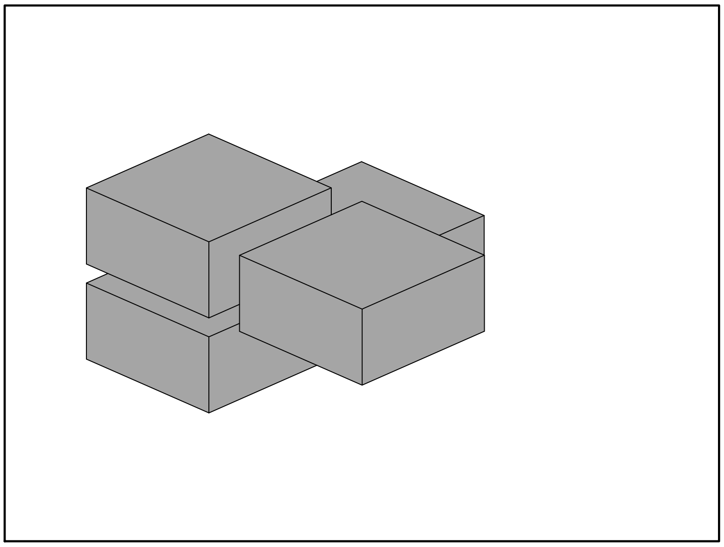
Linear Programming Worksheets

Carl Eberhart, Dept of Math, Univ. of Ky.



July 29, 1997

Contents

1 Calendar of class days: 3

2 A Short Introduction to the Maple Language 4
- 2.1 Arithmetic 4
- 2.2 Expressions, Names, Statements, and Assignments 4
- 2.3 Functions 5
- 2.4 Built in Maple functions and Operations with Functions 6
- 2.5 Using Maple as a fancy graphing calculator. 6
- 2.6 Data types, Expression Sequences, Lists, Sets, Arrays, Tables: 7
- 2.7 Maple control statements 9
- 2.8 A Brief Vocabulary of Maple Words 10
- 2.9 Trouble Shooting Notes 12

3 A look at the graphical method for problems with two decision variables. 13

4 The Wyndor Glass Problem 15

5 Here is a typical minimum cost diet linear programming problem. 17

6 A look at the simplex algorithm 18

7 Training Maple to carry out the simplex method 20

8 Two Illustrative Problems 24

9 Big M vs two phase methods for getting a feasible solution. 26

10 Pig problem Revisited 31

11 Exam 1 36

12 Ma 416G Exam 1 37
- 12.1 Problem 1 37
- 12.2 Problem 2 39
- 12.3 39
  - 12.3.1 Solution: By Maple without the simplex package. 39
- 12.4 40
  - 12.4.1 Part B. Sensitivity Analysis. (right hand sides) 40
- 12.5 41
  - 12.5.1 Part C. Sensitivity Analysis. (price coefficients) 41
- 12.6 Problem 3 42
- 12.7 42
  - 12.7.1 Updated tabular 42
- 12.8 43
  - 12.8.1 Automated Pigfarmer 43

12.8.2 45
12.8.3 Tabular 45
12.8.4 45
12.8.5 FindTab 45

13 Sensitivity Analysis 46

14 The revised simplex method 49
14.1 51

15 The revised simplex method (automated) 52

16 More on Big M and 2 phase 55

17 A Possible Correction 58

18 Project 1: Due Friday before spring break 60

19 More uses of matmax 61

20 The farmer's problem: A case problem 64
20.1 c) and d) The six weather scenarios. 66
20.2 g) and h) Some sensitivity analysis. 73

21 Parametric programming example 100

22 parametric programming example 109

23 Transportation problems. 111

24 Ma 416 Exam 2 123

25 Minimum cost flow problems 133

26 Ma 416 Exam 2 138

27 Tiling problems: solved with netsworks 144

28 Tiling with the simplex algorithm 150

29 Soma Cube solutions using the simplex algorithm. 153

30 155
30.1 Some words 155

31 157
31.1 Data 157

32 **160**
32.1 More words 160

Ma 416 - 001 Operations Research – Spring 1997

- **Time and Place:** 1-1:50 pm MWF. CB 337.

- **Teacher:** Carl Eberhart, 756 POT, 257-1258, carl@ms.uky.edu

- **Office Hours:** 11-12, 2-3 MWF, or by appointment. Or use email.

- **Textbook: Introduction to Operations Research,** by F. Hillier and G. Lieberman. 6nd Ed., Holden-Day, 1986.

- Brief Syllabus: Linear programming was born during the second world war, when the problem of minimizing the time of transport of war materials from various ports to various battles was first formulated by George Danzig. The simplex algorithm, which was developed to solve this problem, has since been used to solve a vast number of other problems. One goal for the semester is to learn how to recognize, set up, and solve linear programming problems for solution using the simplex algorithm. We will also learn to use sensitivity analysis, duality theory, the fast transportation simplex algorithm, and parametric programming when appropriate. Discussion and problems relating to these topics is found in chapters 3-8 of Hillier and Lieberman. As time permits, we will also work through material on network analysis (chapter 9) and integer programming (Chapter 12). For all but the simplest problems, we will use Maple to carry out the actual computations. There is also a disk that comes with the text that has useful routinues and tutorials on it. It runs on any dos machine I think.

- Homework: To be successful in this course, you will find it necessary to read and work problems outside of class on a regular basis. Homework assignments will be discussed in class and taken up periodically for marking. (10 % of grade)

- Projects: There will be two Maple projects during the semester. These will be prepared on Maple worksheets which you will email to me. (25% of grade)

- Exams: We will have two midterms (20 % each) and a final (25 %).

- **Grades**: Your homework, project, and exam grades will be converted to a percentage P and an overall grade assigned as follows: $93 \leq P$ A, $90 \leq P < 93$ A-,

  $87 \leq P < 90$ B+, $83 \leq P < 87$ B, $80 \leq P < 83$ B-, $77 \leq P < 80$ C+, $73 \leq P < 77$ C ,

  $70 \leq P < 73$ C-, $67 \leq P < 70$ D+, $63 \leq P < 67$ D, $60 \leq P < 63$ D- , $P < 60$, E.

1 Calendar of class days:

```
matrix(5,3,
[‘Jan‘,‘15W, 17, 22, 24, 27, 29, 31‘,‘Exam 1 Feb 17‘ ,
‘Feb‘,‘3M, 5, 7, 10, 12, 14, 17, 19, 21, 24, 26, 28‘,‘Proj 1 Mar 7‘
,
‘Mar‘,‘3M, 5, 7, 10, 12, 14, 24, 26, 28, 31‘,‘Exam 2 Apr 11‘
,‘Apr‘,‘2W, 4, 7, 9, 11, 14, 16, 18, 21, 23, 25, 28, 30‘,‘Proj 2
Apr 25‘,
‘May‘,‘2F ‘,‘Final May 6‘ ]);
```

$$\begin{bmatrix} \mathit{Jan} & \mathit{15W,\ 17,\ 22,\ 24,\ 27,\ 29,\ 31} & \mathit{Exam\ 1\ Feb\ 17} \\ \mathit{Feb} & \mathit{3M,\ 5,\ 7,\ 10,\ 12,\ 14,\ 17,\ 19,\ 21,\ 24,\ 26,\ 28} & \mathit{Proj\ 1\ Mar\ 7} \\ \mathit{Mar} & \mathit{3M,\ 5,\ 7,\ 10,\ 12,\ 14,\ 24,\ 26,\ 28,\ 31} & \mathit{Exam\ 2\ Apr\ 11} \\ \mathit{Apr} & \mathit{2W,\ 4,\ 7,\ 9,\ 11,\ 14,\ 16,\ 18,\ 21,\ 23,\ 25,\ 28,\ 30} & \mathit{Proj\ \ 2\ \ Apr\ 25} \\ \mathit{May} & \mathit{2F} & \mathit{Final\ May\ 6} \end{bmatrix}$$

2 A Short Introduction to the Maple Language

This section contains an introduction to the Maple vocabulary used for solving problems. It is not meant to cover everything, just some of the basics. In most instances, only the Maple input is described; I am assuming that you are 'typing along', so that you can see the Maple output as we go.

2.1 Arithmetic

First, there is arithmetic: addition, subtraction, multiplication, division and exponentiation. These can be combined, just as on a calculator. The order of precedence is the the usual one: exponentiation first, then multiplication and division, then addition and subtraction. So entering

```
2-3+4/5*6^7;
```

is the same as entering

```
(2-3)+(4/5)*(6^7);
```

Maple works with fractions whenever possible, changing to decimal numbers only on demand. So typing

```
1/3 + 1/2;
```

will get a return of 5/6. Putting a decimal point in one of the numbers forces Maple to return a decimal answer. Also, you can use the maple word evalf to convert a result to decimal form.

```
evalf(1/2+1/3);
```

Maple does arithmetic with complex numbers too. I is a Maple constant standing for sqrt(-1) . So entering

```
(3+2*I)*(2-I);
```

will produce an output of 8+I .

The name for pi, the area of the circle of radius 1, in [Maplese] is Pi . So to calculate the area of a circle of radius 3, you would enter

```
Pi*3^2;
```

2.2 Expressions, Names, Statements, and Assignments

Quantities to be computed like 1/2+1/3 are called expressions . A name is a string of characters wh
Maple is a string of names and expressions terminated with a semicolon, or a colon if you don't want to see the output, which when entered will produce some action. The assignment statement is one of the most common statements. It is of the form

$name := value$; **For example, the assignment**

```
area := Pi*3^2;
```

stores the value 9*Pi in a location marked by the name area.

A more useful assignment for the area of a circle is

```
area := Pi*r^2;
```

In this case, the expression Pi*r^2 is stored in area and with this assignment, the area of a circle of any given radius can be computed using the Maple word subs. So to calculate the area when r is 3, we enter

```
subs(r=3,area);
```

Here, it is convenient to think of the assignment as defining area as a function of the radius r.

2.3 Functions

We are thinking here of the notion of [function] as a rule f (possibly very complicated) for assigning to each argument x in a given set of numbers a unique number f(x) called the value of f at x.

Functions can be defined in several useful ways in Maple.

As an expression: The assignment

```
area := Pi*r^2;
```

defines the area of a circle as a function of it's radius. The area function defined as an expression is evaluated with subs. Since this function assigns real numbers to real numbers, its values can be plotted on a graph with the Maple word plot. So the statement

```
plot(area,r=0..4);
```

will produce in a separate plot window, the graph of the area function over the interval from r=0 ..4 .

With the arrow operator the assignment:

```
area := r -> Pi*r^2;
```

defines the area function also. Now to find the area of a circle of radius 3, we simply enter the statement

```
area(3);
```

To plot this function over the domain r=0..4 , type

```
plot(area,0..4);
```

Note that the variable r is omitted here.

Use unapply . This ugly little word transforms expressions of one or more variables into fuctions defined by an arrow operator. For example, if we had a polynomial defined by the assignment

```
pol := x^2 + 4*x -1;
```

then the assignment

```
pol := unapply(pol,x);
```

turns po l into a function defined by an arrow operator.

As a procedure: the assignment

```
area := proc(r) Pi*r^2 end;
```

defines the area function too. It is evaluated and plotted as in the [arrow operator] definition. One advantage of this way of defining a function is that the domain can be specified. For example, the domain of the area function for a circle is all positive real numbers. This can be inserted into the procedure, with the Maple word ERROR . The message must be enclosed in backquotes ‘ , which is on the key with the tilde .

```
area := proc(r)
if r <= 0 then ERROR(‘radius must be
```

```
positive‘) else
Pi*r^2 fi end;
area(3);
area(-3);
```

Note the if ... then ... fi; control statement here. Use ?if to learn more about this command. Functions of two variables can be defined and plotted just as easily in Maple as functions of one variable.

