

# Linjärprogrammering - Laboration 1

Vårt dagliga bröd optimerar oss idag

Alexander Gunillasson (alexander.gunillasson@student.umu.se)

Aladdin Persson (aladdin.persson@student.umu.se)

Simon Edman (sied0005@student.umu.se)

Joel Eliasson (joel0089@student.umu.se)

Handledare: Lars-Daniel Öhman

# Innehåll

<b>1</b>	<b>Opponering</b>	<b>2</b>
<b>2</b>	<b>Introduktion</b>	<b>3</b>
2.1	Syfte . . . . .	3
2.2	Teori . . . . .	3
<b>3</b>	<b>Modellbeskrivning</b>	<b>5</b>
<b>4</b>	<b>Utförande</b>	<b>8</b>
<b>5</b>	<b>Resultat</b>	<b>10</b>
<b>6</b>	<b>Diskussion</b>	<b>11</b>
<b>7</b>	<b>Appendix</b>	<b>13</b>
7.1	Appendix Code - lab1.m . . . . .	13
7.2	Appendix opponering . . . . .	14
<b>8</b>	<b>Referenslista</b>	<b>16</b>

# 1 Opponering

Vår granskningsgrupp tyckte överlag att vår rapport var väldigt bra men vi har valt att göra om och utvecklat olika delar av vår rapport efter att ha tagit åt oss av deras kommentarer [7.2](#). De ville bland annat se en tydligare återkoppling till vår teoridel i rapportens gång. Vi har åtgärdat det här problemet genom att försöka ha tydligare koppling till teoridelen. Till exempel försöker vi beskriva att LP-problemet som vi ställer upp i modellbeskrivningen baseras på den allmänna formen som presenteras i Teori-delen. Vi valde även att utöka modellbeskrivningen med en matematisk beskrivning av hur vi löste problemet, som även den baseras på informationen som vi tog upp i Teori-delen. Vi har även tagit bort en del text som de ansåg som överflödig information vilket har gjort rapporten mer informativ. Till följd av rapportgranskningen har vi även utvecklat våra referenser samt försökt göra texten mer lättläst för en utomstående person.

## 2 Introduktion

Baserat på livsmedelsverkets databas över olika livsmedel och näringsinnehåll skall en daglig kost planeras där åtminstone det dagliga referensintaget för alla mineraler och vitaminer skall uppnås. Vidare skall modellen utökas för att ta hänsyn till ytterligare aspekter för att göra det dagliga kostintaget mer trovärdigt. Följaktligen skall den utformade modellen minimera den totala vikten av mat för dagen, vilket medför att modellen beskrivs som ett linjärprogrammeringsproblem (LP) som skall minimeras.

### 2.1 Syfte

Syftet med laborationen är att utveckla förmågan att hantera stora datamängder som sedan kan konkretiseras ned i ett LP-problem. Vidare resulterar laborationen i en förståelse för huruvida LP-problem kan lösas samt tolkas med hjälp av ett programspråk.

### 2.2 Teori

För kunna använda optimeringsmodeller krävs det att något i problemställningen kan variera, vilket definierar variablerna för problemet. Att optimera innebär att bestämma bästa möjliga värden på variablerna som baseras på det givna målet, det vill säga målfunktionen. Vidare ges begränsningarna av värdet på variablerna för ett antal bivillkor.

- alla funktioner  $f, g_1, \dots, g_m$  är linjära funktioner.
- alla variabler är kontinuerliga,  $x \in \mathbb{R}^n$

Ett LP-problem kan skrivas på följande allmänna form:

$$\begin{aligned}\min z &= \sum_{j=1}^n c_j x_j \\ \text{då } \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = 1, \dots, m \\ x_j &\geq 0, \quad j = 1, \dots, n\end{aligned}$$

### 3 Modellbeskrivning

Målet med modellen som vi analyserar är att minimera totalvikten av livsmedlena och säkerhetsställa att intaget innehåller nödvändiga mängder av specificerade näringsämnen.

