Maple and Basic Linear Programs

Maple can be used to visualize the feasible set for linear programs, and solve them. Here are some example problems, where the solutions are done in Maple.

1. Plot the feasible region:

$$\begin{array}{llll} 8x_1 & +2x_2 & \leq 16 \\ 5x_1 & +2x_2 & \leq 12 \\ x_1, & x_2 & \geq 0 \end{array}$$

   SOLUTION: Maple uses the command `inequal` to plot these regions.

```
with(plots):
inequal( { 8*x+2*y<=16, 5*x+2*y<=12,x>=0, y>=0}, x=-1..4, y=-1..8,
         optionsfeasible=[color="Khaki"]);
?plot/colornames
```

   In this code, I colored in the feasible region using khaki- A listing of all the pre-named colors are listed in the help file, given in the last line of code.

2. Plot the feasible region and solve the linear program:

$$\begin{array}{llllll} (LP) & z = \max & 4x_1+ & x_2 & \\ & \text{s.t.} & 2x_1+ & x_2 & \leq 6 \\ & & x_1+ & 3x_2 & \leq 9 \\ & & x_1, & x_2 & \geq 0 \end{array}$$

   SOLUTION: We'll assign the set of inequalities to a variable.

```
with(simplex);
#Define the objective function and constraints
objFn:=4*x[1]+2*x[2];
C:={2*x[1]+x[2]<=6, x[1]+3*x[2]<=9, x[1]>=0, x[2]>=0};
#Plot the feasible set
inequal(C,x[1]=-1..5,x[2]=-1..5);
#Solve:
maximize(objFn,C);
```

   NOTE: As an alternative, we do not need to list each variable as non-negative. Instead, we could have typed:

```
objFn:=4*x[1]+2*x[2];
C:={2*x[1]+x[2]<=6, x[1]+3*x[2]<=9};
maximize(objFn,C,NONNEGATIVE);
```

A word of warning: Even if there are multiple solutions, Maple may return only one value.

3. Reproduce Figure 5, p. 65 of our text.

$$\begin{array}{rll} \max & z = 3x_1 + 2x_2 & \\ \text{s.t.} & \dfrac{1}{40}x_1 + \dfrac{1}{60}x_2 & \leq \quad 1 \\ & \dfrac{1}{50}x_1 + \dfrac{1}{50}x_2 & \leq \quad 1 \end{array}$$

```
with(plots):
F1:=3*x[1]+2*x[2];
A:=contourplot(F1,x[1]=-1..60,x[2]=-1..65,
       contours=[60, 100, 120],linestyle=dot,color=green):
C:={(1/40)*x[1]+(1/60)*x[2]<=1,(1/50)*x[1]+(1/50)*x[2]<=1,
       x[1]>=0,x[2]>=0};
B:=inequal(C,x[1]=-1..60,x[2]=-1..60,optionsfeasible=[color="white"]):
display(A,B);
```

We notice that the contour of 120 is hidden by the constraint.