n a_{i,j} x_{i,j} \leq b_i \quad i = 1, \dots, m \\ &x_{i,j} \in \{0, 1\} \quad i = 1, \dots, m; \quad j = 1, \dots, n\end{aligned}$$

Vi ser här att problemet separeras i m stycken olika delproblem, ett problem för varje maskin. Problemet för maskin i kan formuleras

$$\begin{aligned}\min \quad &\sum_{j=1}^n (c_{i,j} - v_j) x_{i,j} \\ \text{då} \quad &\sum_{j=1}^n a_{i,j} x_{i,j} \leq b_i \\ &x_{i,j} \in \{0, 1\} \quad j = 1, \dots, n\end{aligned}$$

Den här typen av problem kallas för *kappsäcksproblem*. Man ska välja bland olika objekt och lägga dem i sin kappsäck. Multiplikatorn v_j kan tolkas som den vinst som uppstår om jobb j utförs. Frågan blir, hur ska man välja multiplikatorn v_j . Om man väljer v_j för liten blir $(c_{i,j} - v_j) > 0$ och $x_{i,j} = 0$. Väljs v_j tillräckligt stor blir $x_{i,j} = 1$.

Vi provar nu olika värden på multiplikatorerna för att hitta optimum. Anta

att vi har provat k olika värden på multiplikatorerna $v_j^{(k)}$. Då kan två olika fall inträffa i (1)

$$1 - \sum_{i=1}^m x_{i,j} = 0$$

Det första är att $v_j^{(k)}$ för vissa jobb j kommer att vara för liten och vi får

$$1 - \sum_{i=1}^m x_{i,j}^{(k)} = 1 > 0$$

vilket betyder att jobb j inte har tilldelats någon maskin. Slutsatsen är att $v_j^{(k+1)}$ måste öka.

Det andra som kan inträffa är att för vissa jobb kan v_j^k vara för stor

$$1 - \sum_{i=1}^m x_{i,j}^{(k)} = 1 < 0$$

Detta betyder att jobb j har tilldelats flera maskiner och v_j^{k+1} måste minska.

Fall 3 Nu Lagrangerrelaxerar vi bivillkor (2)

$$\begin{aligned} \theta_2(\mathbf{u}) &= \min \left(\sum_{i=1}^m \sum_{j=1}^n c_{i,j} x_{i,j} + \sum_{i=1}^m u_i \left(\sum_{j=1}^n a_{i,j} x_{i,j} - b_i \right) \right) \\ &= \min \left(\sum_{i=1}^m \sum_{j=1}^n (c_{i,j} + u_i a_{i,j}) x_{i,j} - \sum_{i=1}^m b_i u_i \right) \end{aligned}$$

$$\text{då} \quad \sum_{i=1}^m x_{i,j} = 1 \quad j = 1, \dots, n$$

$$x_{i,j} \in \{0, 1\} \quad i = 1, \dots, m; \quad j = 1, \dots, n$$

Här är multiplikatorerna $u_i \geq 0$ för alla $i = 1, \dots, m$. Detta problem separeras i n stycken subproblem, ett för varje jobb. Problemet för ett givet jobb j blir nu

$$\begin{aligned} \min \left(\sum_{i=1}^m (c_{i,j} + u_i a_{i,j}) \right) \\ \text{då} \quad \sum_{i=1}^m x_{i,j} = 1 \end{aligned}$$

$$x_{i,j} \in \{0, 1\} \quad i = 1, \dots, m$$

Multiplikatorns tolkning blir nu priset för att utnyttja maskin i . Man löser detta genom att tilldela varje jobb j till den maskin som är billigast. Om vi nu provar olika värden på u_i som ovan, betecknar vi det k :te värdet på u_i med $u_i^{(k)}$, och ser hur de påverkar (2)

$$\sum_{j=1}^n a_{i,j} x_{i,j} - b_i \leq 0$$

Om vi antar att $u_i^{(k)}$ är satt för högt kommer det att leda till att för få jobb tilldelas maskin i alltså att

$$\sum_{j=1}^n a_{i,j} x_{i,j}^{(k)} - b_i < 0$$

och vi bör därför prova med ett lägre värde på multiplikatorn $u_i^{(k+1)}$. Fast om vi sätter ett för lågt värde kommer för många jobb att tilldelas maskin i

$$\sum_{j=1}^n a_{i,j} x_{i,j}^{(k+1)} - b_i > 0$$

Det gäller alltså att Lagrangerelaxera på ett smart sätt så att man får ett enklare problem att lösa. I exemplet ovan är fall 3 enklare att lösa än fall 2.

4.2 Subgradient

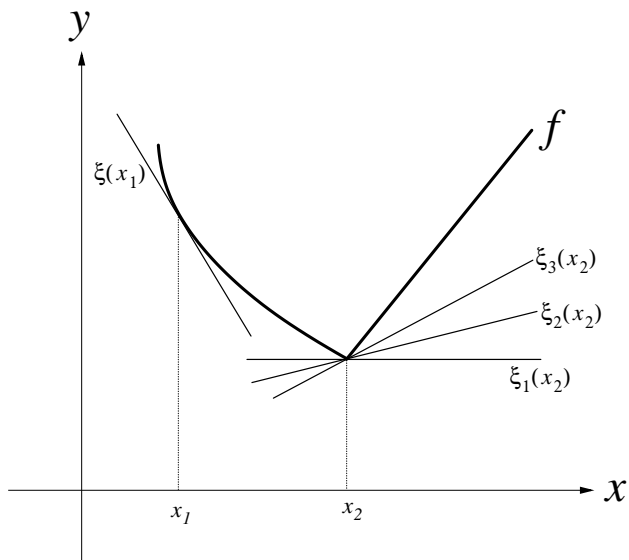
Vi är nu intresserade av att veta hur vi ska flytta oss för att hamna i den duala funktionens optimum. Man skulle kunna tro att man ska gå i gradientens riktning, men det går inte. Anledningen är att den duala funktionen inte behöver vara differentierbar överallt. Vi introducerar nu begreppet subgradient.

Definition 5. Låt \mathbf{K} vara en icke-tom konvex mängd i \mathbb{R}^n , och låt $f : \mathbf{K} \rightarrow \mathbb{R}$ vara en konvex funktion. Då kallas $\boldsymbol{\xi}$ för funktionen f :s subgradient i punkten $\bar{\mathbf{x}} \in \mathbf{K}$ om

$$f(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \boldsymbol{\xi}^t(\bar{\mathbf{x}} - \mathbf{x}) \quad \text{för alla } \mathbf{x} \in \mathbf{K}$$

Eftersom vi är intresserade av den duala funktionen, som är en konkav funktion, så gäller det att om f är en konvex funktion och allt annat samma som i definition 5 så är ξ funktionens subgradient i punkten $\bar{\mathbf{x}}$ om

$$f(\mathbf{x}) \leq f(\bar{\mathbf{x}}) + \xi^t(\bar{\mathbf{x}} - \mathbf{x}) \quad \text{för alla } \mathbf{x} \in \mathbf{K}$$



Figur 4.1: Punkten x_1 har en subgradient, medan punkten x_2 har flera subgradienter.

Om det bara finns en subgradient, så är den ekvivalent med funktionens gradient. Men är funktionen inte differentierbar, så kan vissa punkter innehålla flera subgradienter. Mängden av f 's alla subgradienter i punkten $\bar{\mathbf{x}}$ kallas för *subdifferential* och betecknas $\partial f(\bar{\mathbf{x}})$. $\partial f(\bar{\mathbf{x}})$ är en konvex mängd enligt definitionen ovan.

Vi introducerar nu mängden $\mathbf{X}(\mathbf{w})$ som består av de optimala lösningarna till de duala subproblemen.

$$\mathbf{X}(\mathbf{w}) = \{\mathbf{y} : \mathbf{y} \text{ minimerar } f(\mathbf{x}) + \mathbf{w}^t \boldsymbol{\mu}(\mathbf{x}) \text{ över } \mathbf{x} \in \mathbf{X}\}$$

Vi börjar med att visa att varje dualfunktion har minst en subgradient, nämligen $\boldsymbol{\mu}$.

Sats 6. Låt \mathbf{X} vara en icke-tom kompakt mängd i \mathbb{R}^n och låt $f : \mathbb{R}^n \rightarrow \mathbb{R}$ och $\boldsymbol{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ vara kontinuerliga funktioner, sådana att för något

$\bar{\mathbf{w}} \in \mathbb{R}^{m+l}$ så är inte $\mathbf{X}(\bar{\mathbf{w}})$ tom. Om $\bar{\mathbf{x}} \in \mathbf{X}(\bar{\mathbf{w}})$ så är $\boldsymbol{\mu}(\bar{\mathbf{x}})$ en subgradient till θ i punkten $\bar{\mathbf{w}}$.

Bevis. Låt $\bar{\mathbf{w}} \in \mathbb{R}^{m+l}$ och $\bar{\mathbf{x}} \in \mathbf{X}(\bar{\mathbf{w}})$. Eftersom f och $\boldsymbol{\mu}$ är kontinuerliga funktioner och \mathbf{X} är kompakt samt $\boldsymbol{\mu}(\bar{\mathbf{w}}) \neq \emptyset$ för alla $\bar{\mathbf{w}} \in \mathbb{R}^{m+l}$ så gäller

$$\begin{aligned}\theta(\mathbf{w}) &= \inf\{f(\mathbf{x}) + \mathbf{w}^t \boldsymbol{\mu}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\} \leq f(\bar{\mathbf{x}}) + \mathbf{w}^t \boldsymbol{\mu}(\bar{\mathbf{x}}) \\ &= f(\bar{\mathbf{x}}) + (\mathbf{w} - \bar{\mathbf{w}})^t \boldsymbol{\mu}(\bar{\mathbf{x}}) + \bar{\mathbf{w}}^t \boldsymbol{\mu}(\bar{\mathbf{x}}) \\ &= \theta(\bar{\mathbf{x}}) + (\mathbf{w} - \bar{\mathbf{w}})^t \boldsymbol{\mu}(\bar{\mathbf{x}})\end{aligned}$$

Alltså har vi visat att $\boldsymbol{\mu}(\bar{\mathbf{x}})$ är en subgradient till θ i punkten $\bar{\mathbf{w}}$ enligt definition 5. □

Vi vet dock inte hur subgradienterna kan visa vägen till den optimala punkten, utan vi behöver introducera ytterliggare ett berepp, nämligen riktningsderivata.

Definition 6. Låt \mathbf{M} vara en icke-tom mängd i \mathbb{R}^n och låt $f : \mathbf{M} \rightarrow \mathbb{R}$. Låt $\bar{\mathbf{x}} \in \mathbf{M}$ och \mathbf{d} vara en nollskild vektor sådan att $\bar{\mathbf{x}} + \lambda \mathbf{d} \in \mathbf{M}$ för $\lambda > 0$. Riktningensderivatan till f i punkten $\bar{\mathbf{x}}$ längs vektorn \mathbf{d} , betecknas som $f'(\bar{\mathbf{x}}; \mathbf{d})$ och definieras genom följande gränsvärde, om det existerar

$$f'(\bar{\mathbf{x}}; \mathbf{d}) = \lim_{\lambda \rightarrow 0^+} \frac{f(\bar{\mathbf{x}} + \lambda \mathbf{d}) - f(\bar{\mathbf{x}})}{\lambda}$$

Vi ser att riktningsderivatan ger oss information om hur rörelser påverkar funktionsvärdet.

Nu kommer en sats och ett corollary, som kopplar ihop begreppen subgradient och riktningsderivata.