Livsmedel	Energi (kcal)	Protein (g)	Kalcium (mg)
Havregryn	392	14	7
Kyckling	205	32	12
Standardmjölk	67	3	120
Körsbärspaj	247	2	13
Fläsk och bönor	100	5	31

**Tabell 1** – Livsmedel och näringsinnehåll

Om vi antar att vi har följande livsmedel och information som begränsas av tabell 1, vilket utgår från näringsinnehåll angivet per 100 gram, kan vi ställa oss följande fråga:

*Vad ska vi äta för att minimera totala vikten av livsmedel, men samtidigt lyckas uppfylla kravet att åtminstone få i oss 2000 kcal, 55g protein respektive 800mg kalcium?*

Detta problem går självklart att generalisera om vi hade fler livsmedel att ta hänsyn till och dessutom om vi hade information till fler näringsinnehåll hade vi kunnat skapa fler krav. Notera att för varje livsmedel så är näringsinnehållet angett per 100 gram. Om vi ställer oss frågan ovan och försöker omvandla till matematik får vi följande:

Vi definierar  $x_1, x_2, x_3, x_4, x_5$  som antalet 100 gram av havregryn, kyckling, standardmjölk, körsbärspaj respektive fläsk och bönor. Vidare kan vi ställa upp följande linjärprogrammeringsproblem där målfunktionen är totala vikten livsmedel som vi vill minimera. Det här LP-problemet baseras på den allmänna formen som defi-

nieras i Teori-delen.

$$\begin{aligned} \text{minimera} \quad & z = x_1 + x_2 + x_3 + x_4 + x_5 \\ \text{m.a.p} \quad & 392x_1 + 205x_2 + 67x_3 + 247x_4 + 100x_5 \geq 2000 \\ & 14x_1 + 32x_2 + 3x_3 + 2x_4 + 5x_6 \geq 55 \\ & 7x_1 + 12x_2 + 120x_3 + 13x_4 + 31x_6 \geq 800 \\ & x_1, x_2, x_3, x_4, x_5 \geq 0. \end{aligned}$$

Detta innebär alltså att om  $x_1 = 1$ , tar vi 100 gram av havregryn, eller  $x_2 = 0.5$  innebär att vi tar 50 gram av kyckling. Variabeln som vi vill minimera,  $z$ , det vill säga målfunktionen är summan av alla beslutsvariabler,  $x_i$  och innebär alltså att den beskriver summan av antalet 100 gram från respektive livsmedel. Om vi ska omvandla till gram får vi alltså att detta blir 100 gånger värdet på  $z$ .

$392x_1 + 205x_2 + 67x_3 + 247x_4 + 100x_5 \geq 2000$  beskriver att för varje 100 gram av havregryn får vi i oss 392 kcal. Därmed om vi summerar vad vi ätit av respektive livsmedel och multiplicerar med kcal/100 g i livsmedlet får vi totala kalorieinnehållet i dagsinnehållet. Detta har vi sagt ska vara minst 2000 kcal, eller mer och därmed får vi större eller likamed tecken i vår matematiska modell. På liknande sätt får vi fram resterande krav för protein och kalcium. Notera att enheten inte spelar någon roll eftersom livsmedlet är i samma enhet som kravet.

Detta exempel förtydligar hur man i matematiska termer beskriver vår modell i form av ett LP-problem och vad variablerna betyder samt vad det målfunktionen är, som i detta fall är att minimera totalvikten av livsmedlen. Det blir också tydligt att vi kan lägga till fler livsmedel och innebörden blir endast att vi skulle utöka med en ny variabel  $x_6, x_7, \dots, x_n$ , beroende på hur många vi lägger till. Vidare går det med tillgång till mer information om exempelvis vitamininnehåll, mineralinnehåll och fettinnehåll att skapa fler bivillkor eller krav på vårt LP-problem.