For example, the volume V of a cylinder of height h and radius r is defined by

```
V := (r,h) -> Pi*r^2*h;
```

To see what the graph of V looks like, use plot3d :

```
plot3d(V,0..4,0..4);
```

Which way of defining a function is the preferred way? That really depends
on the situation. The expression method works well for functions which
have only one rule of evaluation, but eventually you cannot avoid using
an -> or proc definitiion. You will find yourself using arrow or proc
definitions more and more as time goes by.

2.4 Built in Maple functions and Operations with Functions

All of the standard scientific functions are built into Maple. For example, sqrt is the square root function, abs is the absolute value function, the trig and inverse trig functions are sin , arcsin , cos , etc., the natural logarithm and exponential functions are ln and exp . For a complete list of built in functions, type

```
?inifcns;
```

New functions can be obtained from old functions by use of the arithmetic operations of addition, subtraction, multiplication, and division together with the operation of composition, which is denoted by @ . Thus the function defined by the assignment

```
y := sin(cos(x^2+3));
```

and evaluated at x=3 by

```
subs(x=3.,y);
```

could also be defined by the assignment

```
y := sin@cos@(x->x^2+3);
```

and evaluated at x=3 by

```
y(3.);
```

2.5 Using Maple as a fancy graphing calculator.

It is convenient to think of Maple as a fancy graphing calculator for
many purposes. For example, suppose you want to find the real solutions
of the equation $x^5 - 30\,x - 2 = 0$ in the interval $-3..3$. Then we can
just plot the right hand side of the equation.

```
f := x -> 10*x^5 - 30*x +10 ;
plot(f,-3..3);
```

By inspection, the graph crosses near 0. We can look closer.

```
plot(f,-1.5..1.5);
```

We see that the graph crosses 3 times, the largest solution being between 1 and 1.5. If we wanted the largest solution more accurately, we could use fsolve .

```
fsolve(f(x)=0,x,1..1.5);
```

2.6 Data types, Expression Sequences, Lists, Sets, Arrays, Tables:

Maple expressions are classified into various [data types] . For example, arithmetic expressions are classified by whether they are sums [type '+'] , products [type '*'] , etc. The Maple word whattype will tell what type a particular expression is.

```
whattype(1/2);
whattype(a + b);
whattype(x^2 + x = 2*x - 1);
whattype(a,b,3);
```

Expression Sequence.

An exprseq , [expression sequence] , is any sequence of expressions separated by commas. For example,

```
expseq := 1,2, w*r+m, a=b+c, 1/2,
(x+y)/z,`hello`;
```

is an assignment to expseq of an expression sequence of 7 expressions. To refer to the sixth expression in this sequence, use the expression exseq[6];

```
expseq[6];
```

List. A [list] is an expression sequence enclosed by square brackets. So

```
explist:= [expseq];
```

makes a list whose terms are those in expseq . As with expression sequences, we can refer to particular terms of a list by appending to its name the number of the term enclosed in square brackets. Thus to get the fifth term of explist , type the expression

```
explist[3];
```

You can also reference the fifth term in this list by typing

```
op(3,explist);
```

In general, op(n,explist); returns the nth term in the list explist .

To count how many terms are in a list, use the word nops . So for example,

```
nops(explist);
```

tells us that there are 7 terms in the list explist . nops comes in handy when you don't want to (or can't) count the terms in a list by hand (this is almost always).

You can't directly use the word nops to count the number of terms in an expression sequence. But you can put square brackets around the expression sequence and count the terms in the resulting list. This device is used again and again.

```
nops(3,4,a);
nops([3,4,a]);
```

One important use of lists is to make lists of points to plot. For example, to draw a picture of the square with vertices (1,1), (3,1), (3,3), (1,3), make a list and then plot it.

```
ab := [[1,1],[3,1],[3,3],[1,3],[1,1]];
```

```
plot(ab);
```

Notice in the graph that the origin is not included in the field of view.

We can specify that by restricting the x and y coordinates. We can also choose to have no axes displayed.

```
plot(ab,x=0..4,y=0..4);
```

Another use of lists is with [parametric plots] . If you have a curve in the

plane described parametrically with $x = \mathrm{f}(t)$, $y = \mathrm{g}(t)$, as the parameter t runs from a to b, then you can draw it by making up a 3 term list to give to plot. Say you wanted to draw the upper half of the circle of radius 4 centered at (1,5). Then the list consists of the expressions for the x and y coordinates followed by an equation giving the range of the parameter.

```
plot(
[1+4*cos(t),5+4*sin(t),t=0..Pi],scaling=constrained);
```

If you had to draw several pieces of circles, you might define a function circ to simplify things.

```
circ := (h,k,r,f,l) ->
[h+r*cos(t),k+r*sin(t),t=f..l];
```

So if we wanted circles of radius 1/2 centered at the corners of the square ab we can construct the sequence of lists

```
circs := seq(circ(op(ab[i]),
1/2,0,2*Pi),i=1..4);
```

In order to plot these circles, you need to enclose them in curly brackets to make a set of the sequence before you give them to plot . See below for a discussion of sets.

```
plot({circs,ab},scaling=constrained);
```

Sometime you might want to split a list of points to plot into a list of x-coordinates and another list of ycoordinates. The Maple word seq is very handy for thisand many other operations. So to split off from ab the odd and even terms–

```
xdat := [ seq(ab[i][1],i=1..nops(ab) )];
ydat := [seq(ab[i][2],i=1..nops(ab) )];
```

What about the converse problem? Building up a list of points to plot from two lists can also be done. The first thing you might think of doesn't work, however.

```
seq([xdat[i],ydat[i]],i=1..nops(xdat));
```

Seq doesn't work well with a pure expression sequence as input. However, with some coaxing we can get it to do what we want.

```
newab
:=[seq([xdat[i],ydat[i]],i=1..nops(xdat))];
```

What did we do to change the input to seq ? We enclosed it in square brackets. If you feed such a list of points to plot, it knows what to do. If you wanted to strip out the inside brackets, that can be done too, but in release

4 of Maple, plot would treat it as a sequence of constant functions.

```
newab :=
[seq(op([xdat[i],ydat[i]]),i=1..nops(xdat))];
plot(newab,color=black);
```

Sets.

A [set] is an expression sequence enclosed by curly brackets. Order of appearance is not important in sets. Sets are important when plotting more than one function at at time. For example,

```
plot({x^2-2,2*x+5},x=-5..5);
```

plots the parabola $y = x^2 - 2$ and the line $y = 2\,x + 5$ over the domain $x = -5..5$ on the same graph.

If you have a very complicated drawing to make, you can use plots[display] fromthe plots package.

Tables and Arrays

A [table] is a special kind of data structure which is very flexible. The packages of special vocabularies are really tables whose indices of the package are the names of the procedures and whose entries are the bodies of the procedures.

An [array] is a special kind of table whose indices are numerical. The most useful arrays are matrices (2 dimensional arrays) and vectors (1 dimesional arrays). Matrix operations are made using Maple word evalm together with the symbol for [matrix multiplication] &* .

```
a := array(1..2,1..2);
```

creates a 2 by 2 matrix, whose entries are accessed as a[1,1] etc.

So to rotate the square $ab := [[1,\,1],\,[3,\,1],\,[3,\,3],\,[1,\,3],\,[1,\,1]]$ through an angle of 31 degrees counter clockwise about the origin and display it, we

could proceed as follows.

```
rot := array([[cos,-sin],[sin,cos]]);
ang := evalf(Pi/180*31);
ab := [[1,1],[3,1],[3,3],[1,3],[1,1]];
rotab := [seq(convert(
evalm(rot(ang)&*ab[i]),list) ,i=1..nops(ab) )];
plot({ [[0,0]],ab,rotab} );
```

2.7 Maple control statements

There are two especially important control statements . One is the repetition loop, and the other is the conditional execution statement. The repetition loop is

for .. from .. by .. to .. while .. do .. od;

This statement can be used interactively or in a procedure to perform repetitive tasks or to do an iterative algorithm.

Example: Add up the first 100 numbers.

```
s := 0:  for i from 1 to 100 do s := s+i
od:
s;
```

Example: Compute the cubes of the first five positive integers and store them in a list. Then do it again, storing them in an array.

Solution with lists:

```
locube := []:  # start with the empty list
for i from 1 to 5 do
locube := [op(locube),i^3] od:
locube ; # look at the list;
```

Note the way the list is built up from nothing. Each time through the loop, one more term is added onto the end of the list. With arrays, one can be more direct.

Solution with arrays:

```
aocube := array(1..5):  # initialize the
array.
for i from 1 to 5 do aocube[i]:= i^3 od;
op(aocube); # to see the array
```

Now the array aocube has the numbers stored in it. To refer to the third element of aocube , we would enter aocube[3] just as if it were a list, rather than an array. Why have arrays at all? Well, for one thing, the terms in an array can be more easily modified. For example, to change the third term in aocube to 0 just enter aocube[3] := 0; . To change the third term in locube to 0, you have to make an entirely new list whose terms are all the same as locube except for the third one.

```
aocube[3]:=0;
print(aocube);
locube :=
[locube[1],locube[2],0,locube[4],locube[5]];
```

Problem.

There is a much less cumbersome way of doing the list solution.

Start with the empty expression sequence locube := NULL:

and just tack the cubes onto the end of the sequence as they are generated.

for i from 1 to 5 do locube := locube,i^3 od: Now you have the first 5 cubes in a sequence. How do we turn that into a list? By putting square brackets around it! locube := [locube]; Test this out.

Conditional execution if .. then .. elif .. else .. fi;

There are lots of times when you need to consider cases, and they can all be handled with the [if .. then .. elif .. else .. fi;] statement. For example, many functions are defined piecewise. The absolute value function abs is such a function.

Problem: Define your own version of the absolute value function.

A solution:

```
myabs := proc(x) if x > 0 then x else -x fi
end;
myabs(-23);
plot(myabs,-2..2,scaling=constrained,title=`m
y absolute value`); # to see what it looks like.
```

2.8 A Brief Vocabulary of Maple Words

We have gathered here are some Maple words useful in problem solving, together with examples of their usage. For more information on these words and others, look at the helpsheets and use the help browser.

```
y := (x+3)/tan(x^2-1); # use 'colon-equal'
to make assignments.
collect(x*2 + 4*x,x); # collects like powers
of x.
diff(cos(x),x); # calculates the derivative
D(cos); # the differential operator
y := denom((a+b)/(e+f)); # assigns e+f to
y.
y := 'y'; # makes y a variable again.
evalc((2+3*I)^3); # performs complex
```

```
arithmetic
evalf(1/2^9); #evaluates 1/2^9 to a decimal
number
expand((x+b)^7); # expands the product
p := x^2+5*x+6; # assigns the quadratic to
p.
factor(p); # factors the polynomial
fsolve(x^5-3*x=1,x,0..2); # solve eqn for x
in 0..2
int(x*exp(x),x); # returns an
antiderivative.
Int(x*exp(x),x=0..1); # A passive
integral.
map(x->x^2,[1,3,2,5]); # returns a list of
squares.
nops([3,4,x,1]); # returns the number of
terms in the list.
numer((a+b)/c); # gives numerator, here a+b
op([3,4,1,x]); # strips the brackets off the
list
plot(x^2+x, x=-3..3); # plots x^2+x as x
goes from -3 to 3.
plot3d(x^2+y,x=-2..2,y=0..2); # plots a
surface
f := x -> x^2; # defines the squaring
function.
f(3); # then returns 9.
quo((x^4-4),(x^2-2),x); # divides
polynomials
iquo(23,2) ; # divides the integers
rem((x^4-4*x+3),(x^2-2),x); # gives the
remainder
irem(23,2) ; # gives the integer remainder
restart; # very handy.  This word resets all
assignments.
eq1 := x^2 + 3*x -1 = a; # assigns the
equation
rhs(eq1); # yields the righthand side of
eq1.  There is also an lhs.
simplify(a/x+b/y); # sometimes simplifies
expr.
solve(a*x+4*y=0,x); # solve the equation
for x.
subs(x=5,x^2+x); # substitute 5 for x where
it occurs in x^2+x.
i := 'i'; # makes i a variable again
sum((i^2,i=2..9)); # add up the 2nd thru 9th
```

`squares`

2.9 Trouble Shooting Notes

Learning to use Maple can be an extremely frustrating experience, if you let it. There are some types of errors which occur from the beginning that can be spotted and corrected easily by a person fluent in Maple, so if you have access to such a person, use them. Here are a few suggestions that may be of use when you're stuck with a worksheet that's not working.