Sats 7. Låt \mathbf{X} vara en icke-tom mängd i \mathbb{R}^n och låt $f : \mathbb{R}^n \rightarrow \mathbb{R}$ och $\boldsymbol{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ vara kontinuerliga funktioner. Låt $\bar{\mathbf{w}}, \mathbf{d} \in \mathbb{R}^{m+l}$, då uppfyller riktningsderivatan till θ i punkten $\bar{\mathbf{w}}$ med riktningen \mathbf{d} följande

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) \geq \mathbf{d}^t \boldsymbol{\mu}(\bar{\mathbf{x}}) \quad \text{för något } \bar{\mathbf{x}} \in \mathbf{X}$$

Bevis. Betrakta $\bar{\mathbf{w}} + \lambda_k \mathbf{d}$ när $\lambda_k \rightarrow 0^+$. För varje k existerar det ett $\mathbf{x}_k \in \mathbf{X}(\bar{\mathbf{w}} + \lambda_k \mathbf{d})$ och eftersom \mathbf{X} är kompakt så finns det konvergenta delsekvenser $\{\mathbf{x}_k\}_{\mathcal{K}}$ med gränsväde $\bar{\mathbf{x}} \in \mathbf{X}$. Givet $\mathbf{x} \in \mathbf{X}$ så gäller följande

$$f(\mathbf{x}) + (\bar{\mathbf{w}} + \lambda_k \mathbf{d})^t \boldsymbol{\mu}(\mathbf{x}) \geq f(\mathbf{x}_k) + (\bar{\mathbf{w}} + \lambda_k \mathbf{d})^t \boldsymbol{\mu}(\mathbf{x}_k) \quad (4.1)$$

för varje $k \in \mathcal{K}$. Olikheten i (4.1) följer av att vi vet att $\mathbf{x}_k \in \mathbf{X}(\bar{\mathbf{w}} + \lambda_k \mathbf{d})$ minimerar funktionen $f(\mathbf{x}) + (\bar{\mathbf{w}} + \lambda_k \mathbf{d})^t \boldsymbol{\mu}(\mathbf{x})$ för alla $\mathbf{x} \in \mathbf{X}$. Om vi låter $k \rightarrow \infty$, går $\lambda_k \rightarrow 0^+$ och vi får följande

$$f(\mathbf{x}) + \bar{\mathbf{w}}^t \boldsymbol{\mu}(\mathbf{x}) \geq f(\bar{\mathbf{x}}) + \bar{\mathbf{w}}^t \boldsymbol{\mu}(\bar{\mathbf{x}}) \quad (4.2)$$

Alltså ser vi att $\bar{\mathbf{x}} \in \mathbf{X}(\bar{\mathbf{w}})$. Vidare får vi genom att använda oss av definitionen av $\theta(\bar{\mathbf{w}} + \lambda_k \mathbf{d})$ och $\theta(\bar{\mathbf{w}})$ följande

$$\theta(\bar{\mathbf{w}} + \lambda_k \mathbf{d}) - \theta(\bar{\mathbf{w}}) = f(\mathbf{x}_k) + (\bar{\mathbf{w}} + \lambda_k \mathbf{d})^t \boldsymbol{\mu}(\mathbf{x}_k) - \theta(\bar{\mathbf{w}}) \quad (4.3)$$

$$\geq \lambda_k \mathbf{d}^t \boldsymbol{\mu}(\mathbf{x}_k) \quad (4.4)$$

där (4.3) gäller, eftersom vi vet att \mathbf{x}_k minimerar vår funktion $f(\mathbf{x}) + (\bar{\mathbf{w}} + \lambda_k \mathbf{d})^t \boldsymbol{\mu}(\mathbf{x})$ över $\mathbf{x} \in \mathbf{X}$. Olikheten (4.4) fås om vi använder olikheten (4.2) och gäller för alla $k \in \mathcal{K}$. Om vi nu delar olikheten $\theta(\bar{\mathbf{w}} + \lambda_k \mathbf{d}) - \theta(\bar{\mathbf{w}}) \geq \lambda_k \mathbf{d}^t \boldsymbol{\mu}(\mathbf{x}_k)$ med λ_k och noterar att $\mathbf{x}_k \rightarrow \bar{\mathbf{x}}$ då $k \in \mathcal{K}$ går mot ∞ , så får vi

$$\lim_{\substack{k \in \mathcal{K} \\ k \rightarrow \infty}} \frac{\theta(\bar{\mathbf{w}} + \lambda_k \mathbf{d}) - \theta(\bar{\mathbf{w}})}{\lambda_k} \geq \mathbf{d}^t \boldsymbol{\mu}(\bar{\mathbf{x}})$$

Nu använder vi oss av ett lemma³ som säger att

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) = \lim_{\lambda \rightarrow 0^+} \frac{\theta(\bar{\mathbf{w}} + \lambda \mathbf{d}) - \theta(\bar{\mathbf{w}})}{\lambda}$$

³Lemma 3.1.5 Nonlinear Programming, Bazaraa, sid 83

existerar och satsen är bevisad. \square

Corollary 1. Låt $\partial\theta(\bar{\mathbf{w}})$ vara mängden av subgradients till θ i punkten $\bar{\mathbf{w}}$ och anta att antagandena i sats 7 gäller. Då gäller följande

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) = \inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})\}$$

Bevis. Låt $\bar{\mathbf{x}}$ vara definerad som i Sats 7 ovan. Enligt Sats 6 gäller $\boldsymbol{\mu} \in \partial\theta(\bar{\mathbf{w}})$ och Sats 7 implicerar att

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) \geq \inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})\}$$

Vi ska nu visa att olikheten gäller åt andra hållet också. Låt $\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})$ och eftersom θ är konkav gäller följande

$$\theta(\bar{\mathbf{w}} + \lambda \mathbf{d}) - \theta(\bar{\mathbf{w}}) \leq \lambda \mathbf{d}^t \boldsymbol{\xi}$$

Dela med $\lambda > 0$ och ta gränsvärdet då $\lambda \rightarrow 0^+$, vilket ger oss att

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) \leq \mathbf{d}^t \boldsymbol{\xi}$$

Eftersom det är sant för alla $\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})$ är det också sant för

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) \leq \inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})\}$$

och beviset är slutfört. \square

En vektor \mathbf{d} kallas för *ascentriktning* till θ i punkten \mathbf{w} om det existerar ett $\delta > 0$ sådant att

$$\theta(\mathbf{w} + \lambda \mathbf{d}) > \theta(\mathbf{w}) \quad \text{för alla } \lambda \in (0, \delta)$$

Eftersom θ är konkav så är \mathbf{d} en ascentriktning till θ i punkten \mathbf{w} om och endast om $\theta'(\bar{\mathbf{w}}; \mathbf{d}) > 0$. θ kommer alltså att anta sitt maximum i en punkt \mathbf{w} om och endast om det inte finns några ascentriktningar i \mathbf{w} . Detta betyder

att $\theta'(\bar{\mathbf{w}}; \mathbf{d}) \leq 0$ för alla \mathbf{d} . Enligt Corollary 1 gäller det alltså att \mathbf{d} är en ascentriktning till θ i punkten \mathbf{w} om och endast om $\inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})\} > 0$. Man kan se det som att följande olikhet måste vara uppfylld för något $\epsilon > 0$

$$\mathbf{d}^t \boldsymbol{\xi} \geq \epsilon > 0 \quad \text{för alla } \boldsymbol{\xi} \in \partial\theta(\mathbf{w})$$

Definition 7. En vektor $\bar{\mathbf{d}}$ kallas den största ascentriktningen till θ i punkten \mathbf{w} om

$$\theta'(\mathbf{w}; \bar{\mathbf{d}}) = \max_{\|\mathbf{d}\| \leq 1} \theta'(\mathbf{w}; \mathbf{d}) \quad (4.5)$$

Det gäller alltså att hitta den största ascentriktningen $\bar{\mathbf{d}}$, och det visar sig att den sammanfaller med den subgradient som har den minsta euklidiska normen.

Sats 8. Låt \mathbf{X} vara en icke-tom kompakt mängd i \mathbb{R}^n och låt $f : \mathbb{R}^n \rightarrow \mathbb{R}$ och $\boldsymbol{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ vara kontinuerliga funktioner. Den största ascentriktningen $\bar{\mathbf{d}}$ till θ i punkten \mathbf{w} anges nedan, där $\bar{\boldsymbol{\xi}}$ är subgradienten i $\partial\theta(\bar{\mathbf{w}})$ som har den minsta euklidiska normen

$$\bar{\mathbf{d}} = \begin{cases} \mathbf{0} & \text{om } \bar{\boldsymbol{\xi}} = \mathbf{0} \\ \frac{\bar{\boldsymbol{\xi}}}{\|\bar{\boldsymbol{\xi}}\|} & \text{om } \bar{\boldsymbol{\xi}} \neq \mathbf{0} \end{cases}$$

Bevis. Enligt corollary 1 gäller följande

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) = \inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})\}$$

Om vi stoppar in uttryck (4.5) för den största ascentriktningen, får vi

$$\begin{aligned} \max_{\|\mathbf{d}\| \leq 1} \theta'(\mathbf{w}; \mathbf{d}) &= \max_{\|\mathbf{d}\| \leq 1} \inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})\} \\ &= \max_{\|\mathbf{d}\| \leq 1} \inf_{\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})} \mathbf{d}^t \boldsymbol{\xi} \end{aligned} \quad (4.6)$$

$$\max_{\|\mathbf{d}\| \leq 1} \inf_{\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})} \mathbf{d}^t \boldsymbol{\xi} \leq \inf_{\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})} \max_{\|\mathbf{d}\| \leq 1} \mathbf{d}^t \boldsymbol{\xi} \quad (4.7)$$

$$\inf_{\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})} \max_{\|\mathbf{d}\| \leq 1} \mathbf{d}^t \boldsymbol{\xi} = \inf_{\boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}})} \|\boldsymbol{\xi}\| = \|\bar{\boldsymbol{\xi}}\| \quad (4.8)$$

där vi har använt vår vetskap, att $\|\bar{\boldsymbol{\xi}}\|$ är den subgradient med den minsta euklidiska normen i $\partial\theta(\bar{\mathbf{w}})$. Från (4.6)-(4.8) ser vi att

$$\theta'(\mathbf{w}; \mathbf{d}) \leq \|\bar{\boldsymbol{\xi}}\|$$

Alltså har vi hittat den största ascentriktningen, om vi lyckas konstruera en vektor $\bar{\mathbf{d}}$ sådan att $\theta'(\mathbf{w}; \bar{\mathbf{d}}) = \|\bar{\boldsymbol{\xi}}\|$.

Om $\bar{\boldsymbol{\xi}} = \mathbf{0}$ så gäller det uppenbarligen för $\bar{\mathbf{d}} = \mathbf{0}$ att $\theta'(\mathbf{w}; \bar{\mathbf{d}}) = \|\bar{\boldsymbol{\xi}}\|$ är uppfyllt.

Nu antar vi att $\bar{\boldsymbol{\xi}} \neq \mathbf{0}$ och sätter in $\bar{\mathbf{d}} = \frac{\bar{\boldsymbol{\xi}}}{\|\bar{\boldsymbol{\xi}}\|}$ i den duala funktionen. Då får vi

$$\theta'(\mathbf{w}; \bar{\mathbf{d}}) = \inf\{\bar{\mathbf{d}}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\mathbf{w})\} = \inf\left\{\frac{\bar{\boldsymbol{\xi}}^t \boldsymbol{\xi}}{\|\bar{\boldsymbol{\xi}}\|} : \boldsymbol{\xi} \in \partial\theta(\mathbf{w})\right\} \quad (4.9)$$

$$= \frac{1}{\|\bar{\boldsymbol{\xi}}\|} \inf\{\|\bar{\boldsymbol{\xi}}\|^2 + \bar{\boldsymbol{\xi}}^t(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) : \boldsymbol{\xi} \in \partial\theta(\mathbf{w})\} \quad (4.10)$$

$$\|\bar{\boldsymbol{\xi}}\| + \frac{1}{\|\bar{\boldsymbol{\xi}}\|} \inf\{\bar{\boldsymbol{\xi}}^t(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) : \boldsymbol{\xi} \in \partial\theta(\mathbf{w})\} = \|\bar{\boldsymbol{\xi}}\| \quad (4.11)$$

där (4.10) följer av att vi kan plocka ut $1/\|\bar{\boldsymbol{\xi}}\|$ ur vårt infimums uttryck, eftersom vi återigen utnyttjar att $\|\bar{\boldsymbol{\xi}}\|$ är den vektor med den minsta euklidiska norm som finns i mängden $\partial\theta(\mathbf{w})$.

Det sista steget följer av att vi använder oss av en sats⁴ som säger att $\bar{\boldsymbol{\xi}}^t(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) \geq \mathbf{0}$ för alla $\boldsymbol{\xi} \in \partial\theta(\mathbf{w})$. Alltså blir $\inf\{\bar{\boldsymbol{\xi}}^t(\boldsymbol{\xi} - \bar{\boldsymbol{\xi}}) : \boldsymbol{\xi} \in \partial\theta(\mathbf{w})\} = \mathbf{0}$ och vi får att $\theta'(\mathbf{w}; \bar{\mathbf{d}}) = \|\bar{\boldsymbol{\xi}}\|$ och beviset är klart. \square

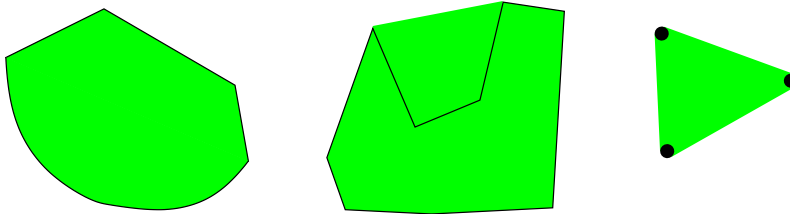
⁴Sats 2.4.1 sid 43 i Nonlinear Programming, Bazaraa

Återstår nu bara frågan om hur man hittar mängden av alla subgradienter i punkten $\bar{\mathbf{w}}$. Svaret är att man hittar alla subgradienter i det konvexa höljet till $\{\boldsymbol{\mu}(\mathbf{y}) : \mathbf{y} \in \mathbf{X}(\bar{\mathbf{w}})\}$

Definition 8. Låt $S \in \mathbb{R}^n$. Konvexa höljet till S betecknas med $H(S)$ och defineras så här: $\mathbf{x} \in H(S)$ om och endast om \mathbf{x} kan skrivas

$$\mathbf{x} = \sum_{j=1}^k \lambda_j \mathbf{x}_j$$

där $\sum_{j=1}^k \lambda_j = 1$ och $\lambda_j \geq 0$ för $j = 1, \dots, k$



Figur 4.2: Konvexa höljet till mängderna i figur 2.1

Man säger att $H(S)$ är mängden av alla konvexa kombinationer av S .

Sats 9. Låt \mathbf{X} vara en icke-tom kompakt mängd i \mathbb{R}^n , och låt $f : \mathbb{R}^n \rightarrow \mathbb{R}$ och $\boldsymbol{\mu} : \mathbb{R}^n \rightarrow \mathbb{R}^{m+l}$ vara kontinuerliga funktioner. Då är $\boldsymbol{\xi}$ en subgradient till θ i punkten $\bar{\mathbf{w}} \in \mathbb{R}^{m+l}$ om och endast om $\boldsymbol{\xi}$ tillhör det konvexa höljet till $\{\boldsymbol{\mu}(\mathbf{y}) : \mathbf{y} \in \mathbf{X}(\bar{\mathbf{w}})\}$.

Bevis. Beteckna mängden $\{\boldsymbol{\mu}(\mathbf{y}) : \mathbf{y} \in \mathbf{X}(\bar{\mathbf{w}})\}$ med Ω och dess konvexa hölje med $H(\Omega)$. Enligt sats 6 så är $\Omega \subseteq \partial\theta(\bar{\mathbf{w}})$, och eftersom $\partial\theta(\bar{\mathbf{w}})$ är konvex så är $H(\Omega) \subseteq \partial\theta(\bar{\mathbf{w}})$. Genom att använda sig av att \mathbf{X} är kompakt och att $\boldsymbol{\mu}$ är kontinuerlig, kan man visa att Ω är kompakt. Vidare så är det konvexa höljet av en kompakt mängd stängd. Därför har vi att $H(\Omega)$ är en stängd konvex mängd. Nu ska vi visa att $H(\Omega) \supseteq \partial\theta(\bar{\mathbf{w}})$

Vi antar motsatsen nämligen att det finns ett $\boldsymbol{\xi}' \in \partial\theta(\bar{\mathbf{w}})$, men inte i $H(\Omega)$. Nu ska vi använda en sats⁵ som säger att det existerar en skalär α och en nollskild vektor \mathbf{d} sådan att

$$\mathbf{d}^t \boldsymbol{\mu}(\mathbf{y}) \geq \alpha \quad \text{för alla } \mathbf{y} \in \mathbf{X}(\bar{\mathbf{w}}) \quad (4.12)$$

$$\mathbf{d}^t \boldsymbol{\xi}' < \alpha \quad (4.13)$$

Enligt sats 4 så existerar det ett $\mathbf{y} \in \mathbf{X}(\bar{\mathbf{w}})$ sådant att $\theta'(\bar{\mathbf{w}}; \mathbf{d}) \geq \mathbf{d}^t \boldsymbol{\mu}(\mathbf{y})$ och enligt (2.1) så måste $\theta'(\bar{\mathbf{w}}; \mathbf{d}) \geq \alpha$, men enligt Corollary 1 och (2.2) får vi

$$\theta'(\bar{\mathbf{w}}; \mathbf{d}) = \inf\{\mathbf{d}^t \boldsymbol{\xi} : \boldsymbol{\xi} \in \partial\theta(\bar{\mathbf{w}}) \leq \mathbf{d}^t \boldsymbol{\xi}' < \alpha$$

vilket är en motsägelse. Alltså måste $\boldsymbol{\xi} \in H(\Omega)$ och $\partial\theta(\bar{\mathbf{w}}) = H(\Omega)$ och beviset är slutfört. \square

4.3 En algoritm

Vi ska nu titta på ett exempel från [3] som visar hur en algoritm som använder sig av subgradientoptimering, kan vara uppbyggd. Ett stort problem är att det är svårt att hitta $\bar{\mathbf{d}}$ eller att ens hitta en subgradient som helt säkert är en ascentriktning. Därför måste vi vara noga när vi väljer steglängd, då vi uppdaterar våra multiplikatorer.

$$\mathbf{w}^{k+1} = \mathbf{w}^k + t^k \boldsymbol{\xi}^k$$

Ett alternativ till steglängd är $t^{(k)} = 1/k$. Det uppfyller det teoretiska kravet att $t^{(k)} \rightarrow 0$ och $\sum_{i=1}^k \rightarrow \infty$ då $k \rightarrow \infty$. Ett vanligt val av steglängd är

$$t^{(k)} = \frac{\lambda(z^* - \theta(\mathbf{w}^{(k)}))^2}{\|\boldsymbol{\xi}\|} \quad (4.14)$$

där $\lambda \in (0, 2]$ och z^* är det optimala målfunktionsvärdet till det primala heltalsproblemet. Visste vi z^* skulle vi ju vara klara, så i praktiken använder vi oss av den bästa approximationen av z^* som vi har.

⁵Theorem 2.4.4 Nonlinear Programming

Vårt värde på λ kan vi sätta till två och sedan halvera för varje gång det duala målfunktionsvärdet inte förbättras under ett visst antal iterationer. Nu till själva algoritmen.

Steg 0 Ta fram θ genom att bestämma vilka villkor som ska relaxeras i det primala problemet. Välj ett startvärde på \mathbf{w} som betecknas $\mathbf{w}^{(0)}$. Sätt $k = 0$, $\text{LBD} = -\infty$ och $\text{UBD} = +\infty$ samt välj initiala värden på $\lambda^0 \in (0, 2]$, ϵ , k_{\max} och p .

LBD och UBD står för problemets undre (*lower bounds*) respektive övre begränsningar (*upper bounds*), ϵ visar vilken felmarginal man är villig att acceptera, k_{\max} bestämmer hur länge man vill fortsätta algoritmen och på liknade sätt säger p hur länge man vill hålla på utan att målfunktionen ökar.

Steg 1 Lös Lagrangesubproblemet för givet $\mathbf{w}^{(k)}$. Detta ger oss $\mathbf{x}^{(k)}$ samt en uppskattning av problemets lösning genom $\theta(\mathbf{w}^{(k)})$. Uppdatera $\text{LBD} = \max\{\text{LBD}, \theta(\mathbf{w}^{(k)})\}$

Steg 2 Vi behöver här en punkt $(\mathbf{y})^k$. $(\mathbf{y})^k$ kan vi sätta till $(\mathbf{x})^k$ om $(\mathbf{x})^k$ skulle vara en tillåten lösning till vårt primala problem. Om så inte är fallet, utgår vi från $(\mathbf{x})^k$ för att hitta vår punkt $(\mathbf{y})^k$ med hjälp av heuristik. Uppdatera $\text{UBD} = \min(\text{UBD}, z((\mathbf{y})^k))$ och låt $\bar{\mathbf{x}}$ vara den bästa lösningen.

Heuristik kan utnyttjas som ett delmoment i avancerade lösningsalgoritmer. Dessa algoritmers mål är inte att hitta en optimal lösning till ett optimeringsproblem, utan deras uppgift är att hitta en tillåten lösning. Vi ska inte här gå in på någon djupare teori om hur de är uppbyggda.

Steg 3 Kontrollera avbrottskriterierna. Om $(\text{UBD} - \text{LBD}) / \text{LBD} \leq \epsilon$ eller $k = k_{\max}$ stannar vi och låter $\bar{\mathbf{x}}$ vara lösningen till problemet.

Steg 4 Beräkna subgradienten $\boldsymbol{\mu}^{(k)} = \boldsymbol{\beta}(\mathbf{x}^{(k)})$

Steg 5 Beräkna steglängden $t^k = \frac{\lambda(\text{UBD} - \theta(\mathbf{w}^{(k)}))}{\|\boldsymbol{\xi}\|^2}$

Steg 6 Uppdatera Lagrangemultiplikatorerna enligt $\mathbf{w}^{k+1} = \mathbf{w}^k + t^k \boldsymbol{\xi}^k$.

Steg 7 Uppdatera $\lambda := \lambda/2$ om LBD inte har förbättrats under de p senaste iterationerna. Om inte $k = k_{\max}$ sätt $k := k + 1$ och gå till steg 1.

Kapitel 5

Schemaläggning

Detta är ett verkligt problem som görs åt städföretaget Rengörare Näslund. Problem uppkommer när företaget ska ge offerter på nya städprojekt och behöver beräkna vad städningen kommer att kosta i arbetskraft per dag. När städfirman får en förfrågan om offert, får de veta hur stor yta som ska städas samt med vilken frekvens de olika städmomenten ska utföras. Den tid det tar att genomföra de olika städmomenten får vi av städfirman, så dem behöver vi inte bekymra oss om. Vi ska nu titta på ett befintligt kontrakt¹.

5.1 Målfunktionen

Problemet som ska lösas är följande: Hur många arbetstimmar måste man utnyttja per dag för att kontraktet ska kunna uppfyllas. Antalet anställda ska vara samma alla dagar i veckan. Vår målfunktion blir alltså att minimera antalet arbetstimmar per dag. Ett biproblem, som också ska lösas, är i vilken ordning man ska utföra de olika städmomenten så att det inte uppstår flaskhalsar.

¹Se appendix för frekvenserna på de olika städmomenten samt de olika ytorna på de olika våningarna.

5.2 Bivillkor

Vi listar nu de olika bivillkoren som vi ska ta hänsyn till när vi formulerar vårt problem matematiskt.

Deadlines De olika städmomenten måste vara utförda före en viss deadline. Det finns sex olika deadline-typer för när städmomenten ska vara avklarade:

- Två deadlines per dag
 - Första deadlinemomentet påbörjas klockan 6.00 och ska vara färdigt klockan 8.00
 - Andra deadlinemomentet påbörjas klockan 12.00 och ska vara färdigt klockan 14.00
- En deadline per dag. Man får börja när man vill under dagen och städmomentet ska vara klart klockan 16.00
- Två deadlines per vecka. Här har man tre olika alternativ på när man ska göra aktiviteterna
 - Gör momentet måndagar och onsdagar
 - Gör momentet tisdagar och torsdagar
 - Gör momentet onsdagar och fredagar
- En deadline per vecka. Får göras valfri dag
- Två deadlines per månad. Här finns två alternativ
 - Gör momentet vecka ett och tre
 - Gör momentet vecka två och fyra
- En deadline per månad. Får göras när som helst under månaden

Arbetspass Vi har olika typer av arbetstider vi kan utnyttja. De olika arbetstiderna listas här

- 6.00-9.00
- 6.00-10.00
- 6.00-11.00
- 6.00-14.00
- 7.00-16.00

Bemanning Det ska alltid vara minst en anställd som jobbar under tiden 6.00-16.00

Raster Det finns två fikaraster och en lunchrast.

Transportkostnad Vi kommer inte ta med några transportkostnader när man städar en våning. Anledning är att problemet då blir för stort. Som det är idag så passeras alla korridorer en gång om dagen, och då får man röra sig med sunt förnuft. Däremot kommer vi att lägga på en transportkostnad när man rör sig mellan våningar. Om man rör sig mellan våningar i samband med en rast eller lunch så försvinner transportkostnaden.

5.3 Uppdelningen av tiden

Vi kommer att dela in arbetsdagarna i olika perioder, som bestäms av deadlinetider och raster.

- **06.00** Arbetsdagen börjar
- **8.00** Första deadline
- **9.00-9.15** Rast
- **11.30-12.00** Lunch
- **14.00** Andra deadline
- **14.00-14.15** Rast
- **16.00** Arbetsdagen slutar

Detta ger oss fem olika arbetsperioder: 6.00-8.00, 8.00-9.00, 9.15-11.30, 12.00-14.00 och 14.15-16.00.

Deadlines av typen två gånger om dagen ska alltså vara färdigstäda före klockan 8.00 respektive klockan 14.00.

En arbetsmånad är 20 dagar.

5.4 Matematisk modellering

Nu ska vi beskriva problemet matematiskt.

5.4.1 Parametrar

Vi har följande parametrar som är givna. De stora bokstäverna symboliserar mängder.

- A - de olika arbetspassen
- D - de olika dagarna
- P - de olika perioderna
- V - de olika våningarna
- M - de olika städmomenten
- $t_{v,m}$ - tid det tar att slutföra ett visst städmoment $v \in V$, $m \in M$
- c_p - den minsta bemanning det måste vara under de olika perioderna $p \in P$
- k_1 och k_2 konstanter som vi har givna

Om vi tittar på städschemat², ser vi, att det finns nästan tvåhundra olika städmoment, vilket betyder att vi kommer att få ett stort antal variabler. För att underlätta, sorterar vi de olika städmomenten efter vilka deadlinetider de har. Mängden M innehåller då $M = \{d1, d2, d3, d4, d5, d6\}$ där

- $d1$ klart två gånger per dag
- $d2$ klart en gång per dag
- $d3$ klart två gånger i veckan
- $d4$ klart en gång per vecka
- $d5$ klart två gånger per månad
- $d6$ klart en gång per månad

Tiderna är som ovan.

²Appendix

5.4.2 Variabler

- $r_{a,p,d,v,m}$ - resurs man lägger ner på ett städmoment m , där $a \in A, p \in P, d \in D, v \in V$
- $i_{a,b,d,v}$ - logisk 0/1-variabel för att hålla koll på om man byter våning under transportkostnadsintervall, där $a \in A, b = 1, 2, 3, 4; d \in D, v \in V$
- w_a - indikerar om man använder anställd $a \in A$
- R_p - total mängd arbetstimmar man har under period $p \in P$, R_p är en linjär funktion av antalet arbetare.

5.4.3 Målfunktion och bivillkor

Målfunktionen är att minimera den totala arbetskostnaden per dag

$$\min \sum_{p \in P} R_p$$

under följande bivillkor.

Definerar R_p

$$R_1 = 120(w_1 + w_2 + w_3 + w_4) + 60w_5$$

$$R_2 = 60(w_1 + w_2 + w_3 + w_4 + w_5)$$

$$R_3 = 135(w_4 + w_5) + 120w_3 + 60w_2$$

$$R_4 = 120(w_4 + w_5)$$

$$R_5 = 105w_5 + 45w_4$$

Tar reda på om det alltid är bemannat under dagarna

$$R_p \geq c_p \quad \forall p \in P$$

Nu går vi igenom alla deadlines.

Moment färdiga två gånger om dagen

$$\begin{aligned} \sum_{a \in A} r_{a,1,d,v,d1} &\geq t_{v,d1} & \forall d \in D, v \in V \\ \sum_{a \in A} r_{a,3,d,v,d1} &\geq t_{v,d1} & \forall d \in D, v \in V \end{aligned}$$

Moment färdiga en gång per dag

$$\sum_{a \in A} \sum_{p \in P} r_{a,p,d,v,d2} \geq t_{v,d2} \quad \forall \quad d \in D, v \in V$$

Moment färdiga två gånger i veckan, börjar med måndag och onsdag

$$\begin{aligned} \sum_{a \in A} \sum_{p \in P} r_{a,p,1+5k,v,d3} &\geq t_{v,d3} x_1 & \forall \quad v \in V, k = 0, \dots, 3 \\ \sum_{a \in A} \sum_{p \in P} r_{a,p,3+5k,v,d3} &\geq t_{v,d3} (x_1 + x_3) & \forall \quad v \in V, k = 0, \dots, 3 \end{aligned}$$

tisdag och torsdag

$$\begin{aligned} \sum_{a \in A} \sum_{p \in P} r_{a,p,2+5k,v,d3} &\geq t_{v,d3} x_2 & \forall \quad v \in V, k = 0, \dots, 3 \\ \sum_{a \in A} \sum_{p \in P} r_{a,p,4+5k,v,d3} &\geq t_{v,d3} x_2 & \forall \quad v \in V, k = 0, \dots, 3 \end{aligned}$$

onsdag och fredag

$$\begin{aligned} \sum_{a \in A} \sum_{p \in P} r_{a,p,3+5k,v,d3} &\geq t_{v,d3} (x_3 + x_1) & \forall \quad v \in V, k = 0, \dots, 3 \\ \sum_{a \in A} \sum_{p \in P} r_{a,p,5+5k,v,d3} &\geq t_{v,d3} x_3 & \forall \quad v \in V, k = 0, \dots, 3 \end{aligned}$$

Summera vikterna

$$x_1 + x_2 + x_3 = 1$$

Moment färdiga en gång i veckan

$$\sum_{a \in A} \sum_{p \in P} \sum_{d=1}^5 r_{a,p,d+5k,v,d4} \geq t_{v,d4} \quad \forall \quad v \in V, k = 0, 1, 2, 3$$

Moment färdiga två gånger i månaden, börjar med vecka ett och tre

$$\sum_{a \in A} \sum_{p \in P} \sum_{d=1}^5 r_{a,p,d+10k,v,d5} \geq t_{v,d5} y_1 \quad \forall \quad v \in V, k = 0, 1$$

vecka två och fyra

$$\sum_{a \in A} \sum_{p \in P} \sum_{d=5}^{10} r_{a,p,d+10k,v,d5} \geq t_{v,d5} y_2 \quad \forall \quad v \in V, k = 0, 1$$

Summera vikterna

$$y_1 + y_2 = 1$$

Moment färdiga en gång i månaden

$$\sum_{a \in A} \sum_{p \in P} \sum_{d \in D} r_{a,p,d,v,d6} \geq t_{v,d6} \quad \forall \quad v \in V$$

Vi undersöker vilka våningar som har besökts

$$\begin{aligned} \sum_{p=1}^2 r_{a,p,d,v,m} &\leq j_{a,1,d,v} k_1 & \forall \quad a \in A, d \in D, v \in V \\ \sum_{m \in M} r_{a,3,d,v,m} &\leq j_{a,2,d,v} k_1 & \forall \quad a \in A, d \in D, v \in V \\ \sum_{m \in M} r_{a,4,d,v,m} &\leq j_{a,3,d,v} k_1 & \forall \quad a \in A, d \in D, v \in V \\ \sum_{m \in M} r_{a,5,d,v,m} &\leq j_{a,4,d,v} k_1 & \forall \quad a \in A, d \in D, v \in V \end{aligned}$$

Undersöker vilka arbetspass som har använts

$$\sum_{p \in P} \sum_{d \in D} \sum_{v \in V} \sum_{m \in M} r_{a,p,d,v,m} \leq w_a k_2 \quad \forall a \in A$$

Ser till att vi har tillräckligt med arbetskraft för de olika perioderna

$$\begin{aligned} \sum_{a \in A} \sum_{p=1}^2 \sum_{v \in V} \sum_{m \in M} r_{a,p,d,v,m} + 5j_{a,1,d,v} - 5w_a &\leq R_1 + R_2 & \forall d \in D \\ \sum_{a \in A} \sum_{v \in V} \sum_{m \in M} r_{a,3,d,v,m} + 5j_{a,2,d,v} - 5w_a &\leq R_3 & \forall d \in D \\ \sum_{a \in A} \sum_{v \in V} \sum_{m \in M} r_{a,4,d,v,m} + 5j_{a,3,d,v} - 5w_a &\leq R_4 & \forall d \in D \\ \sum_{a \in A} \sum_{v \in V} \sum_{m \in M} r_{a,5,d,v,m} + 5j_{a,4,d,v} - 5w_a &\leq R_5 & \forall d \in D \end{aligned}$$

Våra variabler definieras

$$\begin{aligned} r_{a,p,d,v,m} &\in \mathbb{R}_+ & \forall a \in A, p \in P, d \in D, v \in V, m \in M \\ j_{a,b,d,v} &\in \{0, 1\} & \forall a \in A, b = 1, 2, 3, 4; d \in D, v \in V \\ w_a &\in \{0, 1\} & \forall a \in A \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}_+ \end{aligned}$$

5.5 Resultat

För att praktiskt lösa problemet ovan använder jag mig av AMPL [5] som är ett matematiskt modelleringsspråk. Själva algoritmerna som jag använder mig av kommer från ett solverpaket som heter *lpsolve*. Det använder sig av både *simplex iterationer* och *Branch and Bound iterationer*. *Branch and Bound iterationerna* används för heltalsvariablerna, men det visar sig att de tar avsevärd tid att bli klara. Rent praktiskt betyder det att jag får avstå från att ha med transportkostnaden mellan våningar när jag modellerar problemet. Jag kommer att återkomma till detta problem längre fram i texten och där visa hur jag justerar data för att få ett rimligare svar. Eftersom transportavgifterna försvinner, betyder det att cirka sjuhundra heltalsvariabler försvinner från modellen. Det jag gör då, är att öka modellens antal arbetspass, som också är heltalsvariabler, från fem till tio. Det låter inte så mycket, men eftersom antalet r -variabler är indexerade över arbetstid, perioder, dagar, våningar och arbetsmoment betyder det att antalet r -variabler kommer att fördubblas. De nya arbetspassen som modellen kan välja mellan är följande tio

- 6.00-9.00
- 6.00-10.00
- 6.00-11.00

- 6.00-12.00
- 6.00-14.00
- 7.00-12.00
- 7.00-15.00
- 7.00-16.00
- 8.00-12.00
- 8.00-14.00

Koden som skrivs i AMPL återfinns i bilaga B.

Om jag försöker göra modellen större, exempelvis att ha elva arbetspass, så klara inte min dator av att lösa modellen. Om man istället gör modellen mindre genom att ta bort det sista arbetspasset 8.00-14.00 så får man följande

Totala antalet variabler	Antalet heltalsvariabler	Antalet bivillkor	Antalet Branch & Bound	Antalet simplex iterationer
21774	9	4966	24	98 000

Om man istället använder alla tio arbetspassen enligt ovan så får man följande

Totala antalet variabler	Antalet heltalsvariabler	Antalet bivillkor	Antalet Branch & Bound	Antalet simplex iterationer
24335	10	5506	24	180 000

Vi ser att modellen inte växer linjärt utan exponentiellt.

Konsekvenserna av att vi inte har modellerat transportkostnader, kan vi tydligt se i data vi får ut när vi har kört modellen. Det förekommer nämligen orimligt korta besök vid de olika våningarna. Ett exempel på detta är att

arbetspasset 7.00-15.00 under andra perioden ska spendera 0,146 minuter på våning tre. Hela listan med hur arbetspassen ska användas återfinns i bilaga B.

För att korrigera detta problem, så har jag konstruerat en sorteringsalgoritm som tar bort strötider, genom att byta tider mellan de olika arbetspassen. Sorteringsalgoritmen, och den nya data den ger upphov till, återfinns i bilaga C.

På nästa sida visas hur slutdata presenteras. Det gröna representerar arbetsperioder som utnyttjas fullt ut, rött representerar arbetsperioder där arbetstiden inte utnyttjas fullt ut och det gula länkar ihop två arbetsperioder där det inte förkommer någon rast mellan perioderna.

Arbetspass 4 6.00-12.00

	6.00-8.00	8.00-9.00	9.15-11.30	12.00-14.00	14.15-16.00
våning_1	41	0	0	0	
våning_2	0	0	0	0	
våning_3	0	1	0	0	
våning_4	0	35	0	45	
våning_5	78	0	80	0	
våning_6	1	17	55	0	
Totalt	120	54	135	45	

Arbetspass 5 6.00-14.00

	6.00-8.00	8.00-9.00	9.15-11.30	12.00-14.00	14.15-16.00
våning_1	0	14	0	0	7
våning_2	0	0	0	0	0
våning_3	3	0	75	120	0
våning_4	72	37	0	0	0
våning_5	0	10	0	0	4
våning_6	45	0	60	0	34
Totalt	120	60	135	120	45

Arbetspass 7 7.00-15.00

	6.00-8.00	8.00-9.00	9.15-11.30	12.00-14.00	14.15-16.00
våning_1	0	0	0	0	0
våning_2	13	27	38	0	31
våning_3	0	0	0	46	66
våning_4	47	0	0	0	0
våning_5	0	0	0	74	0
våning_6	0	22	97	0	0
Totalt	60	50	135	120	97

Arbetspass 10 8.00-14 8.00-14.00

	6.00-8.00	8.00-9.00	9.15-11.30	12.00-14.00	14.15-16.00
våning_1		0	24	43	3
våning_2		23	68	15	0
våning_3		24	0	30	0
våning_4		0	29	1	42
våning_5		5	15	0	0
våning_6		0	0	25	0
Totalt		52	135	115	45

Bilaga A

Arbetsdata

Vi börjar med att visa hur många kvadratmeter av de olika utrymmestyperna som finns på varje våning. Sedan visas de olika städmomenten och med vilken frekvens de förekommer. Därefter följer avverkningsgraden, som mäts i kvadratmeter per arbetstimme, på samma städmoment

Vi ser här att det finns ett stort antal olika städmoment varför vi blir tvugna att sortera dem i sex olika grupper, efter vilken deadline de har.

Resten av bilderna i detta avsnitt visar hur stor yta av de olika utrymmestyperna som finns på varje våning.

	Våning 1	Våning 2	Våning 3	Våning 4	Våning 5	Våning 6
S01	0	193.59	1299.93	1205.84	1097.50	520.14
S02	78.86	454.37	862.47	845.91	800.41	602.73
S03	0	38.09	29.45	29.45	29.46	22.62
S04	0	0	0	0	0	0
S05	0	28.73	0	0	0	0
S06	0	221.23	260.14	55.20	69.85	0
S07	0	266.28	126.45	20.81	91.77	87.12
S08	0	0	0	0	0	0
S09	65.24	11.28	0	0	0	0
S10	79.99	75.58	67.53	67.44	67.51	37.69
S11	0	370.68	0	0	0	0
S12	0	785.52	464.52	91.89	357.96	209.61
S13	0	0	0	0	0	0
S14	0	1348	0	0	0	0
S15	0	0	0	0	0	0
S16	0	24.34	0	0	0	0

Städskoder : Utrymmestyp : Frekvens :	S01	S02	S03	S04	S05	S06
	Kontor	Korridor	Toalett	Dusch	Omkl.- rum	Pausrum
	Kopierin	Kapprum	Bastu	Tvätttrum		Pentry
	Kont.förr	Hiss	Solarium			Rökrum
	Kont.land					
GOLV						
Moppning (# fuktad moppn+avfläckn)	1V	2V				1D
Damms. heltäck.- & mjuka mattor	(1V)					
Tvättning		*1D	1D	1D	1D	
Skräp upplockas		1D			1D	1D
Papperkorgstömning	2V	2V				
Maskinskurning	1V**	1V**				1V**
MÖBLER						
Bordsytor avtorkas	1/2V	1/2V				1D
Stolar avborstas	1V	1V				VB
Dammsug stoppade möbler	1M	1M				(1M)
Damtorkning fria ytor på möbler	1/2V	1/2V				1/2V
FAST INREDNING						
Fönsterbänkar avtorkas	1V	1V				1V
Avfläckning dörrar karmar m.m.	1V	VB	1D	1D	VB	VB
Socklar dammsuges/avtorkas	1M	1M				1M
Ledstänger & räcken avtorkas						
Sanitär inredning rengörs			1D	1D		
Speglar, kakel vid handfat rengörs		(VB)	1D	1D		
Speglar rengörs					VB	
Glas i dörrar putsas		VB				
Diskbänk rengörs, avtorkn. kaffeaut.						1D
LÖS INREDNING						
Skräpkorg töms (i pentry efter lunch.)	2V	(VB)	1D	1D	1D	1D
Askkoppar töms & rengörs						(1D)
Toalettartiklar påfylls			1D	1D		
Skrivtavlor avtorkas						
Stolar lyfts upp						
Britsar avtorkas						
Arbetsbänkar, fuktroppas						
SÄRSKILD INSTRUKTION						
FÖRKORTNINGAR :						
1D = En gång/dag			1/2V=En gång var annan vecka			
1V = En gång/vecka			1M = En gång/månad			
2V = Två gånger/vecka			VB = Utförs vid behov			
3V = Tre gånger/vecka			() = Där det förekommer			

Städskoder : Utrymmestyp : Frekvens :	S07	S08	S09	S10	S11
	Konfer- ensrum	Utbildning Utställning	Entréer Reception	Trappor	Produk- tion
	Hörsal				Lager
	Filmsal				Gods
GOLV					
Moppling (# fuktad moppn+avfläckn)	1D	1D	2D***	2V	1D
Damms. heltäck.- & mjuka mattor	(1D)	(1D)	2D***		
Tvättning			2D***	1V	
Skräp upplockas	1D	1D	2D***	1D	1D
Papperkorgstömning					
Maskinskurning			1V**		
MÖBLER					
Bordsytor avtorkas	1D	1D	1D		
Stolar avborstas	1D	1D	1D		
Dammsug stoppade möbler	1M	1M	1V		
Damtorkning fria ytor på möbler	1V	1V	1D		
FAST INREDNING					
Fönsterbänkar avtorkas	1V	1V	1V		(1V)
Avfläckning dörrar karmar m.m.	1D	1D	2D		1V
Socklar dammsuges/avtorkas	1M	1M	1M		(1M)
Ledstänger & räcken avtorkas				1V	
Sanitär inredning rengörs					
Speglar, kakel vid handfat rengörs					(1D)
Speglar rengörs			(1D)		
Glas i dörrar putsas			2D		
Diskbänk rengörs, avtorkn. kaffeaut.					
LÖS INREDNING					
Skräpkorg töms (i pentry efter lunch.)	1D	1D	2D		2V
Askkoppar töms & rengörs					
Toalettartiklar påfylls					(1D)
Skrivtavlor avtorkas	1D	1D			
Stolar lyfts upp					
Britsar avtorkas					
Arbetsbänkar, fuktmoppas					
SÄRSKILD INSTRUKTION					
FÖRKORTNINGAR :					
1D = En gång/dag		1/2V=En gång var annan vecka			
1V = En gång/vecka		1M = En gång/månad			
2V = Två gånger/vecka		VB = Utförs vid behov			
3V = Tre gånger/vecka		() = Där det förekommer			

Städskoder : Utrymmestyp :	S12	S13	S14	S15	S16
	Lab (ESD-golv)	Matsalar	Hälsovård	Sporthall	Datahall Syst.rum (datagolv)
Frekvens :					
GOLV					
Moppling (# fuktad moppn+avfläckn)	1D	1D	1D	1D	1V
Damms. heltäck.- & mjuka mattor					
Tvättning			(1D)	1D	
Skräp upplockas	1D	1D	1D	1D	1D
Papperkorgstömning	1D		1D		2V
Maskinskurning					
MÖBLER					
Bordsytor avtorkas			1D		1M
Stolar avborstas	1V	1V	1V		1M
Dammsug stoppade möbler			1M		1M
Dammtorkning fria ytor på möbler				1V	
FAST INREDNING					
Fönsterbänkar avtorkas	(1V)	(1V)	1V		1M
Avfläckning dörrar karmar m.m.	1D		1V	1V	1V
Socklar dammsuges/avtorkas	1M X	1M	1M	1M	1M
Ledstänger & räcken avtorkas					
Sanitär inredning rengörs			(1D)		
Speglar, kakel vid handfat rengörs			(1D)		
Speglar rengörs				1V	
Glas i dörrar putsas		VB			VB
Diskbänk rengörs, avtorkn. kaffeaut.					
LÖS INREDNING					
Skräpkorg töms (i pentry efter lunch.)	2V		VB		
Askkoppar töms & rengörs					
Toalettartiklar påfylls			(1D)		
Skrivtavlor avtorkas					
Stolar lyfts upp					
Britsar avtorkas			1D	1V	
Arbetsbänkar, fuktmoppas					
SÄRSKILD INSTRUKTION			X		X
FÖRKORTNINGAR :					
1D = En gång/dag			1/2V=En gång var annan vecka		
1V = En gång/vecka			1M = En gång/månad		
2V = Två gånger/vecka			VB = Utförs vid behov		
3V = Tre gånger/vecka			() = Där det förekommer		

Städskoder : Utrymmestyp : Frekvens :	S01	S02	S03	S04	S05	S06
	Kontor	Korridor	Toalett	Dusch	Omkf.-rum	Pausrum
	Kopierin	Kaprum	Bastu	Tvätttrum		Pentry
	Kont.förr	Hiss	Solarium			Rökrum
		Kont.land				
GOLV						
Moppning (# fuktad moppn+avfläckn)	1000	1000				500
Damms. heltäck.- & mjuka mattor	800					
Tvättning		400	100	200	600	
Skräp upplockas		6000			600	3000
Papperkorgstömning	2400	2400				
Maskinskurning	1500	1500				750
MÖBLER						
Bordsytor avtorkas	540	540				250
Stolar avborstas	3600	3600				1000
Dammsug stoppade möbler	720	720				3000
Dammtorkning fria ytor på möbler	540	540				540
FAST INREDNING						
Fönsterbänkar avtorkas	1440	1440				1440
Avfläckning dörrar karmar m.m.	1440	1440	100	200	2000	1440
Socklar dammsuges/avtorkas	720	1440				1440
Ledstänger & räcken avtorkas						
Sanitär inredning rengörs			100	200		
Speglar, kakel vid handfat rengörs		10000	100	200		
Speglar rengörs					2000	
Glas i dörrar putsas		1000				
Diskbänk rengörs, avtorkn. kaffeaut.						240
LÖS INREDNING						
Skräpkorg töms (i pentry efter lunch.)	6000	6000	100	200	600	3000
Askkoppar töms & rengörs						10000
Toalettartiklar påfylls			100	200		
Skrivtavlor avtorkas						
Stolar lyfts upp						
Britsar avtorkas						
Arbetsbänkar, fuktmoppas						
SÄRSKILD INSTRUKTION						
FÖRKORTNINGAR :						
1D = En gång/dag			1/2V=En gång var annan vecka			
1V = En gång/vecka			1M = En gång/månad			
2V = Två gånger/vecka			VB = Utförs vid behov			
3V = Tre gånger/vecka			() = Där det förekommer			

Städskoder :	S07	S08	S09	S10	S11
Utrymmestyp :	Konfer- ensrum	Utställning	Entréer Reception	Trappor	Produk- tion
Frekvens :	Hörsal				Lager
	Filmsal				Gods
GOLV					
Moppling (# fuktad moppn+avfläckn)	500	500	400	700	800
Damms. heltäck.- & mjuka mattor	800	800	400		
Tvättning			400	400	
Skräp upplockas	3000	3000	3000	3000	3000
Papperkorgstömning					
Maskinskurning			1000		
MÖBLER					
Bordsytor avtorkas	540	540	540		
Stolar avborstas	1000	1000	1000		
Dammsug stoppade möbler	1500	1500	1500		
Dammtorkning fria ytor på möbler	540	540	540		
FAST INREDNING					
Fönsterbänkar avtorkas	1440	1440	1440		1440
Avfläckning dörrar karmar m.m.	1440	1440	1440		1440
Socklar dammsuges/avtorkas	1440	1440	1440		1440
Ledstänger & räcken avtorkas				1400	
Sanitär inredning rengörs					
Speglar, kakel vid handfat rengörs					10000
Speglar rengörs			10000		
Glas i dörrar putsas			800		
Diskbänk rengörs, avtorkn. kaffeaut.					
LÖS INREDNING					
Skräpkorg töms (i pentry efter lunch.)	6000	6000	6000		6000
Askkoppar töms & rengörs					
Toalettartiklar påfylls					1000
Skrivtavlor avtorkas	720	720			
Stolar lyfts upp					
Britsar avtorkas					
Arbetsbänkar, fuktmoppas					
SÄRSKILD INSTRUKTION					
FÖRKORTNINGAR :					
1D = En gång/dag			1/2V=En gång var annan vecka		
1V = En gång/vecka			1M = En gång/månad		
2V = Två gånger/vecka			VB = Utförs vid behov		
3V = Tre gånger/vecka			() = Där det förekommer		

Städskoder : Utrymmestyp :	S12	S13	S14	S15	S16
	Lab (ESD-golv)	Matsalar	Hälsovård	Sporthall	Datahall Syst.rum (datagolv)
Frekvens :					
GOLV					
Moppning (# fuktad moppn+avfläckn)	1000	500	700	3000	1300
Damms. heltäck.- & mjuka mattor					
Tvättning			400	2000	
Skräp upplockas	6000	3000	3000	10000	6000
Papperkorgstömning	2400		2400		2400
Maskinskurning					
MÖBLER					
Bordsytor avtorkas			540		1000
Stolar avborstas	3600	1000	3600		3600
Dammsug stoppade möbler			720		1500
Dammtorkning fria ytor på möbler				10000	
FAST INREDNING					
Fönsterbänkar avtorkas	1440	1440	1440		1440
Avfläckning dörrar karmar m.m.	1440		1440	10000	1440
Socklar dammsuges/avtorkas	1440	1440	1440	10000	1440
Ledstänger & räcken avtorkas					
Sanitär inredning rengörs			200		
Speglar, kakel vid handfat rengörs			200		
Speglar rengörs				10000	
Glas i dörrar putsas		800			1000
Diskbänk rengörs, avtorkn. kaffeaut.					
LÖS INREDNING					
Skräpkorg töms (i pentry efter lunch.)	6000		6000		
Askkoppar töms & rengörs					
Toalettartiklar påfylls			200		
Skrivtavlor avtorkas					
Stolar lyfts upp					
Britsar avtorkas			1000	10000	
Arbetsbänkar, fukt-moppas					
SÄRSKILD INSTRUKTION			X		X
FÖRKORTNINGAR :					
1D = En gång/dag			1/2V=En gång var annan vecka		
1V = En gång/vecka			1M = En gång/månad		
2V = Två gånger/vecka			VB = Utförs vid behov		
3V = Tre gånger/vecka			() = Där det förekommer		

Bilaga B

AMPL-kod för städproblemet

Detta är vad som skrivs i AMPL miljön

```
model stad.mod;
data stad.dat;
option solver lpsolve;
solve;
```

Här kommer först modeleringsfilen *stad.mod* och därefter datafilen *stad.dat*.

MODELERINGEN: *s.mod*

```
set PER;      #PERIODERNA
set ARB {PER}; #ARBETSPASSEN beror på vilken period man är i
set MOM;      #MOMENTEN
set VAN {MOM}; #VÅNINGARNA bero på vilket moment man gör
set ALLA_VAN;
set ALLA_ARB;  #Alla arbetspassen, används denna mängd
set P_PER {ALLA_ARB}; #display index
set P_MOM {ALLA_VAN}; #display index

param l;
```

```

param dagar;
param s {p in PER,a in ARB[p],v in ALLA_VAN};
param tid {m in MOM, VAN[m]} >= 0;
param rbeg {p in PER, m in ARB[p]} >= 0;
param k2;
param man := dagar - 4;
param tis := dagar - 3;
param ons := dagar - 2;
param tor := dagar - 1;
param fre := dagar;
var r {d in 1..dagar,p in PER,a in ARB[p], m in MOM,v in VAN[m]} >= 0;
var w {a in ALLA_ARB} binary;
var x {i in 1..3} >= 0;
var y {i in 1..2} >= 0;

```

minimize Total_Arbetskraft:

```

120*(w[lokal_vårdare_1"]+w[lokal_vårdare_2"]+w[lokal_vårdare_3"]+w[lokal_vårdare_4"]+w[ll-
okal_vårdare_5"])
+60*(w[lokal_vårdare_6"]+w[lokal_vårdare_7"]+w[lokal_vårdare_8"])
+60*(w[lokal_vårdare_1"]+w[lokal_vårdare_2"]+w[lokal_vårdare_3"]+w[lokal_vårdare_4"]+w[ll-
okal_vårdare_5"]+w[lokal_vårdare_6"]+w[lokal_vårdare_7"]+w[lokal_vårdare_8"]+w[ll-
okal_vårdare_9"]+w[lokal_vårdare_10"])
+135*(w[lokal_vårdare_4"]+w[lokal_vårdare_5"]+w[lokal_vårdare_6"]+w[lokal_vårdare_7"]+w[ll-
okal_vårdare_8"]+w[lokal_vårdare_9"]+w[lokal_vårdare_10"])+60*w[lokal_vårdare_2"]
+120*(w[lokal_vårdare_5"]+w[lokal_vårdare_7"]+w[lokal_vårdare_8"]+w[lokal_vårdare_10"])
+45*(w[lokal_vårdare_4"]+w[lokal_vårdare_6"]+w[lokal_vårdare_9"])
+105*(w[lokal_vårdare_7"]+w[lokal_vårdare_8"])+45*(w[lokal_vårdare_5"]+w[ll-
okal_vårdare_10"]);

```

#Kollar att arbetskraften räcker till

```

subject to Res1 {d in 1..dagar, p in PER, a in ARB[p]}:
sum{m in MOM, v in VAN[m]} r[d,p,a,m,v] <= rbeg[p,a];

```

```

subject to Samma_arbetskraft {d in 1..dagar, p in PER, a in ARB[p], v
in ALLA_VAN}:

```

```

sum{m in P_MOM[v]} r[d,p,a,m,v] = sum {m in P_MOM[v]} r[1,p,a,m,v];

```

#två om dan

subject to Tva_om_dan_1 {d in 1..dagar, v in VAN[d1]}:

sum{a in ARB["6.00-8.00"]} r[d,"6.00-8.00",a,d1",v] >= tid[d1",v];

subject to Tva_om_dan_2 {d in 1..dagar, v in VAN[d1]}:

sum{a in ARB["12.00-14.00"]} r[d,"12.00-14.00",a,d1",v] >= tid[d1",v];

#en om dan

subject to En_om_dan {d in 1..dagar, v in VAN[d2]}:

sum{p in PER,a in ARB[p]} r[d,p,a,d2",v] >= tid[d2",v];

#två i veckan

subject to vikter_1:

x[1]+x[2]+x[3]=1;

subject to Mandag {v in VAN[d3"], d in 1..man by 5}:

sum {p in PER,a in ARB[p]} r[d,p,a,d3",v] >= tid[d3",v]*x[1];

subject to Onsdag {v in VAN[d3"], d in 3..ons by 5}:

sum {p in PER,a in ARB[p]} r[d,p,a,d3",v] >= tid[d3",v]*(x[1]+x[3]);

subject to Tisdag {v in VAN[d3"], d in 2..tis by 5}:

sum {p in PER,a in ARB[p]} r[d,p,a,d3",v] >= tid[d3",v]*x[2];

subject to Torsdag {v in VAN[d3"], d in 4..tor by 5}:

sum {p in PER,a in ARB[p]} r[d,p,a,d3",v] >= tid[d3",v]*x[2];

subject to Fredag{v in VAN[d3"], d in 5..fre by 5}:

sum {p in PER,a in ARB[p]} r[d,p,a,d3",v] >= tid[d3",v]*x[3];

#en i veckan

subject to En_vecka_ett{v in VAN[d4]}:

sum {d in 1..5,p in PER, a in ARB[p]} r[d,p,a,d4",v] >= tid[d4",v];

subject to En_vecka_tva{v in VAN[d4]}:

```

sum {d in 6..10,p in PER, a in ARB[p]} r[d,p,a,d4",v] >= tid[d4",v];

subject to En_vecka_tre{v in VAN[d4"]}:
sum {d in 6..10,p in PER, a in ARB[p]} r[d,p,a,d4",v] >= tid[d4",v];

subject to En_vecka_fyr{v in VAN[d4"]}:
sum {d in 6..10,p in PER, a in ARB[p]} r[d,p,a,d4",v] >= tid[d4",v];

#två gånger per månad
subject to vikter_3:
y[1]+y[2]=1;

subject to Tva_man_ett{v in VAN[d5"]}:
sum {d in 1..5, p in PER, a in ARB[p]} r[d,p,a,d5",v] >= tid[d5",v]*y[1];

subject to Tva_man_tva{v in VAN[d5"]}:
sum {d in 6..10, p in PER, a in ARB[p]} r[d,p,a,d5",v] >= tid[d5",v]*y[2];

subject to Tva_man_tre{v in VAN[d5"]}:
sum {d in 11..15, p in PER, a in ARB[p]} r[d,p,a,d5",v] >= tid[d5",v]*y[1];

subject to Tva_man_fyr{v in VAN[d5"]}:
sum {d in 16..20, p in PER, a in ARB[p]} r[d,p,a,d5",v] >= tid[d5",v]*y[2];

#en i mån
subject to En_man{v in VAN[d6"]}:
sum {d in 1..dagar, p in PER, a in ARB[p]} r[d,p,a,d6",v] >= tid[d6",v];

subject to Anvanda_pass {p in PER,a in ARB[p]}:
sum{d in 1..dagar, m in MOM, v in VAN[m]} r[d,p,a,m,v] <= w[a]*k2;

DATAFILEN stad.dat
data;

set PER := 6.00-8.00 8.00-9.00 9.15-11.30 12.00-14.00 14.15-16.00;
set ARB[6.00-8.00] := a_pass_1 a_pass_2 a_pass_3 a_pass_4 a_pass_5 a_pass_6
a_pass_7 a_pass_8 ;
set ARB[8.00-9.00] := a_pass_1 a_pass_2 a_pass_3 a_pass_4 a_pass_5 a_pass_6
a_pass_7 a_pass_8 a_pass_9 a_pass_10;