I vårt LP-problem för denna rapport har vi använt en databas med information om över 2100 livsmedel samt över 20 bivillkor för kalorier, protein, fett, och för

olika vitaminer och mineraler. Vår målfunktion som är totaltvikten av alla livsmedel formuleras på liknande sätt som för exemplet gett ovan, bara att precis som nämnts tidigare har vi utökat modellen med fler variabler samt utökat antalet bivillkor. Nedan följer även en matematisk beskrivning av hur vi valde att applicera definitionen av det allmänna LP-problemet från Teori-delen på vårt problem:

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

då

$$\sum_{j=1}^n a_{1j} x_j \leq b_1 \quad (2)$$

$\vdots$

$$\sum_{i=1}^n a_{25j} x_i \leq b_{25} \quad (26)$$

Ovan kan vi tolka (1) som målfunktionen som är totala vikten av alla livsmedel vilken vi vill minimera.  $n$  beskriver antalet livsmedel. Vidare beskriver olikheterna (2) till (26) antalet bivillkor som vi valt att ta hänsyn.  $a$  är en konstant som beskriver mängden näringsinnehåll för respektive livsmedel samt så  $b$  de rekommenderade dagliga referensintaget för respektive näringsämne.



## 4 Utförande

För att lösa detta linjärprogrammeringsproblem har vi använt oss av programvaran MATLAB. MATLAB har en inbyggd LP-lösare vilket gör det väldigt smidigt att använda när vi väl har den data vi behöver.

Första steget var att läsa in två separata excel-filer. Den ena, en sammanställning som livsmedelverket skapat av olika livsmedel samt dess innehåll som finns att ladda ned på deras hemsida [2]. Den andra sammanställde vi utifrån livsmedelverkets lista av dagliga referensintag av viktiga mineraler och vitaminer som även denna återfinns på deras hemsida [3].

Dessa två filer läses av som vardera tabell. Innan data från filerna kunde användas började vi med att ta bort onödig data från filerna som inte kommer användas. När de båda filerna rensats upp matchas de genom att sortera tabellen för dagligt referensintag för att få en likadan struktur på de två tabellerna. Exempelvis, antag att kolhydrater har kolumnindex 5 i filen innehållande alla livsmedel. Vi sorterar då i filen för dagligt referensintag vilket medför att kolhydrater placerats på radindex 5. Detta upprepas för alla rader i filen för dagligt referensintag. Efter detta är gjort görs en slutlig rensning där vi bortser från allt innehåll som inte är siffrvärden och lägger kvarvarande värden i en matris. Detta görs för båda inlästa tabeller.

När indata rensats och sorterats ut konverteras matrisen innehållande alla livsmedel till en array. MATLABs LP-lösare kan endast lösa optimeringsproblem på formen (som återfinns i *Teori*):

$$\begin{aligned}a_{ij}x_j &\leq b_i, & i &= 1, \dots, m \\ x_j &\geq 0, & j &= 1, \dots, n\end{aligned}$$

Därav multipliceras matrisen elementvis med -1 för att vända på villkoret eftersom vi är intresserade av fallet då bivillkoren åtminstone skall uppfylla ett visst villkor. Vi tittar även efter index som innehåller "NaN"-värden och ersätter dessa med 0 istället. Samma procedur upprepas för tabellen innehållande dagligt referensintag

av vitaminer och mineraler.

Slutligen när all indata processats används MATLABs inbyggda LP-lösare "linprog". För att använda denna funktion skapar vi en rad variabler som tas som inparametrar till LP-lösaren. Resultatet skrivs ut till en textfil som redovisas i tabell [2](#).

## 5 Resultat

Nedanstående tabell redovisar vilka livsmedel som skall ätas, mängden av vardera livsmedel och det totala intaget. Detta är en sammanställning från det textdokument "Result.txt" som värdena återfinns i vid körning av vår lösning.

