Linear Programming: Giapetto's Workshop

File: `Ch19_Giapetto_sui.m`

This file demonstrates the formulation and solution of simple linear programs in Matlab. To use this demo you will need --

1. Optimization Toolbox: This is included in the version of Matlab distributed to ND students.
2. CVX: This is an optimization modeling language. If you haven't already done so, you'll need to download and install CVX from <http://cvxr.com> .
3. `displaytable.m` This is a simple file for formatting data tables. It's located in the course folder labelled 'Matlab Files'.

Contents

Initialize the workspace to a known state.

```
close all
clear all
clc
addpath('utilities')
```

Giapetto's Workshop (from Winston, Operations Research)

Giapetto's Workshop produces two types of wooden toys:

- Soldiers - Each sells for $27, and requires $10 of raw materials, and $14 of labor consisting of 2 hours of finishing labor and 1 hour of carpentry labor.
- Trains - Each sells for $21, and requires $9 of raw materials, and $10 of labor consisting of 1 hour of finishing labor, and 1 hour of carpentry labor

Weekly production is constrainted by the weekly availability of no more than 100 finishing hours and 80 carpentry hours. At most 40 toy soldiers will be sold each week.

What is the maximum achievable weekly profit?

Analysis

- The decisions variablse will be the number of soldiers produced, `x(1)`, and the number of trains produced, `x(2)`.
- Each soldier sells for $27 and costs $24 to produce a $3 profit. Each train sells for $21 and costs $19 to produce a $2 profit.
- Both `x(1)` and `x(2)` must be non-negative. These form lower bounds on the decision variables.
- There are three constraints on `x(1)` and `x(2)` based on demand and the availability of carpentry and finishing labor.

Solution using `linprog`

The Optimization Toolbox in Matlab provides a linear programming function call `linprog` to **minimize** linear objectives subject to linear constraints. Check the documentation ('doc linprog') for usage. It's quicker and more efficient than CVX, but requires you to create matrices and vectors containing the problem data.

For the Giapetto's workshop problem, let `x` be a two element column representing Soldiers and Trains. Then

```
            minimize f'*x
subject to
                x >= lb
              A*x <= b
```

```
f = [-3 -2];
lb = [0; 0];
A = [1 0; 1 1; 2 1];
b = [40; 80; 100];

x = linprog(f,A,b,[],[],lb,[]);

disp('Solution to Giapetto''s Workshop using linprog.');
displaytable(x,{'Soldiers','Trains'});
```

```
Optimization terminated.
Solution to Giapetto's Workshop using linprog.
Soldiers  20
Trains    60
```

Interactive Demo

```
dval =  40;
cval =  80;
fval = 100;

f = [-3 -2];
lb = [0; 0];
A = [1 0; 1 1; 2 1];
b = [dval; cval; fval];

x = linprog(f,A,b,[],[],lb,[]);

clf;
subplot(8,1,1:7);
hold on;
    demand = plot([dval dval],[0 100],'b','LineWidth',2);
    carpentry = plot([cval 0],[0 cval],'r','LineWidth',2);
    finishing = plot([fval/2 0],[0 fval],'g','LineWidth',2);
    optimum = plot(x(1),x(2),'r.','Markersize',30);
    [X1,X2] = meshgrid(0:100,0:100);
    contour(X1,X2, 3*X1 + 2*X2,'ShowText','on');
hold off;
```

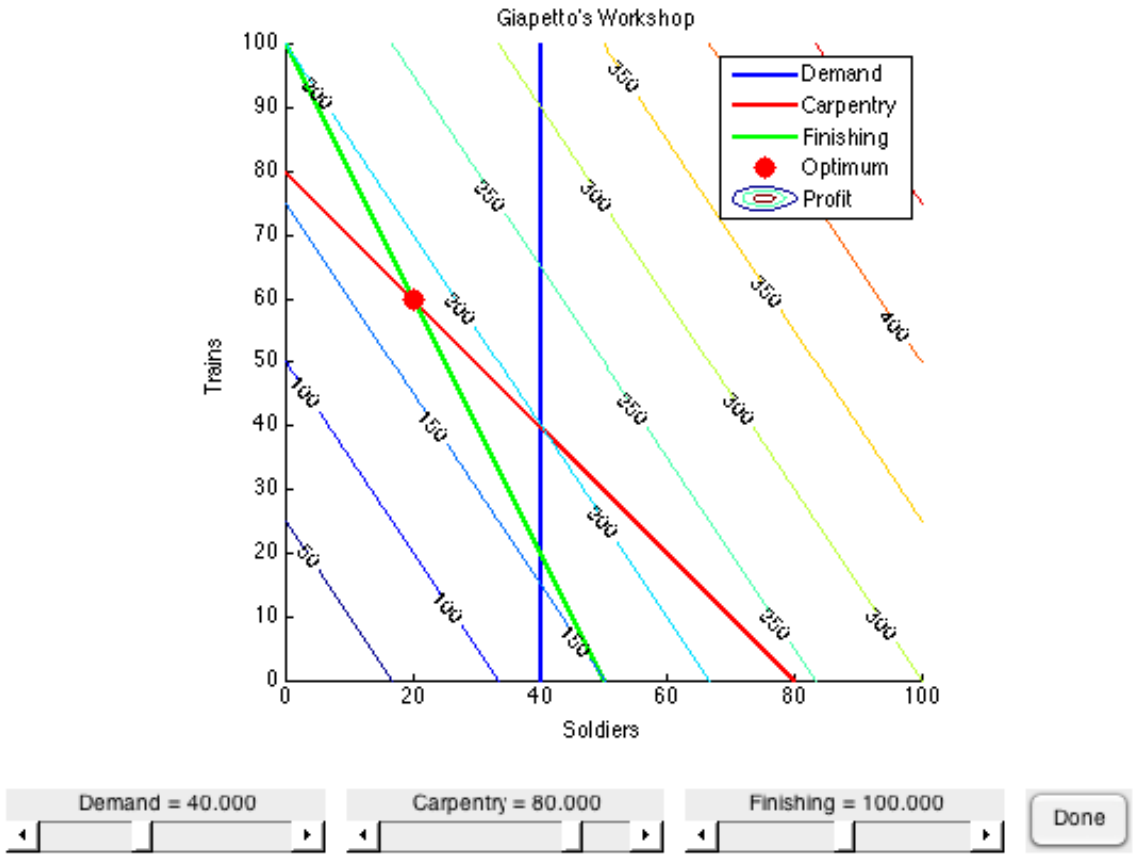
```
legend({'Demand','Carpentry','Finishing','Optimum','Profit'});
axis([0 100 0 100]);
axis 'square';
xlabel('Soldiers');
ylabel('Trains');
title('Giapetto''s Workshop');

update = @(dval,cval,fval) { ...
    suiPlot(demand,[dval dval],[0 100]);
    suiPlot(carpentry,[cval 0],[0 cval]);
    suiPlot(finishing,[fval/2 0],[0 fval]);
    suiPlot(optimum,linprog(f,A,[dval;cval;fval],[],[],lb,[])')};

[d,loc] = suiSlider(0,100,dval,'Demand',[10,10],150);
[c,loc] = suiSlider(0,100,cval,'Carpentry',loc.Right,150);
[f,loc] = suiSlider(0,200,fval,'Finishing',loc.Right,150);
suiButton(@()close('all'),'Done',loc.Right,50);

suiUpdate(@()update(d(),c(),f()));
```

```
Optimization terminated.
```



Published with MATLAB® R2013b