- Use the help facility: There is a help sheet with examples for every Maple word. A quick read thru will often clear up syntax problems. One very common early mistake is to leave out the parentheses around the inputs of a word. For example, typing `plot x^2` will get you a syntax error, because you left out the parentheses.

- The maple prompt is > . You can begin entering input after it.

Make sure you are typing into an input cell, if you are expecting output.

- End maple statements with a semicolon ; . Maple does nothing until it finds a semicolon. If you are getting no output when you should be, tryfeeding in a semicolon. This often works. For example,

- When in doubt, put in parentheses. For example, (x+3)/(x

3 A look at the graphical method for problems with two decision variables.

Here are two words to show the feasible region in the two dimensional case.
First load the plots and simplex package.

```
with(plots):  with(simplex):
```

Now here is a first attempt at showing the feasible region.

```
showfeas1 := proc(obj,cnsts,mx,my)
local n,i,pl1,pl2,m,cx,cy,yobj,xobj,eq ;
n:=nops(cnsts);
m := NULL;
for i from 1 to n do
eq := lhs(cnsts[i])=rhs(cnsts[i]);
if not member(y,indets(eq)) then
cx := solve(eq,x); m := m,[[ cx,0],[cx,my]]
else cy := solve(eq,y);
m := m,[[0,subs(x=0,cy)],[mx,subs(x=mx,cy)]]
fi od;
pl1 := plot({m},view=[0..mx,0..my],color=black);
yobj := subs({x=0,y=my},obj) ;
xobj := solve(subs(y=0,obj=yobj),x) ;
pl2 := plot( [[0,my],[xobj, 0]],color=red,
thickness =2);
plots[display]([pl1,pl2],view=[0..mx,0..my],axes=boxed); end;
showfeas1(x+2*y,[ 3*x+4*y<=30,4 *x+3*y <=
25,2*x+5*y <= 30],10,10);
```

satisfies is used in an improved version of showfeas below.

```
satisfies := proc(pt,cnsts)
local i;
for i from 1 to nops(cnsts) do
if not subs(pt,cnsts[i]) then RETURN(false) fi od; true end:
```

showfeas takes an objective function of x and y, a set of linear constraints (of the form $a\,x + b\,y \leq c$), and a maximum x mx and maximum y my. It returns a plot of the feasible region lying in the rectangle with opposite corners [0,0] and [mx,my]. It also draws a level curve of the objective function through the region. This allows you to 'eyeball' the corner point where the objective function is maximized.

```
showfeas := proc(obj,cnsts,mx,my)
local n, i, pl1, pl2, pl3, j, pts, ppt, pt, m, eqns, br, xr, yr,
cx, cy, yobj, xobj;
Digits := 4:
n:=nops(cnsts);
eqns := [seq(lhs(cnsts[i])=rhs(cnsts[i]),i=1..n),x=0,y=0];
m := NULL;
for i from 1 to n do
if not member(y,indets(eqns[i])) then
cx := solve(eqns[i],x); m := m,[[ cx,0],[cx,my]]
else cy := solve(eqns[i],y);
```

```
m := m,[[0,subs(x=0,cy)],[mx,subs(x=mx,cy)]]
fi od;
pl1 := plot({m},view=[0..mx,0..my],color=black);
yobj := subs({x=0,y=my},obj) ;
xobj := solve(subs(y=0,obj=yobj),x) ;
pl2 := plot( [[0,my],[xobj, 0]],color=red,
thickness =2);
pts := NULL:
for i from 1 to n+1 do for j from i+1 to n+2 do
pt := solve({eqns[i],eqns[j]},{x,y});
ppt := subs(evalf(pt),[x,y]):
if type(evalf(ppt[1]),numeric) and type(evalf(ppt[2]),numeric)
and ppt[1]>=0 and ppt[2]>=0 and satisfies({x=ppt[1],y=ppt[2]},cnsts)
then
pts := pts,ppt fi;
od od;
print([pts]);
pts := simplex[convexhull]([pts]);
pl3:= plots[polygonplot](pts,style=patch,color=blue);
plots[display]([pl1,pl2, pl3]) ;
end;
```

Here's a sample input for showfeas.

```
showfeas (x+2*y,[ 3*x+2*y<=20,4 *x+3*y <=
25,2*x+5*y <= 30],
15,10 ) ;
```

4 The Wyndor Glass Problem

Two packages of words that you will want to load in when working on linear programming problems are linalg and simplex.

```
with(linalg);
with(simplex);
```

Here's the statement of the problem.

Three plants are used to produce two products. Each unit of production of each product provides a unit profit and uses a certain amount of capacity at each plant. Each plant has only a certain amount of capacity available. The table below shows the relevant data.

```
matrix(5,4,[‘plant‘,‘prod 1‘, ‘prod 2‘,
‘cap.  avail‘,
1,1,0,4,
2,0,2,12,
3,3,2,18,
‘unit profit‘,3,5,‘ ‘]);
```

The basic problem is to design a production schedule that maximizes profits.

Setup: Let x_1 and x_2 be the number of units (fractions allowed) of each product produced in the schedule.

```
profit := 3*x[1] + 5*x[2];
```

$$\mathit{profit} := 3\,x_1 + 5\,x_2$$

```
constraints :=
[x[1]<=4,2*x[2]<=12,3*x[1]+2*x[2]<=18];
```

$$\mathit{constraints} := [x_1 \leq 4,\ 2\,x_2 \leq 12,\ 3\,x_1 + 2\,x_2 \leq 18]$$

Solution:

To just see the solution, use the **maximize** command from the simplex package.

```
sol :=
maximize(profit,constraints,NONNEGATIVE);
```

$$\mathit{sol} := \{x_1 = 2,\ x_2 = 6\}$$

Then substitute those values into the profit function and the constraints.

```
subs(sol,[profit,constraints]);
```

$$[36,\ [2 \leq 4,\ 12 \leq 12,\ 18 \leq 18]]$$

So according to the simplex package, the maximum profit of 18 is attained by producing 2 units of product 1 and 6 units of product 2. This schedule uses all the capacity at plants 2 and 3, but only 50 percent of the capacity of plant 1.

To see the steps of the simplex algorithm, set the infolevel for the simplex package to 2 and repeat the **maximize** command.

```
infolevel[simplex]:= 2;
maximize(profit,constraints,NONNEGATIVE);
```

To step through the simplex algorithm, use the commands **setup, pivotvar, pivoteqn, and pivot** from the simplex package, instead of the command maximize.

First turn the constraints into equations using setup. Setup adds a nonnegative 'slack variable' sli to the left hand side of the ith inequality to take up the slack when the variables x[1] and x[2] are set to zero. Thus the initial feasible solution is $x_1 = \mathbf{0}$, $x_2 = \mathbf{0}$ for the nonbasic variables and $\mathit{sl1} = 4$, $\mathit{sl2} = \mathbf{12}$, and $\mathit{sl3} = \mathbf{18}$ for the basic variables.

```
eqns := setup(constraints);
```

$$eqns := [\_SL1 = 4 - x_1,\ \_SL2 = 12 - 2\,x_2,\ \_SL3 = 18 - 3\,x_1 - 2\,x_2]$$

Then get a pivot (or entering) variable using pivotvar. Note: pivotvar simply returns the first (nonbasic) variable with a positive coefficient, not the variable with largest positive coefficient.
If there are no positive coefficients, then it returns FAIL.

```
v := pivotvar(profit);
```

$$v := x_2$$

Now get the pivot equation for that variable. Pivoteqn chooses the equation in which the first basic variable to decrease to zero as the value of the entering variable increases from zero occurs. It does this by calculating the ratio (constant term)/(coefficient of v) for each equation in the list of equations and choosing the negative ratio closest to zero.

```
peq := pivoteqn(eqns,v);
```

$$peq := [\_SL2 = 12 - 2\,x_2]$$

Then make the pivot exchange. pivot solves the pivot equation for the pivot variable and substitutes that into the other equations.

```
eqns:= pivot(eqns,v,peq) ;
```

$$eqns := [\_SL1 = 4 - x_1,\ x_2 = -\frac{1}{2}\,\_SL2 + 6,\ \_SL3 = 6 - 3\,x_1 + \_SL2]$$

At this stage the current solution is $\mathit{sl2} = \mathbf{0},\ x_1 = \mathbf{0}$ for the nonbasic variables and $x_2 = \mathbf{6},\ \mathit{sl1} = 4,\ \mathit{sl3} = \mathbf{6}$ for the basic variables.
Now update the profit function by substituting the new basic variable (x_2) into the profit function , thus expressing it as function of the new set of nonbasic variables ($x_1,\ \_Sl2$).

```
profit := subs(eqns[2],profit);
```

$$profit := 3\,x_1 - \frac{5}{2}\,\_SL2 + 30$$

We can read off the profit of 30 for the current solution from this. However, we can still increase this profit by bringing x_1 into solution (provided it has a negative coefficient in one of the equations). We can look up above and see that it has a negative coefficient in 1st and 3rd equations. Further, the ratios are 4/-1 = -4 for the 1st and 6/(-3) = -2 for the 3rd. This means pivoteqn should return the 3rd equation. Checking this out,

```
v :=pivotvar(profit);
```

$$v := x_1$$

```
peq := pivoteqn(eqns,v);
```

$$peq := [\_SL3 = 6 - 3\,x_1 + \_SL2]$$

Indeed, that is what happened. So make the pivot and recalculate the profit.

```
eqns:= pivot(eqns,v,peq) ;
```

$$eqns := [\_SL1 = 2 + \frac{1}{3}\,\_SL3 - \frac{1}{3}\,\_SL2,\ x_2 = -\frac{1}{2}\,\_SL2 + 6,\ x_1 = -\frac{1}{3}\,\_SL3 + 2 + \frac{1}{3}\,\_SL2]$$

```
profit := subs(eqns[3],profit);
```

$$profit := -\_SL3 + 36 - \frac{3}{2}\,\_SL2$$

Now the current solution is $x_1 = \mathbf{2},\ \mathit{sl1} = \mathbf{2},\ x_2 = \mathbf{6}$ with a profit of 36. We can see this is maximum since the variables in the profit expression both have negative coefficients and so increasing either would decrease the profit.

5 Here is a typical minimum cost diet linear programming problem.

Pig problem

```
restart;with(linalg);
```

Problem: A pig farmer has resources of corn, tankage, and hay to make up a pig diet which satisfies certain minimum daily requirements as to the basic nutrients of carbohydrates, proteins, and vitamins. The table showing the minimum amounts of each nutrient needed and the amount of each nutrient in a kilogram of resource and the cost of the resource is given below. Desired is a diet which satisfies the minimum daily requirement at minimum cost.

```
tab := matrix(5,5,[
‘Nutrients‘,‘corn(Kg)‘,‘tankage(Kg)‘,‘hay(Kg)‘,‘mdr‘,
‘carbo‘,90,20,40,200,
‘protein‘,30,80,60,180,
‘vitamins‘,10,20,60,150,
‘cost(.01$)‘,35,30,25,‘ ‘]);
```

Setup: Let c, t, and h denote the number of kg of corn, tankage, and hay per pig in the diet. Then the constraints are linear inequalities in these quanitities.

```
cnsts := {90*c + 20*t + 40*h >=200,
30*c + 80*t + 60*h >=180,
10*c + 20*t + 60*h >=150};
```

The cost is a linear function of these quantitities.

```
cost := 35*c + 30*t + 25*h;
```

We can use the simplex package to solve this problem.

```
with(simplex);
infolevel[simplex]:=1;
Digits := 4:
minimize(cost,cnsts,NONNEGATIVE);
sol:=
minimize(evalf(cost),evalf(cnsts),NONNEGATIVE) ;
evalf(sol),subs(evalf(sol),[cost,cnsts]) ;
```

So, each pig should get no tankage, 2.43 kg of hay, and 1.14 kg of corn each day at at cost of about 1.01 dollars. The pig gets his minimum carbo and protein, and 7 units over his minimum vitamins.