Livsmedel	Mängd (gram)
Mjölkpulver fett 1%	21.18
Nyponsoppa pulver berik	18.31
Vällingpulver majs berik	16.72
Gelatinblad gelatinpulver	59.71
Jordnötter rostade saltade	60.64
Paranötter	0.0102
Popcorn mikro fett 22%	37.32
Mjölchoklad m hackade hasselnötter Schweizernöt	43.84
Måltidsersättning chokladsmak pulver Nutrilett Intensiv	78.17
Sorbitol m sackarin	211.1
Joderat salt	0.555
Rapsolja	29.15
Totalt intag	576.7449

**Tabell 2** – Resultat vid körning

## 6 Diskussion

Vårt resultat är för de allra flesta inte en realistisk kost, men om målet är att ha en kost med otroligt mycket energi och tillräckligt med näring för att uppnå grundnivån på mineral och vitaminer samtidigt som man vill äta så lite som möjligt kanske det kan tyckas vara rimligt. Vidare har vi valt att utveckla modellen så den även tar hänsyn på hur mycket kolhydrater, proteiner, fetter samt fibrer en vuxen man i genomsnitt behöver för att göra modellen trovärdigare [1].

I resultatet ingår däremot inte vatten som man också skulle behöva blanda ut pulver med. Vatten/vätska är också en väldigt viktig beståndsdel som kroppen behöver, att endast äta denna kost som består av mycket pulver och lite vätska skulle kunna leda till uttorkning. Att addera 100g kranvatten [4] till menyn skulle i det stora hela inte påverka intaget av mineraler och vitaminer allt för mycket men givetvis öka totalvikten. Det innehåller små mängder av exempelvis mineralerna Kalcium, Kalium, Magnesium och Natrium, så det kan vara något man väljer att lägga till för att förgylla matupplevelsen.

För att få en mer realistisk kost kan ett alternativ vara att sortera ut lite av pulverkosten i excel-filen för försöka optimera näringsintaget utifrån en lite mer ”normal” kost. De olika livsmedel vi fått fram i vårt resultat är teoretiskt en väldigt bra kost om man vill få i sig alla näringsämnen samtidigt som totalvikten minimeras. Hade detta problem lösts utifrån mer traditionell kost så som kyckling, pasta, ris etc. blir det givetvis svårare att hålla totalvikten nere samtidigt som man vill få i sig alla näringsämnen.

En realistisk kost om man försöker tänka på miljöpåverkan och etiska anledningar är till stor del en subjektiv fråga. Mer tydligt blir det med miljöpåverkan eftersom det är mätbart hur mycket olika livsmedel kostar i koldioxidutsläpp för att producera. Man hade därmed kunnat ha ett extra bivillkor där man säger att summan av utsläppen av livsmedel som vi valt i vår kost inte får vara större än ett visst värde. Detta skulle gå att komma fram till från en vetenskaplig synpunkt eftersom det faktiskt är mätbart.

Svårare blir det att analysera etiska anledningar eftersom att det skulle det variera från individ till individ. Vissa äter inget kött alls och är vegetarian, andra vegan och andra äter mer kött. Det finns en stor spridning i kost och objektivt inget rätt svar som det skulle finnas med miljöpåverkan. Det skulle därför vara svårt att skapa ett bivillkor för detta eftersom vi hade behövt anpassa den till en specifik individ.

