

**Laborationsuppgift 2**  
**Tillämpad optimeringslära för MMT**  
**(5B1722)**

Februari 2004

Avdelningen för Optimeringslära och Systemteori  
Institutionen för Matematik  
Kungliga Tekniska Högskolan  
Stockholm

## Allmän information

Laborationen utgör ett obligatoriskt moment i kursen. Den syftar till att ge praktisk träning på genomgångna teoriavsnitt. Laborationen består av tre dellaborationer som tar upp linjär- och heltalsprogrammering samt olinjära problem.

## Redovisning

Laborationerna ska utföras individuellt eller i grupper om **högst** två personer. Varje laboration ska redovisas skriftligt.

Anvisningar för rapporter:

- Rapporterna skall inledas med ett försättsblad där gruppmedlemmarnas namn, personnummer och e-postadress tydligt skall anges.
- Rapporterna skall vara skrivna m.h.a. ett lämpligt ordbehandlingsprogram, inte för hand.
- Den MATLAB-kod som har använts för att lösa problemen skall ingå i rapporten (lämpligen som en bilaga).
- Innehållet skall vara sådant att en annan person som går kursen men ej är bekant med laborationsuppgiften skall kunna läsa rapporten och lätt förstå följande:
  1. Vad är problemet? Bakgrunden till att man vill lösa detta problem. Detta innebär *inte* att man skall skriva av uppgiftens beskrivning, utan att man skall göra en lämplig sammanfattning av nödvändig information, som behövs för att förstå problemställningen.
  2. Hur gruppen valt att överföra problemet till en matematisk problemformulering. Vilka antaganden har gjorts? Om dessa antaganden kan antas påverka lösningen bör detta påpekas.
  3. Vad är lösningen till det uppställda optimeringsproblemet? Om det bedöms nödvändigt, skall den matematiska lösningen återföras till den icke-matematiska problemformuleringens terminologi.
- De flesta uppgiftslydelserna innehåller ett antal frågor vilka skall besvaras i rapporten. Svaret på dessa *skall* tydligt framgå i rapporten. De bör dock på ett naturligt sätt vävas in i rapporten och inte radas upp i en "svarslista".

- Ett förslag på lämplig uppläggning av rapporten är följande:
  1. Eventuellt en kort inledande sammanfattning.
  2. Problembeskrivning och bakgrundsinformation.
  3. Matematisk formulering.
  4. Resultat och analys (tolkningar av resultat).
  5. Ett avslutande avsnitt med sammanfattning och slutsatser.

Vi förbehåller oss rätten att kräva muntlig redovisning om vi finner det lämpligt.

Det förutsätts att Ni inom gruppen egenhändigt skriver den MATLAB-kod och rapport som efterfrågas. Det betraktas som **fusk** att kopiera andra grupperns MATLAB-kod, lösningar eller rapporter.

## **Dator**

Laborationerna utförs på dator enligt eget önskemål.

## **Frågor**

Frågor om laborationerna besvaras av:

Claes Trygger (trygger@math.kth.se, tel 790 4919) och Stefan Feltenmark (stefanf@math.kth.se).

# 1 Heltalsprogrammering

Företaget Soda Unlimited Company har bestämt sig för att göra en inbrytning på den prestigefyllda och lukrativa Ruritanska julläskmarknaden. Med hjälp av sin nyutvecklade vita läskeblask vill SUC konkurrera ut såväl den inhemska årgångsdrycken som den internationella storsäljaren. För att kunna etablera sig måste dock SUC investera i ett antal fabriker. Detta glädde den Ruritanske näringsministern Ulf Tulpanstjälk som genast lovade glesbygdsstöd om fabriker placerades på mindre orter. Efter flera fantastiskt lyckade mutresor, bland annat till de välbekanta bruksorterna Annorlunda, Någorlunda och Sålunda, enades den exekutiva kommittén i SUC om  $m$  stycken kandidatorter. Att driva en fabrik på ort  $i$  kostar  $p_i$  Mega Oj'rå per år. På grund av personalbrist kan endast en anläggning finnas på varje ort. Den goda tillgången till de viktigaste råvarorna socker och vatten, medför att en byggd fabrik inte har någon övre produktionsbegränsning. Med god noggrannhet kan efterfrågan på drycken antas komma från de  $n$  största städerna i Ruritanien. Kostnaden för att täcka hela efterfrågan i stad  $j$  från fabrik  $i$  är  $c_{i,j}$  Mega Oj'rå per år.<sup>1</sup>

Er uppgift är att hjälpa SUC att planera investeringarna och transportflödet till lägsta kostnad.

1. Hjälpa SUC att formulera ett **linjärt blandat heltalsprogrammeringsproblem** som bestämmer vilka fabriker som skall drivas och vilken fabrik/vilka fabriker som skall leverera till vilken stad/vilka städer. SUC vill tillfredsställa efterfrågan i städerna till lägsta kostnad. Notera att hela efterfrågan i stad  $j$  inte behöver täckas av en och samma fabrik. I denna modell får endast  $n + m$  bivillkor användas.<sup>2</sup> Redovisa noggrant betydelsen av införda variabler och bivillkor.