Questions:

1. How low would the price of tankage have to get before it would be used in the pig diet?
2. How low would the amount of vitamins in hay have to get before it becomes economical to use tankage in the diet?
3. Suppose there is only 2 kg of hay per pig available. What is the minimum cost pig diet then?
4. Suppose there is an additional resource of sardine meal costing 33 cents per kg and containing 60, 40, and 15 units of carbo, protein, and vitamins respectively. What is the minimum cost diet then?

6 A look at the simplex algorithm

```
obj := z = 3*x + 4*y;
cnstrts := [5*x + 2*y <= 20, 4*x + 7*y <= 40];
```

$$obj := z = 3\,x + 4\,y$$

$$cnstrts := [5\,x + 2\,y \leq 20,\ 4\,x + 7\,y \leq 40]$$

First introduce **slack variables** to turn the constraints into equations.

```
eqns := [s1 = 20-5*x - 2*y ,s2 = 40- 4*x -
7*y ];
```

$$eqns := [s1 = 20 - 5\,x - 2\,y,\ s2 = 40 - 4\,x - 7\,y]$$

We have a **current feasible solution** of [x,y,s1,s2]= [0,0,20,40] with z = 0. The **basic** (or dependent) variables are s1 and s2. The **nonbasic** (or independent) variables are x and y. The current feasible solution is obtained by assigning the value 0 to the nonbasic variables. Can we improve the value of z? Yes. Looking at the current obj, we see that increasing x or y will result in an increase in z, since the coefficients of x and y are positive. We might as well choose to increase y since that results in the quickest increase in z. (note: the simplex package simply chooses the first variable in the expression for z that has a positive coefficient.) So the pivot variable is y;

```
pivotvar := y;
```

$$pivotvar := y$$

Now which basic variable becomes zero first as y increases? Looking at the first equation in the list of equations, we see that as y increases one unit, s1 decreases by 2 units, so y can increase to 20/2 = 10 units before s1 is 0. In the same way, from the second equation, we see that y can increase to 40/7 before s2 becomes 0. So s2 becomes 0 before s1 does as y increases. This means we want to interchange y and s2 in our equations. So the pivot equation is the one have s2 as the left hand side.

```
pivoteqn := eqns[2];
```

$$pivoteqn := s2 = 40 - 4\,x - 7\,y$$

```
neweqn := pivotvar =
solve(pivoteqn,pivotvar);
```

$$neweqn := y = -\frac{1}{7}\,s2 + \frac{40}{7} - \frac{4}{7}\,x$$

```
eqns := [subs(neweqn,eqns[1]),neweqn];
obj := subs(neweqn,obj);
```

$$eqns := [s1 = \frac{60}{7} - \frac{27}{7}\,x + \frac{2}{7}\,s2,\ y = -\frac{1}{7}\,s2 + \frac{40}{7} - \frac{4}{7}\,x]$$

$$obj := z = \frac{5}{7}\,x - \frac{4}{7}\,s2 + \frac{160}{7}$$

One iteration of the simplex algorithm has been completed. Now the current feasible solution is x=0, s2 = 0, s1 = 60/7, y = 40/7. z has improved to 160/7 = 22 and 6/7.

Can we further improve the value of z? Yes. By increasing x from zero. Is that permissible? Check what happens to the basic variable values as x increases. (They can't become negative.) Looking at eqns[1], increasing x by a unit decreases s1 by 27/7 so x can increase to (60/7) / (27/7) = 60/27 before s1 gets down to 0. And from eqsn[2] we see that x can be increased to (40/7)/(4/7)=10 before y decreases to 0. So we want to interchange x and s1.

```
pivotvar := x;
pivoteqn := eqns[1];
```

$$pivotvar := x$$

$$pivoteqn := s1 = \frac{60}{7} - \frac{27}{7}\,x + \frac{2}{7}\,s2$$

```
neweqn := pivotvar =
solve(pivoteqn,pivotvar);
eqns := [neweqn,subs(neweqn,eqns[2])];
obj := subs(neweqn,obj);
```

$$neweqn := x = -\frac{7}{27}\,s1 + \frac{20}{9} + \frac{2}{27}\,s2$$

$$eqns := [x = -\frac{7}{27}\,s1 + \frac{20}{9} + \frac{2}{27}\,s2,\ y = -\frac{5}{27}\,s2 + \frac{40}{9} + \frac{4}{27}\,s1]$$

$$obj := z = -\frac{5}{27}\,s1 + \frac{220}{9} - \frac{14}{27}\,s2$$

This completes the second iteration of the simplex method. Now the basic variables are x and y, and the nonbasic variables are s1 and s2. When the nonbasic variables are set to 0 we read the current feasible solution as [x,y,s1,s2]=[20/9,40/9,0,0] and the objective has been improved to 220/9=24 and 4/9.

Can we further improve the value of z? No. Since increasing either of s1 or s2 will actually decrease the value of z (look at the equation for z), we are at a maximum.

7 Training Maple to carry out the simplex method

We have seen how to use Maple to help up carry out the simplex method, first using the simplex package and then without using the simplex package. In this worksheet we develop a word (**mymax**) which steps through the simplex algorithm, printing out the standard tableau after each iteration. We use **setup**, **pivotvar** and **pivot** from the simplex package.

```
restart;with(linalg):with(simplex):
```

```
Warning, new definition for norm

Warning, new definition for trace

Warning, new definition for basis

Warning, new definition for maximize

Warning, new definition for minimize

Warning, new definition for pivot
```

First, we need a word to display the standard tableau. **tabular** takes a list of equations, an objective function and (maybe) a list of variables and returns the standard tableau.

```
tabular := proc (f ,obj, v )
local t, A, m,i, n, x,ob,c,Z, vars,zrow;
t := convert(map(lhs-rhs = 0,f),list);
if nargs = 2 then vars := sort(convert(indets(t,name),list))
else vars := v fi;
ob:=subs([seq(h=0,h=vars)],obj);
zrow := linalg[genmatrix]([obj=0],vars,'flag');
zrow[1,nops(vars)+1] := ob;
A := linalg['genmatrix'](t,vars,'flag');
m := linalg['rowdim'](A); n := linalg['coldim'](A);
A := stack(zrow,A);
A := stack([op(vars),Z],A);
end:
```

Now let's test tabular out to see if it does what we want.

```
cnsts := [x1+2*x2<=4,x1+x2<=3];
```

$$cnsts := [x1 + 2\,x2 \le 4,\ x1 + x2 \le 3]$$

```
eqns := setup(cnsts);
```

$$eqns := [\_SL1 = 4 - x1 - 2\,x2,\ \_SL2 = 3 - x1 - x2]$$

```
z := 2*x1+3*x2;
```

$$z := 2\,x1 + 3\,x2$$

```
tabular(eqns,z);
```

$$\begin{bmatrix} \_SL1 & \_SL2 & x1 & x2 & Z \\ 0 & 0 & 2 & 3 & 0 \\ 1 & 0 & 1 & 2 & 4 \\ 0 & 1 & 1 & 1 & 3 \end{bmatrix}$$

step takes a list of equations, an objective, and (maybe) a precision, and returns the result after a pivot is attempted. If the attempt fails, an error message explaining why is returned. If the attempt suceeds, then v is the new basic variable, vout is the basic variable it replaces, tmp is eqns solved for the basic variables and tz is the objective function expressed in terms of the nonbasic variables.

```
step := proc(eqns,Z,n)
local v,vout, peq,i,tmp,tZ;
v :=pivotvar(Z);
if v = FAIL then RETURN(‘finished:  no pivot variable.‘) fi;
peq := pivoteqn(eqns,v);
if peq = FAIL then RETURN(‘finished:  no pivot equation.‘) fi;
vout := lhs(op(peq));
tmp := pivot(eqns,v,peq) ;
for i from 1 to nops(tmp) do if v = lhs(tmp[i])
then break fi; od;
tZ := subs(tmp[i],Z);
if nargs = 3 then print(evalf(tabular(tmp,tZ),n));
else print(tabular(tmp,tZ)) fi;
[tmp,tZ,[v,vout]];
end:
```

To test out step, we will use eqns and z that we have already set up.

```
sol := step(eqns,z);
```

$$\begin{bmatrix} \mathit{\_SL1} & \mathit{\_SL2} & \mathit{x1} & \mathit{x2} & \mathit{Z} \\ 0 & -2 & 0 & 1 & 6 \\ 1 & -1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 3 \end{bmatrix}$$

$$\mathit{sol} := [[\mathit{\_SL1} = 1 + \mathit{\_SL2} - \mathit{x2},\ \mathit{x1} = -\mathit{\_SL2} + 3 - \mathit{x2}],\ -2\,\mathit{\_SL2} + 6 + \mathit{x2},\ [\mathit{x1},\ \mathit{\_SL2}]]$$

To iterate, we defined step so that we can feed the first two parts of the output from step back into step.

```
sol := step(sol[1],sol[2]);
```

$$\begin{bmatrix} \mathit{\_SL1} & \mathit{\_SL2} & \mathit{x1} & \mathit{x2} & \mathit{Z} \\ -1 & -1 & 0 & 0 & 7 \\ 1 & -1 & 0 & 1 & 1 \\ -1 & 2 & 1 & 0 & 2 \end{bmatrix}$$

$$\mathit{sol} := [[\mathit{x2} = -\mathit{\_SL1} + 1 + \mathit{\_SL2},\ \mathit{x1} = -2\,\mathit{\_SL2} + 2 + \mathit{\_SL1}],\ -\mathit{\_SL2} + 7 - \mathit{\_SL1},\ [\mathit{x2},\ \mathit{\_SL1}]]$$

```
sol := step(sol[1],sol[2]);
```

$$\mathit{sol} := \mathit{finished} :\ \mathit{no\ pivot\ variable.}$$

So this seems to work. Now lets define **mymax**.

mymax takes the original eqns, and obj z and prints the sequence of tableau leading to an optimal solution. It returns the final set of equations and the sequence of pivots used to get there. This is for use later when we do sensitivity analysis.

```
mymax := proc(eqns,z)
local sol, oldsol,vseq;
sol := step(eqns,z);
oldsol := sol;
vseq := NULL;
while nops(sol)=3 do
```

```
oldsol := sol;
sol := step(sol[1],sol[2]);
if nops(oldsol) = 3 then vseq :=
vseq,oldsol[3] fi;
#print(sol);
od;
print(sol);
[oldsol[1],oldsol[2],[vseq]];
end;
```

$$\begin{array}{l}
\mathit{mymax} := \mathrm{proc}(\mathit{eqns},\, z) \\
\quad \mathrm{local}\,\mathit{sol},\, \mathit{oldsol},\, \mathit{vseq}; \\
\qquad \mathit{sol} := \mathrm{step}(\mathit{eqns},\, z)\,; \\
\qquad \mathit{oldsol} := \mathit{sol}\,; \\
\qquad \mathit{vseq} := \mathit{NULL}\,; \\
\qquad \mathrm{while}\;\mathrm{nops}(\mathit{sol}) = 3\;\mathrm{do} \\
\qquad\quad \mathit{oldsol} := \mathit{sol}\,;\; \mathit{sol} := \mathrm{step}(\mathit{sol}_1,\, \mathit{sol}_2)\,;\; \mathrm{if}\;\mathrm{nops}(\mathit{oldsol}) = 3\;\mathrm{then}\;\mathit{vseq} := \mathit{vseq},\, \mathit{oldsol}_3\;\mathrm{fi} \\
\qquad \mathrm{od}; \\
\qquad \mathrm{print}(\mathit{sol})\,; \\
\qquad [\mathit{oldsol}_1,\, \mathit{oldsol}_2,\, [\mathit{vseq}]] \\
\quad \mathrm{end}
\end{array}$$

Now check out mymax with the same eqns and z that we were using before.

```
mymax(eqns,z);
```

$$\left[\begin{array}{ccccc}
\_SL1 & \_SL2 & x1 & x2 & Z \\
0 & -2 & 0 & 1 & 6 \\
1 & -1 & 0 & 1 & 1 \\
0 & 1 & 1 & 1 & 3
\end{array}\right]$$

$$\left[\begin{array}{ccccc}
\_SL1 & \_SL2 & x1 & x2 & Z \\
-1 & -1 & 0 & 0 & 7 \\
1 & -1 & 0 & 1 & 1 \\
-1 & 2 & 1 & 0 & 2
\end{array}\right]$$

$$\mathit{finished}\;:\;\;\mathit{no\ pivot\ variable.}$$

$$[[x2 = -\_SL1 + 1 + \_SL2,\; x1 = -2\,\_SL2 + 2 + \_SL1],\; -\_SL2 + 7 - \_SL1,\; [[x1,\, \_SL2],\, [x2,\, \_SL1]]]$$

Check it on a 4 variable, 4 constraint problem.

```
consts := [x1+23*x2+14*x3+7*x4 <= 200,
13* x1+12*x2+ 7*x3+17*x4 <= 200 ,
6*x1+ 3*x2+ 4*x3+2*x4 <= 200,
2*x1+15*x2+2*x3+11*x4 <= 200];
```

$$\mathit{consts} := [x1 + 23\,x2 + 14\,x3 + 7\,x4 \le 200,\; 13\,x1 + 12\,x2 + 7\,x3 + 17\,x4 \le 200,\; 6\,x1 + 3\,x2 + 4\,x3 + 2\,x4 \le 200,\; 2\,x1 + 15\,x2 + 2\,x3 + 11\,x4 \le 200]$$

```
z := 2*x1+5*x2+4*x3+7*x4;
```

$$z := 2\,x1 + 5\,x2 + 4\,x3 + 7\,x4$$

```
eqns := setup(consts);
```

$$\begin{array}{l}
\mathit{eqns} := [\_SL1 = 200 - x1 - 23\,x2 - 14\,x3 - 7\,x4, \\
\quad \_SL2 = 200 - 13\,x1 - 12\,x2 - 7\,x3 - 17\,x4,\; \_SL3 = 200 - 6\,x1 - 3\,x2 - 4\,x3 - 2\,x4, \\
\quad \_SL4 = 200 - 2\,x1 - 15\,x2 - 2\,x3 - 11\,x4]
\end{array}$$

```
sol :=mymax(eqns,z);
```

$$\left[\begin{array}{ccccccccc}
\mathit{\_SL1} & \mathit{\_SL2} & \mathit{\_SL3} & \mathit{\_SL4} & \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{x4} & \mathit{Z} \\
0 & \frac{-2}{13} & 0 & 0 & 0 & \frac{41}{13} & \frac{38}{13} & \frac{57}{13} & \frac{400}{13} \\
1 & \frac{-1}{13} & 0 & 0 & 0 & \frac{287}{13} & \frac{175}{13} & \frac{74}{13} & \frac{2400}{13} \\
0 & \frac{1}{13} & 0 & 0 & 1 & \frac{12}{13} & \frac{7}{13} & \frac{17}{13} & \frac{200}{13} \\
0 & \frac{-6}{13} & 1 & 0 & 0 & \frac{-33}{13} & \frac{10}{13} & \frac{-76}{13} & \frac{1400}{13} \\
0 & \frac{-2}{13} & 0 & 1 & 0 & \frac{171}{13} & \frac{12}{13} & \frac{109}{13} & \frac{2200}{13}
\end{array}\right]$$

$$\left[\begin{array}{ccccccccc}
\mathit{\_SL1} & \mathit{\_SL2} & \mathit{\_SL3} & \mathit{\_SL4} & \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{x4} & \mathit{Z} \\
\frac{-1}{7} & \frac{-1}{7} & 0 & 0 & 0 & 0 & 1 & \frac{25}{7} & \frac{400}{7} \\
\frac{13}{287} & \frac{-1}{287} & 0 & 0 & 0 & 1 & \frac{25}{41} & \frac{74}{287} & \frac{2400}{287} \\
\frac{-12}{287} & \frac{23}{287} & 0 & 0 & 1 & 0 & \frac{-1}{41} & \frac{307}{287} & \frac{2200}{287} \\
\frac{33}{287} & \frac{-135}{287} & 1 & 0 & 0 & 0 & \frac{95}{41} & \frac{-1490}{287} & \frac{37000}{287} \\
\frac{-171}{287} & \frac{-31}{287} & 0 & 1 & 0 & 0 & \frac{-291}{41} & \frac{1433}{287} & \frac{17000}{287}
\end{array}\right]$$

$$\left[\begin{array}{ccccccccc}
\mathit{\_SL1} & \mathit{\_SL2} & \mathit{\_SL3} & \mathit{\_SL4} & \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{x4} & \mathit{Z} \\
\frac{-1}{307} & \frac{-126}{307} & 0 & 0 & \frac{-1025}{307} & 0 & \frac{332}{307} & 0 & \frac{25400}{307} \\
\frac{17}{307} & \frac{-7}{307} & 0 & 0 & \frac{-74}{307} & 1 & \frac{189}{307} & 0 & \frac{2000}{307} \\
\frac{-12}{307} & \frac{23}{307} & 0 & 0 & \frac{287}{307} & 0 & \frac{-7}{307} & 1 & \frac{2200}{307} \\
\frac{-27}{307} & \frac{-25}{307} & 1 & 0 & \frac{1490}{307} & 0 & \frac{675}{307} & 0 & \frac{51000}{307} \\
\frac{-123}{307} & \frac{-148}{307} & 0 & 1 & \frac{-1433}{307} & 0 & \frac{-2144}{307} & 0 & \frac{7200}{307}
\end{array}\right]$$

$$\left[\begin{array}{ccccccccc}
\mathit{\_SL1} & \mathit{\_SL2} & \mathit{\_SL3} & \mathit{\_SL4} & \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{x4} & \mathit{Z} \\
\frac{-19}{189} & \frac{-10}{27} & 0 & 0 & \frac{-551}{189} & \frac{-332}{189} & 0 & 0 & \frac{17800}{189} \\
\frac{17}{189} & \frac{-1}{27} & 0 & 0 & \frac{-74}{189} & \frac{307}{189} & 1 & 0 & \frac{2000}{189} \\
\frac{-1}{27} & \frac{2}{27} & 0 & 0 & \frac{25}{27} & \frac{1}{27} & 0 & 1 & \frac{200}{27} \\
\frac{-2}{7} & 0 & 1 & 0 & \frac{40}{7} & \frac{-25}{7} & 0 & 0 & \frac{1000}{7} \\
\frac{43}{189} & \frac{-20}{27} & 0 & 1 & \frac{-1399}{189} & \frac{2144}{189} & 0 & 0 & \frac{18400}{189}
\end{array}\right]$$

$$\mathit{finished} : \ \mathit{no\ pivot\ variable.}$$

$$\begin{aligned}
\mathit{sol} := [[\mathit{x3} &= -\frac{307}{189}\,\mathit{x2} - \frac{17}{189}\,\mathit{\_SL1} + \frac{2000}{189} + \frac{1}{27}\,\mathit{\_SL2} + \frac{74}{189}\,\mathit{x1}, \\
\mathit{x4} &= -\frac{25}{27}\,\mathit{x1} - \frac{2}{27}\,\mathit{\_SL2} + \frac{200}{27} + \frac{1}{27}\,\mathit{\_SL1} - \frac{1}{27}\,\mathit{x2},\ \mathit{\_SL3} = \frac{1000}{7} + \frac{2}{7}\,\mathit{\_SL1} + \frac{25}{7}\,\mathit{x2} - \frac{40}{7}\,\mathit{x1}, \\
\mathit{\_SL4} &= \frac{18400}{189} + \frac{20}{27}\,\mathit{\_SL2} - \frac{43}{189}\,\mathit{\_SL1} - \frac{2144}{189}\,\mathit{x2} + \frac{1399}{189}\,\mathit{x1}], \\
&-\frac{10}{27}\,\mathit{\_SL2} + \frac{17800}{189} - \frac{19}{189}\,\mathit{\_SL1} - \frac{332}{189}\,\mathit{x2} - \frac{551}{189}\,\mathit{x1}, \\
&[[\mathit{x1},\ \mathit{\_SL2}],\ [\mathit{x2},\ \mathit{\_SL1}],\ [\mathit{x4},\ \mathit{x1}],\ [\mathit{x3},\ \mathit{x2}]]]
\end{aligned}$$

8 Two Illustrative Problems

Problem 1:

This problem llustrates how to use the simplex method to demonstrate that the objective function may be unbounded on the feasible set.

Maximize the objective subject to the constraints (all variables assumed nonnegative.)

```
obj := 5*x1 + x2 + 3*x3 + 4*x4;
cnsts := [x1 - 2*x2 + 4*x3 + 3*x4 <= 20,
-4*x1 + 6*x2 + 5*x3 - 4*x4 <= 40,
2*x1 - 3*x2 + 3*x3 + 8*x4 <= 50];
```

$$obj := 5\,x1 + x2 + 3\,x3 + 4\,x4$$

$$cnsts := [x1 - 2\,x2 + 4\,x3 + 3\,x4 \leq 20,\ -4\,x1 + 6\,x2 + 5\,x3 - 4\,x4 \leq 40,\ 2\,x1 - 3\,x2 + 3\,x3 + 8\,x4 \leq 50]$$

```
with(linalg):with(simplex):
eqns := setup(cnsts);
```

$$eqns := [\_SL1 = 20 - x1 + 2\,x2 - 4\,x3 - 3\,x4,\ \_SL2 = 40 + 4\,x1 - 6\,x2 - 5\,x3 + 4\,x4,\ \_SL3 = 50 - 2\,x1 + 3\,x2 - 3\,x3 - 8\,x4]$$

We will use the mymax command from the previous worksheet.

```
mymax(eqns,obj);
```

$$\begin{bmatrix} \_SL1 & \_SL2 & \_SL3 & x1 & x2 & x3 & x4 & Z \\ -5 & 0 & 0 & 0 & 11 & -17 & -11 & 100 \\ 1 & 0 & 0 & 1 & -2 & 4 & 3 & 20 \\ 4 & 1 & 0 & 0 & -2 & 21 & 8 & 120 \\ -2 & 0 & 1 & 0 & 1 & -5 & 2 & 10 \end{bmatrix}$$

$$\begin{bmatrix} \_SL1 & \_SL2 & \_SL3 & x1 & x2 & x3 & x4 & Z \\ 17 & 0 & -11 & 0 & 0 & 38 & -33 & 210 \\ -3 & 0 & 2 & 1 & 0 & -6 & 7 & 40 \\ 0 & 1 & 2 & 0 & 0 & 11 & 12 & 140 \\ -2 & 0 & 1 & 0 & 1 & -5 & 2 & 10 \end{bmatrix}$$

$$\begin{bmatrix} \_SL1 & \_SL2 & \_SL3 & x1 & x2 & x3 & x4 & Z \\ 17 & \frac{-38}{11} & \frac{-197}{11} & 0 & 0 & 0 & \frac{-819}{11} & \frac{7630}{11} \\ -3 & \frac{6}{11} & \frac{34}{11} & 1 & 0 & 0 & \frac{149}{11} & \frac{1280}{11} \\ 0 & \frac{1}{11} & \frac{2}{11} & 0 & 0 & 1 & \frac{12}{11} & \frac{140}{11} \\ -2 & \frac{5}{11} & \frac{21}{11} & 0 & 1 & 0 & \frac{82}{11} & \frac{810}{11} \end{bmatrix}$$

$$finished :\ no\ pivot\ equation.$$

$$\begin{aligned} &[[x1 = 3\,\_SL1 + \frac{1280}{11} - \frac{34}{11}\,\_SL3 - \frac{6}{11}\,\_SL2 - \frac{149}{11}\,x4, \\ &x3 = -\frac{1}{11}\,\_SL2 + \frac{140}{11} - \frac{2}{11}\,\_SL3 - \frac{12}{11}\,x4, \\ &x2 = -\frac{21}{11}\,\_SL3 + \frac{810}{11} + 2\,\_SL1 - \frac{5}{11}\,\_SL2 - \frac{82}{11}\,x4], \\ &17\,\_SL1 + \frac{7630}{11} - \frac{197}{11}\,\_SL3 - \frac{38}{11}\,\_SL2 - \frac{819}{11}\,x4,\ [[x1, \_SL1], [x2, \_SL3], [x3, \_SL2]]] \end{aligned}$$

After 3 pivots, we see by looking at the objective that by increasing sl1, obj increases without bound since the values of the basic variables x1 and x2 actually increase as sl1 increases and the value of x3 stays unchanged at 140/11.