Avslutningsvis bör det poängteras att vårt resultat inte är helt korrekt då det fanns sju ämnen i excelfilen som inte återfanns på livsmedelsverkets hemsida gällande det dagliga referensintaget för mineraler och vitaminer. Det medför att vår svar inte tar hänsyn till följande ämnen:

- Krom
- Molybden
- Fluorid
- Mangan
- Klorid
- Pantotensyra
- Biotin

## 7 Appendix

### 7.1 Appendix Code - lab1.m

Källkod for lab1.m.

```
clear all; close all; clc

req = readtable('vitamin_mineraler.xlsx');
foods = readtable('livsmedel.xlsx');

% clean, remove unnecessary data
foods(:, [2,4,9,10,11,12,13,14,15,16,17:35,37,45,56,57,60,61]) = [];
req([1,2,14,15,17,24,25,27,28], :) = [];

% make rows, columns match with foods and requirements
req(1:25,:) = req([21,22,23,24,25,1,2,3,4,6,7,5,8,9,11,10,...
    14,20,15,13,12,18,16,19,17],:);

% A matrix remove unnecessary column and make table->array
A = foods(1:end, 2:end);
A = table2array(A);

% linprog requires Ax <= b and make all NaNs to 0
A = A' .* (-1);
A(isnan(A)) = 0;

b = req(1:end, 2:end);
b = str2double(table2array(b)) .* (-1);

% linprog
f = ones(1,2111);
Aeq = [];
beq = [];
```

```
lb = zeros(1,2111);
ub = [];
[x, fval] = linprog(f,A,b,Aeq,beq,lb,ub);

% in column 1 of foods we have the food names
food_names = table2array(foods(:,1));

% find the names of the foods we should eat,
% the rowindex for each food in livsmedel.xls
% and the amount. Write to file Result.txt
food = string(food_names(find(x)));
foodIndex = find(x)+1;
amount = x(find(x)) .* 100;
fileID = fopen('Result.txt','w');

% Loop to write all data to result.txt
for i = 1:length(amount)
    fprintf(fileID,'(%d) %s %s\t %0.4f\n',foodIndex(i),...
        food(i),':', amount(i));
end
fprintf(fileID,'\n%s %12.4f','Totalt intag:',fval .* 100);
fclose(fileID);
```

## 7.2 Appendix opponering

# Opponering

## Inledning och syfte

- Kortfattat och ger bra överblick av vad rapporten ska handla om.

## Teori

- Vi ser gärna att man gör teorin mer anpassad efter läsaren. Förklara variablerna mer (f, g, a, b, c osv) utförligt så att exempelvis någon som inte utfört labben får en inblick och inte blandar ihop variablerna.
- Om ni använder er av definitioner, skriv gärna utförligt om det är just en definition så läsarna får en känsla för det och eventuellt kan återkoppla till kursmaterialet.

## Modellbeskrivning

- Första stycket hör mer hemma i inledning än i modellbeskrivningen. Lite överflödigt eventuellt!
- Bra att ni använder er av uppvärmningsproblemet och refererar till denna!

## Utförande

- Ni skriver optimeringsproblemet på formen  $Ax < b$ , vilket ni inte hänvisar till eller definierar i teoridelen.
- Istället för att skriva "här" bör man referera till referenslistan och inte till länkar.

## Resultat

- En liten grej vi tänkte på var att ni skriver livsmedelsprodukternas indexnummer före dess namn, vilket kan vara lite otydligt för en som inte sett excelfilen och vet vad dessa siffror står för. Annars bra!

## Diskussion

- Bra aspekt att ta upp om 100 g vatten! Bra diskuterat om att vissa livsmedel är mer realistiska (kyckling, pasta osv) men att detta skulle medföra en större totalvikt.

## Språket

- Väldigt välskriven och ni använder er av ett enkelt språk och är pedagogiska!
- Särskrivning i rubriken (Linjär Programmering).



## 8 Referenslista

### Referenser

- [1] Wendy Bumgardner. “Daily Diet Composition Charts for Carbs, Protein, and Fat”. I: 2018.
- [2] Livsmedelsverket. “<http://www7.slv.se/SokNaringsinnehall>”. I: 2017.
- [3] Livsmedelsverket. “<https://www.livsmedelsverket.se/livsmedel-och-innehall/text-pa-forpackning-markning/naringsvarde>”. I:
- [4] Matkalkyl. “<http://matkalkyl.se/se-kranvatten.php>”. I: