

Linjär optimering

Exempel på användning av analoga och digitala verktyg i undervisningen

Kursavsnittet linjär optimering i Matematik 3b kan introduceras med såväl analoga som digitala verktyg. I artikeln beskriver artikelförfattaren utöver en traditionell presentation på tavlan även en fysisk trämodell samt datorbaserad visualisering och programmering.

Optimering av olika slag används inom så vitt skilda områden som produktionsplanering, transport och logistik, telekommunikation, trafikplanering och strukturanalys. I ämnesplanen för Matematik 3b anges en punkt under det centrala innehållet *Samband och förändring* som innefattar linjär optimering: "Användning av begreppet geometrisk summa samt linjär optimering i tillämpningar som är relevanta för karaktärsämnen."

Kursavsnitt om linjär optimering presenteras traditionellt i form av en lärarledd genomgång vid tavlan följt av elevernas arbete med problemlösning. En traditionell genomgång över ämnesområdet kan emellertid uppfattas som ganska teoretisk av eleverna och det är många abstrakta steg i beräkningarna. Det kan därför vara en fördel att även visualisera och konkretisera kursinnehållet på olika sätt, tex genom att använda ett analogt hjälpmedel som en fysisk trämodell och ett digitalt hjälpmedel som Geogebra (www.geogebra.org) eller Desmos (www.desmos.com/calculator). Presentationen av kursmaterialet kan med fördel även delas upp på flera lektioner som att exempelvis börja med att grafiskt representera olikheter i ett koordinatsystem, sedan fortsätta med att repetera lösning av ekvationssystem för att därefter introducera målfunktioner i två variabler och utvärdera dessa i olika punkter.

Programmering underlättar beräkningar

För att underlätta beräkningen av målfunktionens värden i olika punkter kan programmering med exempelvis programspråken Python 3 (se tex <https://repl.it>) eller Octave (se tex <http://octave-online.net>) användas. Detta ligger även i linje med den nya reviderade ämnesplanen för matematik där kursen Ma3b innehåller programmering som en punkt under det centrala innehållet *Problemlösning*: "Strategier för matematisk problemlösning inklusive modellering av olika situationer, såväl med som utan digitala verktyg och programmering."

Problemställning

Problemet består av att bestämma största och minsta värdet för en målfunktion, tex:

$$z = m(x, y) = 3 + \frac{1}{2}x + \frac{1}{4}y$$

inom området som definieras av olikheterna (bivillkoren)

$$\begin{cases} x \geq 0 \\ y \geq 0 \\ y \leq -x + 6 \\ y \leq -\frac{1}{2}x + 5 \end{cases}$$

Traditionell undervisning

Problemet löses traditionellt genom att det tillåtna området markeras grafiskt i ett koordinatsystem där områdena som inte uppfyller olikheterna skuggas. Skärningspunkterna mellan linjerna beräknas sedan, varvid områdets hörnpunkter bestäms genom att lösa de fyra ekvationssystemen:

$$\begin{cases} x = 0 \\ y = 0 \end{cases}$$

$$\begin{cases} x = 0 \\ y = -\frac{1}{2}x + 5 \end{cases}$$

$$\begin{cases} y = -x + 6 \\ y = -\frac{1}{2}x + 5 \end{cases}$$

$$\begin{cases} y = 0 \\ y = -x + 6 \end{cases}$$

Ekvationssystemen har lösningar i form av koordinaterna: (0,0), (0,5), (2,4) och (6,0). Dessa koordinater används sedan för att beräkna målfunktionens värde för de olika punkterna:

$$\begin{cases} m(0,0) = 3 \\ m(0,5) = 4,25 \\ m(2,4) = 5 \\ m(6,0) = 6 \end{cases}$$

Av beräkningarna framgår att målfunktionens största värde är 6 och fås i punkten (6,0) och minsta värdet är 3 och fås i origo.

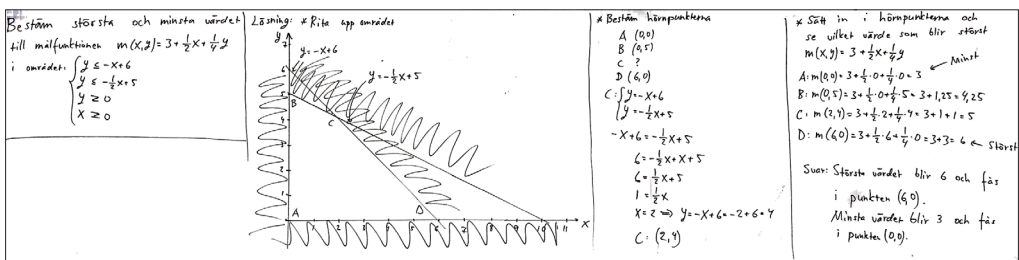
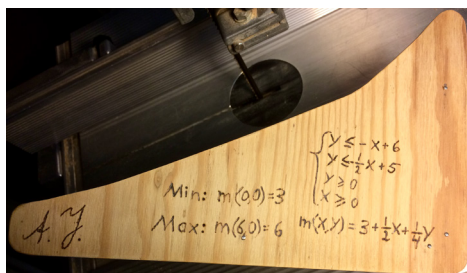
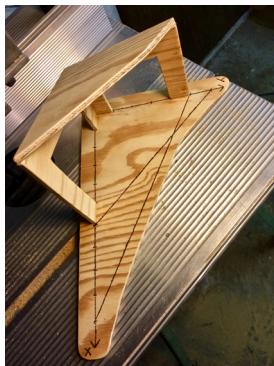


Foto från whiteboardtavlan efter en traditionell lektionsgenomgång.
På Nämnaren på nätet finns en större bild.

Fysisk modell i trä

För att visualisera problemet och även motivera varför endast målfunktionens värden i hörnpunkterna behöver bestämmas, tillverkades en modell i trä.



Trämodell för visualisering av målfunktion och bivillkor, olika vyer.

Problemställningens målfunktion spänner vid linjär optimering upp ett plan i rummen. Detta plan kan endast ha största respektive minsta värde i någon av hörnpunkterna. På undersidan av trämodellen är problemet och dess lösning inskrivna. Det finns en didaktisk tanke med detta, nämligen att trämodellen kan användas inte bara för visualisering utan även för en svårare problemställning av typen:

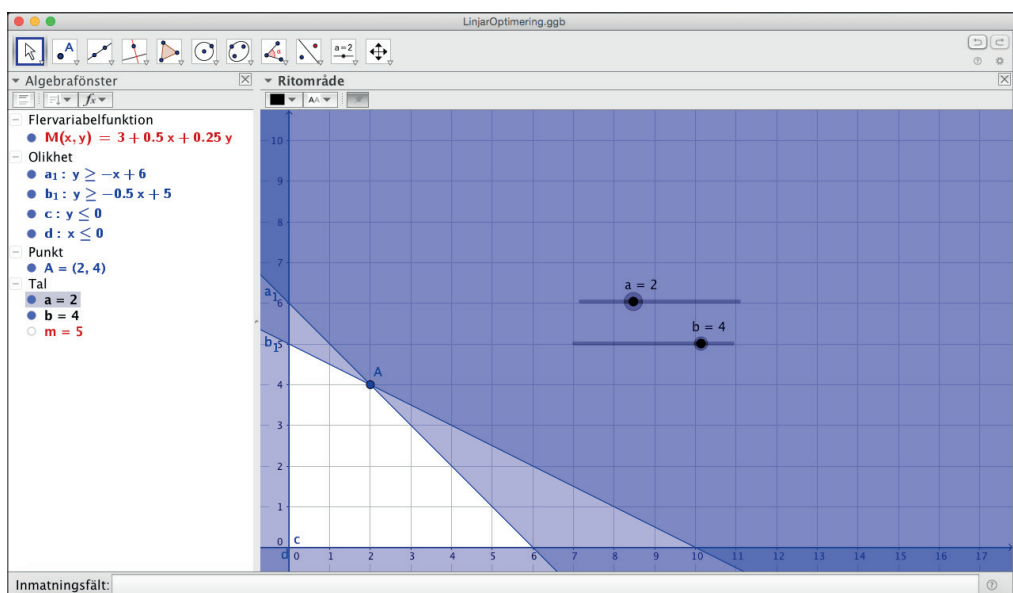
Studera modellen och bestäm bivillkor och målfunktion genom att mäta på modellen.

Målfunktionens exakta värden fås genom mätning från bottenplattans översida till målfunktionsplattans undersida. Då det inte finns några numeriska värden inskrivna på koordinataxlarna, kan det finnas flera möjliga lösningar på problemet.

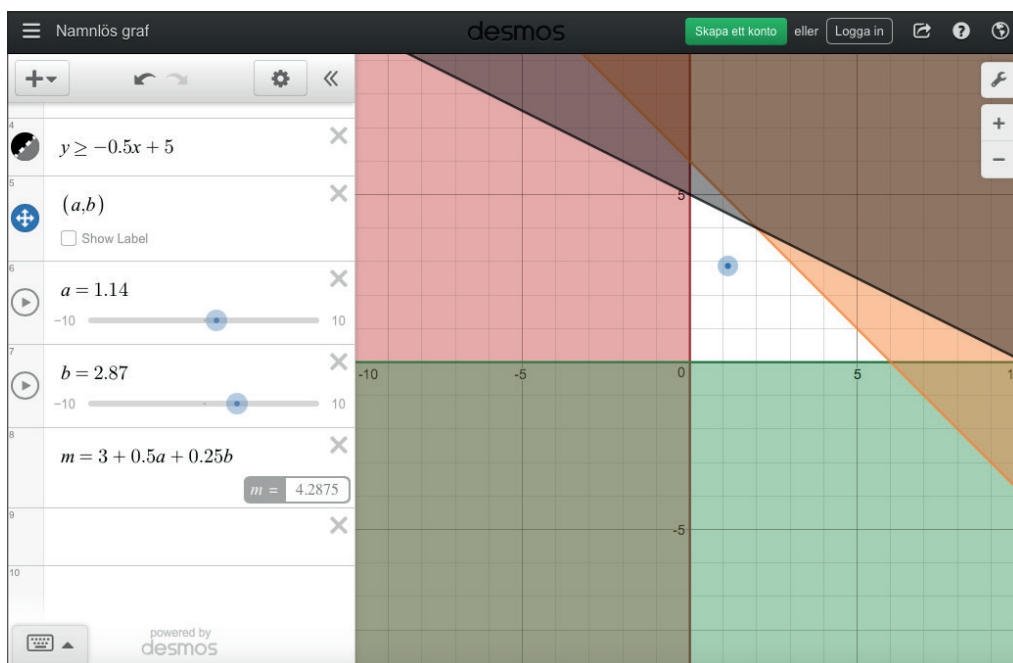
Virtuella modeller med Geogebra och Desmos

Problemet kan även lösas på ett smidigt sätt i någon av programvarorna Geogebra eller Desmos. En bild från Geogebra-lösningen och en bild från motsvarande lösning i Desmos visas på nästa sida.

Olikheterna och målfunktionen matas in i inmatningsfältet i nedre delen av Geogebra-fönstret respektive i vänstra "kolumnen" i Desmos-fönstret. En punkt A med koordinaterna (a, b) införs, vilken kan förflyttas med muspekaren eller med "glidarna" för a och b . Då punkten A flyttas, visas värdet på målfunktionen som värdet för m , exempelvis $m(2; 4) = 5$ och $m(1; 14; 2; 87) = 4; 2875$.



Lösning med Geogebra ovan och Desmos nedan.



Programmering

För att illustrera vad som är möjligt att göra när programmering ska implementeras i Ma3b inom en snar framtid, redovisas ett par enkla script för att:

1. Beräkna hörnpunkter från ett par givna olikheter.
2. Beräkna målfunktionen för hörnpunkterna.

Den numeriska implementeringen kan tex göras i programspråken Python3 eller Octave som båda kan köras direkt i en webbläsare. Programvarorna är enkla att använda för nybörjare på gymnasienivå, men har också en stor potential för mer avancerade beräkningar. Flera matematiska verktyg finns inbyggda, såsom fördefinierade funktioner, matrisalgebra, numerisk integration och numerisk lösning av differentialekvationer. En mer avancerad programmerbar algoritm för linjär optimering redovisas i boken *Linjär och icke linjär optimering*.

```

1 # Program för att lösa ett linjärt ekvationssystem på formen
2 # y=ax+b
3 # y=cx+d
4
5 # Välj värden på koefficienterna a, b, c och d
6
7 a=-1
8 b=6
9 c=-0.5
10 d=-5
11
12 #a=0
13 #b=0
14 #c=-1
15 #d=6
16
17 if a==c:
18     print('Parallella linjer, det finns ingen lösning!')
19 else:
20     # Beräkning av lösningarna med en allmän formel
21     x=(d-b)/(a-c)
22     y=(a*d-b*c)/(a-c)
23
24 # Skriv ut lösningarna
25 print('x=',x)
26 print('y=',y)
27 #endif
28

```

Python 3.6.1 (default, Dec 2015, 13:05:11)
[GCC 4.8.2] on linux
x= 2.0
y= 4.0

Bild från användargränssnittet för repl.it ovan och octave-online.net nedan.

OCTAVE ONLINE

Files: linopt.m

```

1 %------%
2 % Linjär optimering %
3 %
4 % Anders Johansson %
5 % Falkenberg gymnasieskola %
6 % 2017-05-01 %
7 %------%
8
9 % Städa bort gamla variabler och bilder %
10 clear %
11 clf %
12
13 % Definiera målfunktion %
14 function z=z(x,y) %
15 z=3+0.5*x+0.25*y; %
16 end %
17
18 % Ange punkterna att undersöka i form av vektorer %
19 xx=[0, 0, 2, 6]; %
20 yy=[0, 5, 4, 0]; %
21
22 % Loop (smorra) för att på ett smidigt sätt beräkna %
23 % målfunktionens värde för alla hörnpunkterna %
24 for k=1:4 %
25     xx(k); %
26     yy(k); %
27     zz(k)=z(xx(k),yy(k)); %
28 end %
29
30 % Beräkna största och minsta värde ,samt ange för %
31 % vilket index %
32 [z_max max_index]=max(zz); %
33 [z_min min_index]=min(zz); %
34
35 % Skriv ut största värde och vilken koordinat %
36 disp('=====') %
37 disp(['Största värde:',num2str(z_max)]) %

```

Vars: # ans, # k, # max_index, # min_index, [1x4] xx, [1x4] yy, # z_max, # z_min, [1x4] zz

```

octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4>
octave:4> linopt
=====
Största värde:6
Punkt:
x-värde:6
y-värde:0
=====
Minsta värde:3
Punkt:
x-värde:0
y-värde:0
=====
octave:5> zz
zz =
    3.0000    4.2500    5.0000    6.0000

```

LITTERATUR

Lundgren, J., Rönnqvist, M. & Värbrand, P. (2001). *Linjär och icke linjär optimering*. Lund: Studentlitteratur.