- **Problem 2:** This problem illustrates how one can have multiple corner optimal points.

Maximize the objective subject to the constraints (all variables assumed non negative.)

```
obj := x1 + x2 + x3 +x4;
cnsts := [x1 + x2 <= 3, x3 + x4 <= 2];
```

$$\mathit{obj} := \mathit{x1} + \mathit{x2} + \mathit{x3} + \mathit{x4}$$

$$\mathit{cnsts} := [\mathit{x1} + \mathit{x2} \leq 3,\ \mathit{x3} + \mathit{x4} \leq 2]$$

```
eqns := setup(cnsts);
```

$$\mathit{eqns} := [\mathit{\_SL1} = 3 - \mathit{x1} - \mathit{x2},\ \mathit{\_SL2} = 2 - \mathit{x3} - \mathit{x4}]$$

```
sol :=mymax(eqns,obj);
```

$$\begin{bmatrix} \mathit{\_SL1} & \mathit{\_SL2} & \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{x4} & \mathit{Z} \\ -1 & 0 & 0 & 0 & 1 & 1 & 3 \\ 1 & 0 & 1 & 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\begin{bmatrix} \mathit{\_SL1} & \mathit{\_SL2} & \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{x4} & \mathit{Z} \\ -1 & -1 & 0 & 0 & 0 & 0 & 5 \\ 1 & 0 & 1 & 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 1 & 1 & 2 \end{bmatrix}$$

$$\mathit{finished : \ no\ pivot\ variable.}$$

$$\mathit{sol} :=$$
$$[[\mathit{x1} = -\mathit{\_SL1} + 3 - \mathit{x2},\ \mathit{x3} = -\mathit{\_SL2} + 2 - \mathit{x4}],\ -\mathit{\_SL1} + 5 - \mathit{\_SL2},\ [[\mathit{x1},\ \mathit{\_SL1}],\ [\mathit{x3},\ \mathit{\_SL2}]]]$$

So we have an optimum pt at (x1,x2, x3,x4)=(3,0,0,2). We can get another by choosing x2 to be the pivot variable and the first equation as the pivot equation.

```
eqns :=pivot(sol[1],x2,sol[1][1]);
```

$$\mathit{eqns} := [\mathit{x2} = -\mathit{\_SL1} + 3 - \mathit{x1},\ \mathit{x3} = -\mathit{\_SL2} + 2 - \mathit{x4}]$$

The objective doesn't change since x2 does not appear in the current form of the objective. Now we have an optimum pt at (0,3,0,2). And another optimum point occurs at (0,3,2,0) and also at (3,0,2,0). This seems to be all of them.

9 Big M vs two phase methods for getting a feasible solution.

Problem Maximize $2\,x1 + 5\,x2 + 3\,x3$ subject to $20 \leq x1 - 2\,x2 + x3$, and $2\,x1 + 4\,x2 + x3 = 50$, $0 \leq xj$ for j = 1,2,3.

Use both the big M method and the Phase 1/Phase 2 method. Also do it graphically by eliminating x3.

```
with(linalg):  with(simplex):
```

```
Warning, new definition for norm

Warning, new definition for trace

Warning, new definition for basis

Warning, new definition for maximize

Warning, new definition for minimize

Warning, new definition for pivot
```

We first introduce a surplus variable s1 to turn the first constraint into an equation and then add an artificial variable a1 as an initial basic variable. $20 \leq x1 - 2\,x2 + x3$ converts to $a1 = 20 - x1 + 2\,x2 - x3 + s1$. Then an artificial variable a2 is introduced into the second constraint an initial basic variable. So the constraints are now converted to the equations:

```
eqns := [a1 = 20 -x1 + 2*x2 -x3 + s1,a2 =
50 -2*x1-4*x2-x3 ];
```

$$eqns := [a1 = 20 - x1 + 2\,x2 - x3 + s1,\ a2 = 50 - 2\,x1 - 4\,x2 - x3]$$

The big M method

Here the objective is modified by subtracting an indeterminately large multiple of each artificial variable. This will force these variables out of solution if the original problem has a solution. (The original set of constraints could be inconsistent).

```
obj := 2*x1 + 5*x2 + 3*x3 - M*a1 -M*a2;
```

We can use tabular from mymax to display the equations for making pivoting decisions.

```
tabular(eqns,obj);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ -M & -M & 0 & 2 & 5 & 3 & 0 \\ 1 & 0 & -1 & 1 & -2 & 1 & 20 \\ 0 & 1 & 0 & 2 & 4 & 1 & 50 \end{bmatrix}$$

Actually, we need to eliminate the basic variables from the objective so that we can decide what the possible pivot variables (entering basic variables) are.

```
obj :=
collect(simplify(subs(eqns,obj)),[x1,x2,x3]);
```

$$obj := (2 + 3\,M)\,x1 + (2\,M + 5)\,x2 + (2\,M + 3)\,x3 - 70\,M - M\,s1$$

```
tabular(eqns,obj);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ 0 & 0 & -M & 2 + 3\,M & 2\,M + 5 & 2\,M + 3 & -70\,M \\ 1 & 0 & -1 & 1 & -2 & 1 & 20 \\ 0 & 1 & 0 & 2 & 4 & 1 & 50 \end{bmatrix}$$

Looking at either the table (either table) or the equations, we see that bringing any of x1, x2, or x3 into solution will increase Z. For no good reason, we choose x2. Then we must pivot on the second equation (why? because as x2 increases, a1 increases and a2 decreases. We can allow x2 to increase only to 50/4 before a2 becomes 0).

So, make the pivot, using pivot from the simplex package.

```
eqns2 := pivot(eqns,x2,eqns[2]);
```

$$\mathit{eqns2} := [\mathit{a1} = 45 - 2\,\mathit{x1} - \frac{1}{2}\,\mathit{a2} - \frac{3}{2}\,\mathit{x3} + \mathit{s1},\ \mathit{x2} = -\frac{1}{4}\,\mathit{a2} + \frac{25}{2} - \frac{1}{2}\,\mathit{x1} - \frac{1}{4}\,\mathit{x3}]$$

Again, eliminate the basic variables from the objective.

```
obj2 :=
collect(simplify(subs(eqns2,obj)),[x1,x3,a2]);
```

$$\mathit{obj2} := (2\,M - \frac{1}{2})\,\mathit{x1} + (\frac{3}{2}\,M + \frac{7}{4})\,\mathit{x3} + (-\frac{5}{4} - \frac{1}{2}\,M)\,\mathit{a2} - M\,\mathit{s1} - 45\,M + \frac{125}{2}$$

Make a table to facilitate decisions on the next iteration.

```
tabular(eqns2,obj2);
```

$$\left[\begin{array}{ccccccc}
\mathit{a1} & \mathit{a2} & \mathit{s1} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\
0 & -\frac{5}{4} - \frac{1}{2}\,M & -M & 2\,M - \frac{1}{2} & 0 & \frac{3}{2}\,M + \frac{7}{4} & \frac{125}{2} - 45\,M \\
1 & \frac{1}{2} & -1 & 2 & 0 & \frac{3}{2} & 45 \\
0 & \frac{1}{4} & 0 & \frac{1}{2} & 1 & \frac{1}{4} & \frac{25}{2}
\end{array}\right]$$

So, we see that we could pivot x3 and the first equation.

```
eqns3 := pivot(eqns2,x3,eqns2[1]);
```

$$\mathit{eqns3} := [\mathit{x3} = -\frac{2}{3}\,\mathit{a1} + 30 - \frac{4}{3}\,\mathit{x1} - \frac{1}{3}\,\mathit{a2} + \frac{2}{3}\,\mathit{s1},\ \mathit{x2} = -\frac{1}{6}\,\mathit{a2} + 5 - \frac{1}{6}\,\mathit{x1} + \frac{1}{6}\,\mathit{a1} - \frac{1}{6}\,\mathit{s1}]$$

```
obj3 :=
collect(simplify(subs(eqns3,obj2)),[x1,a1,a2]);
```

$$\mathit{obj3} := -\frac{17}{6}\,\mathit{x1} + (-\frac{7}{6} - M)\,\mathit{a1} + (-M - \frac{11}{6})\,\mathit{a2} + 115 + \frac{7}{6}\,\mathit{s1}$$

```
tabular(eqns3,obj3);
```

$$\left[\begin{array}{ccccccc}
\mathit{a1} & \mathit{a2} & \mathit{s1} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\
-\frac{7}{6} - M & -M - \frac{11}{6} & \frac{7}{6} & \frac{-17}{6} & 0 & 0 & 115 \\
\frac{2}{3} & \frac{1}{3} & \frac{-2}{3} & \frac{4}{3} & 0 & 1 & 30 \\
\frac{-1}{6} & \frac{1}{6} & \frac{1}{6} & \frac{1}{6} & 1 & 0 & 5
\end{array}\right]$$

Pivot on s1 and the second equation.

```
eqns4 := pivot(eqns3,s1,eqns3[2]);
```

$$\mathit{eqns4} := [\mathit{x3} = 50 - 2\,\mathit{x1} - \mathit{a2} - 4\,\mathit{x2},\ \mathit{s1} = -6\,\mathit{x2} - \mathit{a2} + 30 - \mathit{x1} + \mathit{a1}]$$

```
obj4 :=
collect(simplify(subs(eqns4,obj3)),[x1,x2,a2]);
```

$$\mathit{obj4} := -4\,\mathit{x1} - 7\,\mathit{x2} + (-M - 3)\,\mathit{a2} + 150 - M\,\mathit{a1}$$

At this stage we are done. obj is maximized at 150 when (x1,x2,x3,s1,a1,a2) is (0,0,50,30,0,0).

```
tabular(eqns4,obj4);
```

$$\left[\begin{array}{ccccccc}
\mathit{a1} & \mathit{a2} & \mathit{s1} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\
-M & -M - 3 & 0 & -4 & -7 & 0 & 150 \\
0 & 1 & 0 & 2 & 4 & 1 & 50 \\
-1 & 1 & 1 & 1 & 6 & 0 & 30
\end{array}\right]$$

Phase 1 - Phase 2 method

Here, the idea is to first maximize the negative of the sum of the artificial variables as the first phase. If that maximizes to 0 then both a1 and a2 are zero and we will have a feasible solution to the original problem.