```

```

set ARB[9.15-11.30] := a_pass_2 a_pass_3 a_pass_4 a_pass_5 a_pass_6 a_pass_7
a_pass_8 a_pass_9 a_pass_10;
set ARB[12.00-14.00] := a_pass_4 a_pass_5 a_pass_6 a_pass_7 a_pass_8 a_pass_9
a_pass_10;
set ARB[14.15-16.00] := a_pass_5 a_pass_7 a_pass_8 a_pass_10;
set ALLA_ARB := a_pass_1 a_pass_2 a_pass_3 a_pass_4 a_pass_5 a_pass_6
a_pass_7 a_pass_8 a_pass_9 a_pass_10;

```

```

set MOM := d1 d2 d3 d4 d5 d6;
set VAN[d1] := våning_1 våning_2;
set VAN[d2] := våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 ;
set VAN[d3] := våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 ;
set VAN[d4] := våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 ;
set VAN[d5] := våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 ;
set VAN[d6] := våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 ;
set ALLA_VAN := våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 ;

```

```

set P_PER[a_pass_1] := 6.00-8.00 8.00-9.00;
set P_PER[a_pass_2] := 6.00-8.00 8.00-9.00 9.15-11.30;
set P_PER[a_pass_3] := 6.00-8.00 8.00-9.00 9.15-11.30;
set P_PER[a_pass_4] := 6.00-8.00 8.00-9.00 9.15-11.30 12.00-14.00;
set P_PER[a_pass_5] := 6.00-8.00 8.00-9.00 9.15-11.30 12.00-14.00 14.15-
16.00;
set P_PER[a_pass_6] := 6.00-8.00 8.00-9.00 9.15-11.30 12.00-14.00;
set P_PER[a_pass_7] := 6.00-8.00 8.00-9.00 9.15-11.30 12.00-14.00 14.15-
16.00;
set P_PER[a_pass_8] := 6.00-8.00 8.00-9.00 9.15-11.30 12.00-14.00 14.15-
16.00;
set P_PER[a_pass_9] := 8.00-9.00 9.15-11.30 12.00-14.00;
set P_PER[a_pass_10] := 8.00-9.00 9.15-11.30 12.00-14.00 14.15-16.00;

```

```

set P_MOM[våning_1] := d1 d2 d3 d4 d5 d6;
set P_MOM[våning_2] := d1 d2 d3 d4 d5 d6;
set P_MOM[våning_3] := d2 d3 d4 d5 d6;
set P_MOM[våning_4] := d2 d3 d4 d5 d6;
set P_MOM[våning_5] := d2 d3 d4 d5 d6;
set P_MOM[våning_6] := d2 d3 d4 d5 d6;

```

```

param l := 0;
param dagar := 20;

param k2 := 10000;

# DATA

param tid : våning_1 våning_2 våning_3 våning_4 våning_5 våning_6 :=
d1 39 7 . . . .
d2 37 125 257 200 178 250
d3 14 64 80 120 70 74
d4 41 166 218 158 225 230
d5 18 168 309 261 129 249
d6 13 179 370 330 309 177 ;

#Resurs begränsningar
param rbeg : a_pass_1 a_pass_2 a_pass_3 a_pass_4 a_pass_5 a_pass_6 a_pass_7
a_pass_8 a_pass_9 a_pass_10 :=
6.00-8.00 120 120 120 120 120 60 60 60
8.00-9.00 60 60 60 60 60 60 60 60 60
9.15-11.30 . 60 120 135 135 135 135 135 135
12.00-14.00 . . . 45 120 45 120 120 45 120
14.15-16.00 . . . . 60 . 105 105 . 60 ;

```

PERIOD	ARBETSPASS	VÅNING	tid
6.00-8.00	arbetspass_5	våning_1	6,270797427
6.00-8.00	arbetspass_5	våning_2	5,923862053
6.00-8.00	arbetspass_5	våning_3	3,225448264
6.00-8.00	arbetspass_5	våning_4	59,61852413
6.00-8.00	arbetspass_5	våning_5	0
6.00-8.00	arbetspass_5	våning_6	44,96136812
6.00-8.00	arbetspass_7	våning_1	10,05291418
6.00-8.00	arbetspass_7	våning_2	7
6.00-8.00	arbetspass_7	våning_3	0
6.00-8.00	arbetspass_7	våning_4	42,94708582
6.00-8.00	arbetspass_7	våning_5	0
6.00-8.00	arbetspass_7	våning_6	0
6.00-8.00	arbetspass_4	våning_1	24,42770085
6.00-8.00	arbetspass_4	våning_2	0
6.00-8.00	arbetspass_4	våning_3	0
6.00-8.00	arbetspass_4	våning_4	16,58281548
6.00-8.00	arbetspass_4	våning_5	78,42769114
6.00-8.00	arbetspass_4	våning_6	0,561792526
8.00-9.00	arbetspass_5	våning_1	5,965785826
8.00-9.00	arbetspass_5	våning_2	0
8.00-9.00	arbetspass_5	våning_3	0
8.00-9.00	arbetspass_5	våning_4	44,5411208
8.00-9.00	arbetspass_5	våning_5	5,471644143
8.00-9.00	arbetspass_5	våning_6	4,021449236
8.00-9.00	arbetspass_7	våning_1	2,683726199
8.00-9.00	arbetspass_7	våning_2	26,69076258
8.00-9.00	arbetspass_7	våning_3	0,146107275
8.00-9.00	arbetspass_7	våning_4	9,937108183
8.00-9.00	arbetspass_7	våning_5	0,916014853
8.00-9.00	arbetspass_7	våning_6	9,228423835

PERIOD	ARBETSPASS	VÅNING	tid
8.00-9.00	arbetspass_10	våning_5	2,888661351
8.00-9.00	arbetspass_10	våning_6	0
9.15-11.30	arbetspass_5	våning_1	0
9.15-11.30	arbetspass_5	våning_2	0
9.15-11.30	arbetspass_5	våning_3	59,6595977
9.15-11.30	arbetspass_5	våning_4	0,088622837
9.15-11.30	arbetspass_5	våning_5	0
9.15-11.30	arbetspass_5	våning_6	75,25177946
9.15-11.30	arbetspass_7	våning_1	11,76662126
9.15-11.30	arbetspass_7	våning_2	15,26964824
9.15-11.30	arbetspass_7	våning_3	10,93842852
9.15-11.30	arbetspass_7	våning_4	0
9.15-11.30	arbetspass_7	våning_5	17,48399861
9.15-11.30	arbetspass_7	våning_6	79,54130336
9.15-11.30	arbetspass_4	våning_1	0
9.15-11.30	arbetspass_4	våning_2	45,13253403
9.15-11.30	arbetspass_4	våning_3	0
9.15-11.30	arbetspass_4	våning_4	14,26640319
9.15-11.30	arbetspass_4	våning_5	47,9301949
9.15-11.30	arbetspass_4	våning_6	27,67086788
9.15-11.30	arbetspass_10	våning_1	12,14631175
9.15-11.30	arbetspass_10	våning_2	45,84825594
9.15-11.30	arbetspass_10	våning_3	4,887485358
9.15-11.30	arbetspass_10	våning_4	14,26640319
9.15-11.30	arbetspass_10	våning_5	28,8305283
9.15-11.30	arbetspass_10	våning_6	29,02101545

PERIOD	ARBETSPASS	VÅNING	tid
12.00-14.00	arbetspass_5	våning_1	2,692167708
12.00-14.00	arbetspass_5	våning_2	0
12.00-14.00	arbetspass_5	våning_3	114,0415628
12.00-14.00	arbetspass_5	våning_4	0
12.00-14.00	arbetspass_5	våning_5	0
12.00-14.00	arbetspass_5	våning_6	3,266269479
12.00-14.00	arbetspass_7	våning_1	1,745387258
12.00-14.00	arbetspass_7	våning_2	1,009963948
12.00-14.00	arbetspass_7	våning_3	39,30603425
12.00-14.00	arbetspass_7	våning_4	0
12.00-14.00	arbetspass_7	våning_5	74,47895784
12.00-14.00	arbetspass_7	våning_6	3,45965671
12.00-14.00	arbetspass_4	våning_1	0
12.00-14.00	arbetspass_4	våning_2	1,404585401
12.00-14.00	arbetspass_4	våning_3	0
12.00-14.00	arbetspass_4	våning_4	43,5954146
12.00-14.00	arbetspass_4	våning_5	0
12.00-14.00	arbetspass_4	våning_6	0
12.00-14.00	arbetspass_10	våning_1	39
12.00-14.00	arbetspass_10	våning_2	12,11246498
12.00-14.00	arbetspass_10	våning_3	42,62184049
12.00-14.00	arbetspass_10	våning_4	2,316412284
12.00-14.00	arbetspass_10	våning_5	0
12.00-14.00	arbetspass_10	våning_6	18,64482962
14.15-16.00	arbetspass_5	våning_1	6,904124766
14.15-16.00	arbetspass_5	våning_2	0
14.15-16.00	arbetspass_5	våning_3	0
14.15-16.00	arbetspass_5	våning_4	0
14.15-16.00	arbetspass_5	våning_5	3,804676204
14.15-16.00	arbetspass_5	våning_6	34,29119903
14.15-16.00	arbetspass_7	våning_1	0
14.15-16.00	arbetspass_7	våning_2	31,26747119
14.15-16.00	arbetspass_7	våning_3	65,62468707
14.15-16.00	arbetspass_7	våning_4	0
14.15-16.00	arbetspass_7	våning_5	0
14.15-16.00	arbetspass_7	våning_6	0
14.15-16.00	arbetspass_10	våning_1	2,692167708
14.15-16.00	arbetspass_10	våning_2	0
14.15-16.00	arbetspass_10	våning_3	0
14.15-16.00	arbetspass_10	våning_4	42,30783229
14.15-16.00	arbetspass_10	våning_5	0
14.15-16.00	arbetspass_10	våning_6	0

Bilaga C

Sorteringsalgoritmen

Sorteringsalgoritmen fungerar på följande vis. Den går igenom de olika arbetspassens tider på de olika våningarna under de olika tidsperioderna. När en nollskild tid för ett visst arbetspass på en viss våning upptäcks, undersöker algoritmen om någon av de andra arbetspassen också har en nollskild tid på samma våningsplan under samma tidsperiod. Om så är fallet undersöks om de har ytterligare någon nollskild tid på något av de andra våningsplanen, men dock under samma arbetsperiod. Om så är fallet kan de två olika arbetspassen byta tider med varandra. Den arbetstid av de fyra olika arbetstiderna som är den minsta, blir den tid som byts mellan de olika arbetspassen.