Ledning: Om  $0 \leq y_k \leq u_k$  för  $k = 1, \dots, \ell$  och  $x \in \{0, 1\}$  så är följande bivillkor ekvivalenta:

$$(a) \sum_{k=1}^{\ell} y_k \leq x \sum_{k=1}^{\ell} u_k$$
$$(b) y_k \leq u_k x \text{ för } k = 1, \dots, \ell.$$

---

<sup>1</sup>Problem av den här typen kallas *uncapacitated facility location problems*

<sup>2</sup>Denna modell kallas *weak formulation*

2. Lös problemet ovan med MATLAB-funktionen `lips.m` som finns att ladda ned från kursens hemsida:  
<http://www.math.kth.se/optsys/studinfo/5B1722/5B1722kursinfo.html>.  
 Observera att `lips.m` behöver både `alps.m` och `simple.m`.

Kostnaderna ges av:

$$m = 13, n = 9, p = (1 \ 2 \ 3 \ 4 \ 1 \ 2 \ 3 \ 4 \ 1 \ 2 \ 3 \ 4 \ 1)^T,$$

$$C = \begin{pmatrix} 9 & 8 & 2 & 3 & 2 & 1 & 2 & 4 & 5 \\ 4 & 6 & 6 & 5 & 1 & 3 & 1 & 2 & 4 \\ 6 & 3 & 6 & 2 & 5 & 3 & 5 & 2 & 7 \\ 0 & 2 & 1 & 7 & 1 & 4 & 7 & 4 & 5 \\ 1 & 2 & 1 & 4 & 8 & 3 & 2 & 3 & 7 \\ 3 & 4 & 5 & 3 & 0 & 2 & 3 & 7 & 1 \\ 9 & 6 & 3 & 4 & 2 & 7 & 1 & 3 & 4 \\ 2 & 1 & 4 & 3 & 5 & 6 & 4 & 5 & 3 \\ 4 & 6 & 3 & 4 & 4 & 6 & 1 & 8 & 3 \\ 7 & 4 & 3 & 1 & 2 & 5 & 3 & 3 & 2 \\ 3 & 4 & 5 & 6 & 1 & 1 & 4 & 3 & 5 \\ 4 & 6 & 7 & 1 & 2 & 3 & 1 & 7 & 5 \\ 3 & 5 & 6 & 8 & 1 & 2 & 9 & 5 & 3 \end{pmatrix}.$$

Rita sökträdet med hjälp av funktionen `mytreeplot`.

3. Formulera nu istället problemet med  $n + mn$  bivillkor. Redovisa noggrant betydelsen av införda variabler och bivillkor.<sup>3</sup>
4. Lös problemet med  $n+mn$  bivillkor med MATLAB-funktionen `lips.m`.
5. Vilka fördelar finns med den större modellen ? Vilken är den uppenbara nackdelen med den större modellen ?

---

<sup>3</sup>Denna modell kallas *strong formulation*

Lös problemet:

$$\begin{aligned} & \text{maximera } c^T x \\ & \text{då } Ax \leq b, \\ & 0 \leq x_j \leq 1, \quad j = 1, \dots, n \\ & x_j \text{ heltalig, } \quad j = 1, \dots, n \end{aligned}$$

med hjälp av `lips`.

Filen `ex2.m`:

```
m = 2; % Antal bivillkor
n = 9; % Antal variabler

c = -[77 6 3 6 33 13 110 21 47]';
A = [774 76 22 42 21 760 818 62 785
67 27 794 53 234 32 797 97 435];
b = [1500 1500]';

l = zeros(n,1); % Variabelgränser
u = ones(n,1); % Variabelgränser

sense = ones(m,1); % <=
I = 1:n; % Samtliga variabler är binära
opttol = 1.0e-7; % Optimalitetstolerans
maxit = 1000; % Maximalt antal noder i sökträdet
bs = 3; % Branchningsstrategi
prtlvl = 1; % Skriv ut information vid varje nod

[x,ub,lb,bbtree,status] = lips(c,A,b,l,u,sense,I,opttol,maxit,bs,prtlvl);

figure
mytreeplot(bbtree(:,2)) % Visa sökträdet
```

Utskriften från `lips` skall tolkas på följande sätt:

<code>It</code>	Hur många noder har analyserats i sökträdet (iteration).
<code>#act</code>	Antalet aktiva (icke uttömda ) noder i sökträdet (active).
<code>curr</code>	Nodnummer (current).
<code>lb</code>	Målfunktionsvärdet av nodens LP-relaxering (lower bound).
<code>glb</code>	Lägsta målfunktionsvärdet av alla aktiva LP-relaxeringar (global lower bound).
<code>ub</code>	Målfunktionsvärdet av den tillåtna lösning som hittas i noden (upper bound).
<code>gub</code>	Målfunktionsvärdet av den bästa tillåtna lösning som hittats (global upper bound).
<code>gap</code>	Dualitetsgapet.
<code>st</code>	Nodstatus.
<code>dp</code>	Det djup som noden ligger på i sökträdet (depth).