Start of Phase 1

```
aobj := -a1 -a2;
```

$$aobj := -a1 - a2$$

```
aobj := subs(eqns,aobj);
```

$$aobj := -70 + 3\,x1 + 2\,x2 + 2\,x3 - s1$$

```
eqns;
```

$$[a1 = 20 - x1 + 2\,x2 - x3 + s1,\ a2 = 50 - 2\,x1 - 4\,x2 - x3]$$

```
tabular(eqns,aobj);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ 0 & 0 & -1 & 3 & 2 & 2 & -70 \\ 1 & 0 & -1 & 1 & -2 & 1 & 20 \\ 0 & 1 & 0 & 2 & 4 & 1 & 50 \end{bmatrix}$$

```
eqns2 := pivot(eqns,x1,eqns[1]);
```

$$eqns2 := [x1 = -a1 + 20 + 2\,x2 - x3 + s1,\ a2 = 10 + 2\,a1 - 8\,x2 + x3 - 2\,s1]$$

```
aobj2 := subs(eqns2,aobj);
```

$$aobj2 := -10 - 3\,a1 + 8\,x2 - x3 + 2\,s1$$

```
tabular(eqns2,aobj2);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ -3 & 0 & 2 & 0 & 8 & -1 & -10 \\ 1 & 0 & -1 & 1 & -2 & 1 & 20 \\ -2 & 1 & 2 & 0 & 8 & -1 & 10 \end{bmatrix}$$

```
eqns3 := pivot(eqns2,x2,eqns2[2]);
```

$$eqns3 := [x1 = -\frac{1}{2}\,a1 + \frac{45}{2} - \frac{1}{4}\,a2 - \frac{3}{4}\,x3 + \frac{1}{2}\,s1,\ x2 = -\frac{1}{8}\,a2 + \frac{5}{4} + \frac{1}{4}\,a1 + \frac{1}{8}\,x3 - \frac{1}{4}\,s1]$$

```
aobj3 := subs(eqns3,aobj2);
```

$$aobj3 := -a1 - a2$$

```
tabular(eqns3,aobj3);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{4} & \frac{-1}{2} & 1 & 0 & \frac{3}{4} & \frac{45}{2} \\ \frac{-1}{4} & \frac{1}{8} & \frac{1}{4} & 0 & 1 & \frac{-1}{8} & \frac{5}{4} \end{bmatrix}$$

We have maximized aobj to 0. This is the end of phase 1.

We have a feasible solution to the original problem. x1=45/2, x2=5/4. z= 0

So set the objective back to the original and remove a1 and a2 from the set of nonbasic variables (i.e., set them to 0 in eqns and forget 'em).

```
obj := 2*x1 + 5*x2 + 3*x3;
```

$$obj := 2\,x1 + 5\,x2 + 3\,x3$$

```
eqns:=subs({a1=0,a2=0},eqns3);
```

$$eqns := [x1 = \frac{45}{2} - \frac{3}{4}\,x3 + \frac{1}{2}\,s1,\ x2 = \frac{5}{4} + \frac{1}{8}\,x3 - \frac{1}{4}\,s1]$$

```
tabular(eqns,obj);
```

$$\left[\begin{array}{ccccc} s1 & x1 & x2 & x3 & Z \\ 0 & 2 & 5 & 3 & 0 \\ \frac{-1}{2} & 1 & 0 & \frac{3}{4} & \frac{45}{2} \\ \frac{1}{4} & 0 & 1 & \frac{-1}{8} & \frac{5}{4} \end{array}\right]$$

What pivot var? x3 (s1 could be used, but Z would not increase) What pivoteqn? equation 1, since increasing x3 will decrease x1, but increase x2.

```
eqns2 :=pivot(eqns,x3,eqns[1]);
```

$$eqns2 := [x3 = -\frac{4}{3}\,x1 + 30 + \frac{2}{3}\,s1,\ x2 = 5 - \frac{1}{6}\,x1 - \frac{1}{6}\,s1]$$

```
obj2 := subs(eqns2,obj);
```

$$obj2 := -\frac{17}{6}\,x1 + 115 + \frac{7}{6}\,s1$$

```
tabular(eqns2,obj2);
```

$$\left[\begin{array}{ccccc} s1 & x1 & x2 & x3 & Z \\ \frac{7}{6} & \frac{-17}{6} & 0 & 0 & 115 \\ \frac{-2}{3} & \frac{4}{3} & 0 & 1 & 30 \\ \frac{1}{6} & \frac{1}{6} & 1 & 0 & 5 \end{array}\right]$$

What pivot var ? s1. What pivot eqn? the second one.

```
eqns3 := pivot(eqns2,s1,eqns2[2]);
```

$$eqns3 := [x3 = -2\,x1 + 50 - 4\,x2,\ s1 = -6\,x2 + 30 - x1]$$

```
obj3 := subs(eqns3,obj2);
```

$$obj3 := -4\,x1 - 7\,x2 + 150$$

```
tabular(eqns3,obj3);
```

$$\left[\begin{array}{ccccc} s1 & x1 & x2 & x3 & Z \\ 0 & -4 & -7 & 0 & 150 \\ 0 & 2 & 4 & 1 & 50 \\ 1 & 1 & 6 & 0 & 30 \end{array}\right]$$

Why are we done? Because increasing either nonbasic variable (x1 or x3) results in a decrease in the objective Z.

Graphical

Here we eliminate the variable x3 by solving the second constraint for x3

```
cnsts :=[20<=x1-2*x2+x3 ,
50=2*x1+4*x2+x3];
```

$$cnsts := [20 \leq x1 - 2\,x2 + x3,\ 50 = 2\,x1 + 4\,x2 + x3]$$

```
elim :=solve(cnsts[2],{x3});
```

$$elim := \{x3 = -2\,x1 + 50 - 4\,x2\}$$

```
   neqns :=
{subs(elim,cnsts[1]),0=rhs(op(elim))};
```

$$neqns := \{0 = -2\,x1 + 50 - 4\,x2,\ 20 \leq -x1 - 6\,x2 + 50\}$$

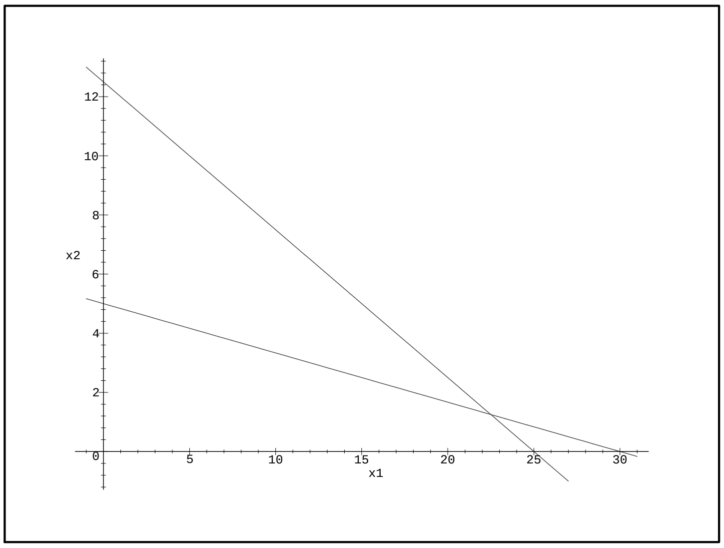
We see that x1=x20 =0 is a feasible soln

```
nobj := subs(elim,obj);
```

$$nobj := -4\,x1 - 7\,x2 + 150$$

By inspection, nobj is maximum 150 when x1=x2=0 and x3 =50.

```
plots[implicitplot](neqns,x1=-1..31,x2=-1..26
);
```



10 Pig problem Revisited

```
restart;with(linalg);
```

Problem: A pig farmer has resources of corn, tankage, and hay to make up a pig diet which satisfies certain minimum daily requirements as to the basic nutrients of carbohydrates, proteins, and vitamins. The table showing the minimum amounts of each nutrient needed and the amount of each nutrient in a kilogram of resource and the cost of the resource is given below. Desired is a diet which satisfies the minimum daily requirement at minimum cost.

```
tab := matrix(5,5,[
`Nutrients`,`corn(Kg)`,`tankage(Kg)`,`hay(Kg)`,`mdr`,
`carbo`,90,20,40,200,
`protein`,30,80,60,180,
`vitamins`,10,20,60,150,
`cost(.01$)`,35,30,25,` `]);
```

$$tab := \begin{bmatrix} \mathit{Nutrients} & \mathit{corn(Kg)} & \mathit{tankage(Kg)} & \mathit{hay(Kg)} & \mathit{mdr} \\ \mathit{carbo} & 90 & 20 & 40 & 200 \\ \mathit{protein} & 30 & 80 & 60 & 180 \\ \mathit{vitamins} & 10 & 20 & 60 & 150 \\ \mathit{cost(.01\$)} & 35 & 30 & 25 & \end{bmatrix}$$

Setup: Let c, t, and h denote the number of kg of corn, tankage, and hay per pig in the diet. Then the constraints are linear inequalities in these quanitities.

```
cnsts := {90*c + 20*t + 40*h >=200,
30*c + 80*t + 60*h >=180,
10*c + 20*t + 60*h >=150};
```

$$cnsts := \{200 \leq 90\,c + 20\,t + 40\,h,\ 180 \leq 30\,c + 80\,t + 60\,h,\ 150 \leq 10\,c + 20\,t + 60\,h\}$$

The cost is a linear function of these quantitities.

```
cost := 35*c + 30*t + 25*h;
```

$$cost := 35\,c + 30\,t + 25\,h$$

We can use the simplex package to solve this problem.

```
with(simplex);
```

```
Warning, new definition for basis

Warning, new definition for convexhull

Warning, new definition for cterm

Warning, new definition for display

Warning, new definition for dual

Warning, new definition for feasible

Warning, new definition for maximize

Warning, new definition for pivoteqn

Warning, new definition for pivotvar

Warning, new definition for ratio

Warning, new definition for setup
```

$$[\mathit{basis},\ \mathit{convexhull},\ \mathit{cterm},\ \mathit{define\_zero},\ \mathit{display},\ \mathit{dual},\ \mathit{feasible},\ \mathit{maximize},\ \mathit{minimize},\ \mathit{pivot},\ \mathit{pivoteqn},\ \mathit{pivotvar},\ \mathit{ratio},\ \mathit{setup},\ \mathit{standardize}]$$

```
sol := minimize(cost,cnsts,NONNEGATIVE);
```

$$\mathit{sol} := \{t = 0,\ c = \frac{8}{7},\ h = \frac{17}{7}\}$$

```
subs(sol,[cost,cnsts]); evalf(");
```

$$[\frac{705}{7},\ \{200 \leq 200,\ 180 \leq 180,\ 150 \leq \frac{1100}{7}\}]$$

$$[100.7142857,\ \{200. \leq 200.,\ 180. \leq 180.,\ 150. \leq 157.1428571\}]$$

So, each pig should get no tankage, 17/7 kg of hay, and 8/7 kg of corn each day at at cost of about 1100/7 dollars. The pig gets his minimum carbo and protein, and about 7 units over his minimum vitamins.