Nu till själva koden som börjar på nästa sida.

```

model s.mod;
data t.dat;
option solver lpsolve;
solve;
for {p in PER} {
  for {a in ARB[p]} {
    for {v in ALLA_VAN} {
      let s[p,a,v] := sum {m in P_MOM[v]} r[1,p,a,m,v];
    }
  }
}

```

```

table B1 OUT "ODBC" "oman.xls" :
[PERIODER,ARBETS_PASS,VÅNINGAR], s ~ tidsresurs;
write table B1;

```

```

for {p in PER} {
  for {a in ARB[p]} {
    for {v in ALLA_VAN} {
      if s[p,a,v] > 0 && s[p,a,v] != rbeg[p,a] then {
        for {k in ARB[p]} {
          if k != a then {
            if s[p,k,v] > 0 && s[p,k,v] != rbeg[p,k] then {
              for {b in ALLA_VAN}
                if b != v then {
                  if s[p,k,b] > 0 && s[p,a,b] > 0 then {
                    if s[p,k,b] < s[p,k,v] && s[p,k,b] < s[p,a,b] && s[p,k,b] < s[p,a,v] then {
                      if s[p,k,b]+s[p,a,b] > rbeg[p,a] then {
                        let s[p,k,b] := s[p,k,b]-(rbeg[p,a]-s[p,a,b]) ;
                        let s[p,k,v] := s[p,k,v]+rbeg[p,a]-s[p,a,b] ;
                        let s[p,a,v] := s[p,a,v]-(rbeg[p,a]-s[p,a,b]) ;
                        let s[p,a,b] := rbeg[p,a];
                      }
                    }
                  }
                }
              }
            }
          }
        }
      }
    }
  }
}

```

```

else {
    let s[p,k,v] := s[p,k,v]+s[p,k,b] ;
    let s[p,a,v] := s[p,a,v]-s[p,k,b] ;
    let s[p,a,b] := s[p,a,b]+s[p,k,b] ;
    let s[p,k,b] := 0;
}
}
if s[p,k,v] < s[p,k,b] && s[p,k,v] < s[p,a,b] && s[p,k,v] < s[p,a,v] then {
    if s[p,a,v]+s[p,k,v] > rbeg[p,a] then {
        let s[p,k,v] := s[p,k,v]-(rbeg[p,a]-s[p,a,v]) ;
        let s[p,a,b] := s[p,a,b]-(rbeg[p,a]-s[p,a,v]) ;
        let s[p,k,b] := s[p,k,b]+(rbeg[p,a]-s[p,a,v]) ;
        let s[p,a,v] := rbeg[p,a] ;
    }
    else {
        let s[p,a,v] := s[p,a,v]+s[p,k,v] ;
        let s[p,a,b] := s[p,a,b]-s[p,k,v] ;
        let s[p,k,b] := s[p,k,b]+s[p,k,v] ;
        let s[p,k,v] := 0 ;
    }
}
}
if s[p,a,b] < s[p,k,b] && s[p,a,b] < s[p,k,v] && s[p,a,b] < s[p,a,v] then {
    if s[p,k,b]+s[p,a,b] > rbeg[p,k] then {
        let s[p,k,v] := s[p,k,v]-(rbeg[p,k]-s[p,k,b]) ;
        let s[p,a,b] := s[p,a,b]-(rbeg[p,k]-s[p,k,b]) ;
        let s[p,a,v] := s[p,a,v]+(rbeg[p,k]-s[p,k,b]) ;
        let s[p,k,b] := rbeg[p,k] ;
    }
    else {
        let s[p,k,b] := s[p,k,b]+s[p,a,b] ;
        let s[p,k,v] := s[p,k,v]-s[p,a,b] ;
        let s[p,a,v] := s[p,a,v]+s[p,a,b] ;
        let s[p,a,b] := 0;
    }
}
}

```


PERIOD	ARBETSPASS	VÅNING	tid
6.00-8.00	arbetspass_5	våning_1	6,270797427
6.00-8.00	arbetspass_5	våning_2	5,923862053
6.00-8.00	arbetspass_5	våning_3	3,225448264
6.00-8.00	arbetspass_5	våning_4	59,61852413
6.00-8.00	arbetspass_5	våning_5	0
6.00-8.00	arbetspass_5	våning_6	44,96136812
6.00-8.00	arbetspass_7	våning_1	10,05291418
6.00-8.00	arbetspass_7	våning_2	7
6.00-8.00	arbetspass_7	våning_3	0
6.00-8.00	arbetspass_7	våning_4	42,94708582
6.00-8.00	arbetspass_7	våning_5	0
6.00-8.00	arbetspass_7	våning_6	0
6.00-8.00	arbetspass_4	våning_1	24,42770085
6.00-8.00	arbetspass_4	våning_2	0
6.00-8.00	arbetspass_4	våning_3	0
6.00-8.00	arbetspass_4	våning_4	16,58281548
6.00-8.00	arbetspass_4	våning_5	78,42769114
6.00-8.00	arbetspass_4	våning_6	0,561792526
8.00-9.00	arbetspass_5	våning_1	5,965785826
8.00-9.00	arbetspass_5	våning_2	0
8.00-9.00	arbetspass_5	våning_3	0
8.00-9.00	arbetspass_5	våning_4	44,5411208
8.00-9.00	arbetspass_5	våning_5	5,471644143
8.00-9.00	arbetspass_5	våning_6	4,021449236
8.00-9.00	arbetspass_7	våning_1	2,683726199
8.00-9.00	arbetspass_7	våning_2	26,69076258
8.00-9.00	arbetspass_7	våning_3	0,146107275
8.00-9.00	arbetspass_7	våning_4	9,937108183
8.00-9.00	arbetspass_7	våning_5	0,916014853
8.00-9.00	arbetspass_7	våning_6	9,228423835

PERIOD	ARBETSPASS	VÅNING	tid	sorteringsalgoritmens tid
8.00-9.00	arbetspass_4	våning_1	5,188317995	0
8.00-9.00	arbetspass_4	våning_2	0,715721917	0
8.00-9.00	arbetspass_4	våning_3	1,077604011	1,223711286
8.00-9.00	arbetspass_4	våning_4	15,28043805	35,48239202
8.00-9.00	arbetspass_4	våning_5	5,498235975	0
8.00-9.00	arbetspass_4	våning_6	25,9552997	17,00951434
8.00-9.00	arbetspass_10	våning_1	0,047843779	0
8.00-9.00	arbetspass_10	våning_2	23,10398024	23,10398024
8.00-9.00	arbetspass_10	våning_3	23,86383908	23,86383908
8.00-9.00	arbetspass_10	våning_4	2,316412284	0
8.00-9.00	arbetspass_10	våning_5	2,888661351	5,252917415
8.00-9.00	arbetspass_10	våning_6	0	0
9.15-11.30	arbetspass_5	våning_1	0	0
9.15-11.30	arbetspass_5	våning_2	0	0
9.15-11.30	arbetspass_5	våning_3	59,6595977	75,48551159
9.15-11.30	arbetspass_5	våning_4	0,088622837	0
9.15-11.30	arbetspass_5	våning_5	0	0
9.15-11.30	arbetspass_5	våning_6	75,25177946	59,51448841
9.15-11.30	arbetspass_7	våning_1	11,76662126	0
9.15-11.30	arbetspass_7	våning_2	15,26964824	38,34892555
9.15-11.30	arbetspass_7	våning_3	10,93842852	0
9.15-11.30	arbetspass_7	våning_4	0	0
9.15-11.30	arbetspass_7	våning_5	17,48399861	0
9.15-11.30	arbetspass_7	våning_6	79,54130336	96,65107445
9.15-11.30	arbetspass_4	våning_1	0	0
9.15-11.30	arbetspass_4	våning_2	45,13253403	0
9.15-11.30	arbetspass_4	våning_3	0	0
9.15-11.30	arbetspass_4	våning_4	14,26640319	0
9.15-11.30	arbetspass_4	våning_5	47,9301949	79,6805967
9.15-11.30	arbetspass_4	våning_6	27,67086788	55,3194033
9.15-11.30	arbetspass_10	våning_1	12,14631175	23,91293301
9.15-11.30	arbetspass_10	våning_2	45,84825594	67,90151266
9.15-11.30	arbetspass_10	våning_3	4,887485358	0
9.15-11.30	arbetspass_10	våning_4	14,26640319	28,62142922
9.15-11.30	arbetspass_10	våning_5	28,8305283	14,56412511
9.15-11.30	arbetspass_10	våning_6	29,02101545	0

PERIOD	ARBETSPASS	VÅNING	tid	sorteringsalgoritmens tid
12.00-14.00	arbetspass_5	våning_1	2,692167708	0
12.00-14.00	arbetspass_5	våning_2	0	0
12.00-14.00	arbetspass_5	våning_3	114,0415628	120
12.00-14.00	arbetspass_5	våning_4	0	0
12.00-14.00	arbetspass_5	våning_5	0	0
12.00-14.00	arbetspass_5	våning_6	3,266269479	0
12.00-14.00	arbetspass_7	våning_1	1,745387258	0
12.00-14.00	arbetspass_7	våning_2	1,009963948	0
12.00-14.00	arbetspass_7	våning_3	39,30603425	45,52104216
12.00-14.00	arbetspass_7	våning_4	0	0
12.00-14.00	arbetspass_7	våning_5	74,47895784	74,47895784
12.00-14.00	arbetspass_7	våning_6	3,45965671	0
12.00-14.00	arbetspass_4	våning_1	0	0
12.00-14.00	arbetspass_4	våning_2	1,404585401	0
12.00-14.00	arbetspass_4	våning_3	0	0
12.00-14.00	arbetspass_4	våning_4	43,5954146	45
12.00-14.00	arbetspass_4	våning_5	0	0
12.00-14.00	arbetspass_4	våning_6	0	0
12.00-14.00	arbetspass_10	våning_1	39	43,43755497
12.00-14.00	arbetspass_10	våning_2	12,11246498	14,52701433
12.00-14.00	arbetspass_10	våning_3	42,62184049	30,44839539
12.00-14.00	arbetspass_10	våning_4	2,316412284	0,911826883
12.00-14.00	arbetspass_10	våning_5	0	0
12.00-14.00	arbetspass_10	våning_6	18,64482962	25,37075581
14.15-16.00	arbetspass_5	våning_1	6,904124766	6,904124766
14.15-16.00	arbetspass_5	våning_2	0	0
14.15-16.00	arbetspass_5	våning_3	0	0
14.15-16.00	arbetspass_5	våning_4	0	0
14.15-16.00	arbetspass_5	våning_5	3,804676204	3,804676204
14.15-16.00	arbetspass_5	våning_6	34,29119903	34,29119903
14.15-16.00	arbetspass_7	våning_1	0	0
14.15-16.00	arbetspass_7	våning_2	31,26747119	31,26747119
14.15-16.00	arbetspass_7	våning_3	65,62468707	65,62468707
14.15-16.00	arbetspass_7	våning_4	0	0
14.15-16.00	arbetspass_7	våning_5	0	0
14.15-16.00	arbetspass_7	våning_6	0	0
14.15-16.00	arbetspass_10	våning_1	2,692167708	2,692167708
14.15-16.00	arbetspass_10	våning_2	0	0
14.15-16.00	arbetspass_10	våning_3	0	0
14.15-16.00	arbetspass_10	våning_4	42,30783229	42,30783229
14.15-16.00	arbetspass_10	våning_5	0	0
14.15-16.00	arbetspass_10	våning_6	0	0

Bilaga D

AMPL-kod till exempel 1 och exempel 2

Detta är koden till **Exempel 1**

```
set VAROR;  
  
param kost {VAROR} >= 0;  
param vit_a {VAROR} >= 0;  
param vit_b {VAROR} >= 0;  
param vit_c {VAROR} >= 0;  
  
var Buy {v in VAROR} >= 0;  
  
minimize Tot_kost: sum {v in VAROR} kost[v]*Buy[v];  
  
subject to A: sum {v in VAROR} vit_a[v]*Buy[v] >= 700;  
subject to B: sum {v in VAROR} vit_b[v]*Buy[v] >= 700;  
subject to C: sum {v in VAROR} vit_c[v]*Buy[v] >= 700;
```

och detta är koden till **Exempel 2**

```
set VAROR;
```

```
param kost {VAROR} >= 0;
param vit_a {VAROR} >= 0;
param vit_b {VAROR} >= 0;
param vit_c {VAROR} >= 0;

var Buy {v in VAROR} integer >= 0;

minimize Tot_kost: sum {v in VAROR} kost[v]*Buy[v];

subject to A: sum {v in VAROR} vit_a[v]*Buy[v] >= 700;
subject to B: sum {v in VAROR} vit_b[v]*Buy[v] >= 700;
subject to C: sum {v in VAROR} vit_c[v]*Buy[v] >= 700;
```

Litteraturförteckning

- [1] Lundgren, Jan.; Rönnqvist, Mikael. och Värbrand, Peter. *Optimeringslära*, Studentlitteratur, Sverige, Andra upplagan, 2003
- [2] Peterman, Eike. *Optimeringslära och konvexitet*, 1985
- [3] Dantzig, George. *Linear Programming and Extensions*, Princeton University Press, United States of America, sixth printing, 1974
- [4] Bazaraa, Mokhtar S.; Sherali, Hanif D. and Shetty, C. M. *Non Linear Programing and Algorithms*, John Wiley & Sons, United States of America, Second Edition, 1979
- [5] Fourer, Robert.; Gay, David M. and Kernighan, Brian W. *AMPL A Modeling Language for Mathematical Programming*, Thomson, Canada, Second edition, 2003