```
eqns := [a1
=lhs(cnsts[1])-rhs(cnsts[1])+s1,
a2 =lhs(cnsts[2])-rhs(cnsts[2])+s2,
a3 =lhs(cnsts[3])-rhs(cnsts[3])+s3];
```

$$\mathit{eqns} := [\mathit{a1} = 200 - 90\,c - 20\,t - 40\,h + \mathit{s1},\ \mathit{a2} = 180 - 30\,c - 80\,t - 60\,h + \mathit{s2},\ \mathit{a3} = 150 - 10\,c - 20\,t - 60\,h + \mathit{s3}]$$

```
phase1cost := subs(eqns,a1 +a2 +a3);
```

$$\mathit{phase1cost} := 530 - 130\,c - 120\,t - 160\,h + \mathit{s1} + \mathit{s2} + \mathit{s3}$$

```
tabular(eqns,phase1cost);
```

$$\begin{bmatrix} \mathit{a1} & \mathit{a2} & \mathit{a3} & c & h & \mathit{s1} & \mathit{s2} & \mathit{s3} & t & Z \\ 0 & 0 & 0 & -130 & -160 & 1 & 1 & 1 & -120 & 530 \\ 1 & 0 & 0 & 90 & 40 & -1 & 0 & 0 & 20 & 200 \\ 0 & 1 & 0 & 30 & 60 & 0 & -1 & 0 & 80 & 180 \\ 0 & 0 & 1 & 10 & 60 & 0 & 0 & -1 & 20 & 150 \end{bmatrix}$$

```
pivotvar := h;
```

$$\mathit{pivotvar} := h$$

```
pivoteqn := eqns[3];
```

$$\mathit{pivoteqn} := \mathit{a3} = 150 - 10\,c - 20\,t - 60\,h + \mathit{s3}$$

```
pivoteqn := op(solve(pivoteqn
,{pivotvar})) ;
```

$$\mathit{pivoteqn} := h = -\frac{1}{60}\,\mathit{a3} + \frac{5}{2} - \frac{1}{6}\,c - \frac{1}{3}\,t + \frac{1}{60}\,\mathit{s3}$$

```
eqns2 :=
[subs(pivoteqn,eqns[1]),subs(pivoteqn,eqns[2]),pivoteqn];
```

$$\mathit{eqns2} := [\mathit{a1} = 100 - \frac{250}{3}\,c - \frac{20}{3}\,t + \frac{2}{3}\,\mathit{a3} - \frac{2}{3}\,\mathit{s3} + \mathit{s1},\ \mathit{a2} = 30 - 20\,c - 60\,t + \mathit{a3} - \mathit{s3} + \mathit{s2},\ h = -\frac{1}{60}\,\mathit{a3} + \frac{5}{2} - \frac{1}{6}\,c - \frac{1}{3}\,t + \frac{1}{60}\,\mathit{s3}]$$

```
phase1cost := subs(pivoteqn,phase1cost)
;
```

$$\mathit{phase1cost} := 130 - \frac{310}{3}\,c - \frac{200}{3}\,t + \frac{8}{3}\,\mathit{a3} - \frac{5}{3}\,\mathit{s3} + \mathit{s1} + \mathit{s2}$$

```
tabular(eqns2,phase1cost);
```

$$\left[\begin{array}{cccccccccc} \mathit{a1} & \mathit{a2} & \mathit{a3} & c & h & \mathit{s1} & \mathit{s2} & \mathit{s3} & t & Z \\ 0 & 0 & \frac{8}{3} & \frac{-310}{3} & 0 & 1 & 1 & \frac{-5}{3} & \frac{-200}{3} & 130 \\ 1 & 0 & \frac{-2}{3} & \frac{250}{3} & 0 & -1 & 0 & \frac{2}{3} & \frac{20}{3} & 100 \\ 0 & 1 & -1 & 20 & 0 & 0 & -1 & 1 & 60 & 30 \\ 0 & 0 & \frac{1}{60} & \frac{1}{6} & 1 & 0 & 0 & \frac{-1}{60} & \frac{1}{3} & \frac{5}{2} \end{array}\right]$$

```
pivotvar := c;
```

$$\mathit{pivotvar} := c$$

```
ratios := 100/(250/3),30/20, (5/2)/(1/6);
```

$$\mathit{ratio} := \frac{6}{5},\ \frac{3}{2},\ 15$$

```
pivoteqn := eqns2[1];
```

$$\mathit{pivoteqn} := \mathit{a1} = 100 - \frac{250}{3}\,c - \frac{20}{3}\,t + \frac{2}{3}\,\mathit{a3} - \frac{2}{3}\,\mathit{s3} + \mathit{s1}$$

```
pivoteqn := op( solve(pivoteqn ,{pivotvar})
);
```

$$\mathit{pivoteqn} := c = -\frac{3}{250}\,\mathit{a1} + \frac{6}{5} + \frac{3}{250}\,\mathit{s1} - \frac{2}{25}\,t + \frac{1}{125}\,\mathit{a3} - \frac{1}{125}\,\mathit{s3}$$

```
eqns3 := [pivoteqn,subs(pivoteqn,eqns2[2]),
subs(pivoteqn,eqns2[3]) ];
```

$$\begin{gathered} \mathit{eqns3} := [c = -\frac{3}{250}\,\mathit{a1} + \frac{6}{5} + \frac{3}{250}\,\mathit{s1} - \frac{2}{25}\,t + \frac{1}{125}\,\mathit{a3} - \frac{1}{125}\,\mathit{s3}, \\ \mathit{a2} = 6 + \frac{6}{25}\,\mathit{a1} - \frac{6}{25}\,\mathit{s1} - \frac{292}{5}\,t + \frac{21}{25}\,\mathit{a3} - \frac{21}{25}\,\mathit{s3} + \mathit{s2}, \\ h = -\frac{9}{500}\,\mathit{a3} + \frac{23}{10} + \frac{1}{500}\,\mathit{a1} - \frac{1}{500}\,\mathit{s1} - \frac{8}{25}\,t + \frac{9}{500}\,\mathit{s3}] \end{gathered}$$

```
phase1cost := subs(pivoteqn,phase1cost) ;
```

$$\mathit{phase1cost} := 6 + \frac{31}{25}\,\mathit{a1} - \frac{6}{25}\,\mathit{s1} - \frac{292}{5}\,t + \frac{46}{25}\,\mathit{a3} - \frac{21}{25}\,\mathit{s3} + \mathit{s2}$$

```
tabular(eqns3,phase1cost);
```

$$\left[\begin{array}{cccccccccc} \mathit{a1} & \mathit{a2} & \mathit{a3} & c & h & \mathit{s1} & \mathit{s2} & \mathit{s3} & t & Z \\ \frac{31}{25} & 0 & \frac{46}{25} & 0 & 0 & \frac{-6}{25} & 1 & \frac{-21}{25} & \frac{-292}{5} & 6 \\ \frac{3}{250} & 0 & \frac{-1}{125} & 1 & 0 & \frac{-3}{250} & 0 & \frac{1}{125} & \frac{2}{25} & \frac{6}{5} \\ \frac{-6}{25} & 1 & \frac{-21}{25} & 0 & 0 & \frac{6}{25} & -1 & \frac{21}{25} & \frac{292}{5} & 6 \\ \frac{-1}{500} & 0 & \frac{9}{500} & 0 & 1 & \frac{1}{500} & 0 & \frac{-9}{500} & \frac{8}{25} & \frac{23}{10} \end{array}\right]$$

```
pivotvar := t;
```

$$\mathit{pivotvar} := t$$

```
ratios :=
6/5/(2/25),6/(292/5),23/10/(8/25);
```

$$\mathit{ratio} := 15,\ \frac{15}{146},\ \frac{115}{16}$$

```
pivoteqn := eqns3[2];
```

$$\mathit{pivoteqn} := \mathit{a2} = 6 + \frac{6}{25}\,\mathit{a1} - \frac{6}{25}\,\mathit{s1} - \frac{292}{5}\,t + \frac{21}{25}\,\mathit{a3} - \frac{21}{25}\,\mathit{s3} + \mathit{s2}$$

```
pivoteqn := op( solve(pivoteqn ,{pivotvar})
);
```

$$\mathit{pivoteqn} := t = -\frac{5}{292}\,\mathit{a2} + \frac{15}{146} + \frac{3}{730}\,\mathit{a1} - \frac{3}{730}\,\mathit{s1} + \frac{21}{1460}\,\mathit{a3} - \frac{21}{1460}\,\mathit{s3} + \frac{5}{292}\,\mathit{s2}$$

```
eqns4 := [subs(pivoteqn,eqns3[1]),pivoteqn,
subs(pivoteqn,eqns3[3])];
```

$$\begin{aligned}\mathit{eqns4} := [c = -\frac{9}{730}\,\mathit{a1} + \frac{87}{73} + \frac{9}{730}\,\mathit{s1} + \frac{1}{730}\,\mathit{a2} + \frac{1}{146}\,\mathit{a3} - \frac{1}{146}\,\mathit{s3} - \frac{1}{730}\,\mathit{s2},\\ t = -\frac{5}{292}\,\mathit{a2} + \frac{15}{146} + \frac{3}{730}\,\mathit{a1} - \frac{3}{730}\,\mathit{s1} + \frac{21}{1460}\,\mathit{a3} - \frac{21}{1460}\,\mathit{s3} + \frac{5}{292}\,\mathit{s2},\\ h = -\frac{33}{1460}\,\mathit{a3} + \frac{331}{146} + \frac{1}{1460}\,\mathit{a1} - \frac{1}{1460}\,\mathit{s1} + \frac{2}{365}\,\mathit{a2} + \frac{33}{1460}\,\mathit{s3} - \frac{2}{365}\,\mathit{s2}]\end{aligned}$$

```
phase1cost := subs(pivoteqn,phase1cost);
```

$$\mathit{phase1cost} := \mathit{a1} + \mathit{a2} + \mathit{a3}$$

```
tabular(eqns4,phase1cost);
```

$$\left[\begin{array}{cccccccccc} \mathit{a1} & \mathit{a2} & \mathit{a3} & c & h & \mathit{s1} & \mathit{s2} & \mathit{s3} & t & Z \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \frac{9}{730} & \frac{-1}{730} & \frac{-1}{146} & 1 & 0 & \frac{-9}{730} & \frac{1}{730} & \frac{1}{146} & 0 & \frac{87}{73} \\ \frac{-3}{730} & \frac{5}{292} & \frac{-21}{1460} & 0 & 0 & \frac{3}{730} & \frac{-5}{292} & \frac{21}{1460} & 1 & \frac{15}{146} \\ \frac{-1}{1460} & \frac{-2}{365} & \frac{33}{1460} & 0 & 1 & \frac{1}{1460} & \frac{2}{365} & \frac{-33}{1460} & 0 & \frac{331}{146} \end{array}\right]$$

End of Phase 1

```
eqns := subs({a1=0,a2=0,a3=0},eqns4);
```

$$\begin{aligned}\mathit{eqns} := [c = \frac{87}{73} + \frac{9}{730}\,\mathit{s1} - \frac{1}{146}\,\mathit{s3} - \frac{1}{730}\,\mathit{s2},\ t = \frac{15}{146} - \frac{3}{730}\,\mathit{s1} - \frac{21}{1460}\,\mathit{s3} + \frac{5}{292}\,\mathit{s2},\\ h = \frac{331}{146} - \frac{1}{1460}\,\mathit{s1} + \frac{33}{1460}\,\mathit{s3} - \frac{2}{365}\,\mathit{s2}]\end{aligned}$$

```
cost := subs(eqns,cost);
```

$$\mathit{cost} := \frac{14815}{146} + \frac{85}{292}\,\mathit{s1} - \frac{31}{292}\,\mathit{s3} + \frac{24}{73}\,\mathit{s2}$$

```
tabular(eqns,cost);
```

$$\left[\begin{array}{ccccccc} c & h & \mathit{s1} & \mathit{s2} & \mathit{s3} & t & Z \\ 0 & 0 & \frac{85}{292} & \frac{24}{73} & \frac{-31}{292} & 0 & \frac{14815}{146} \\ 1 & 0 & \frac{-9}{730} & \frac{1}{730} & \frac{1}{146} & 0 & \frac{87}{73} \\ 0 & 0 & \frac{3}{730} & \frac{-5}{292} & \frac{21}{1460} & 1 & \frac{15}{146} \\ 0 & 1 & \frac{1}{1460} & \frac{2}{365} & \frac{-33}{1460} & 0 & \frac{331}{146} \end{array}\right]$$

```
pivotvar := s3;
```

$$\mathit{pivotvar} := \mathit{s3}$$

```
ratios :=
87/73/(1./146),21/1460/(15/146.);
```

$$\mathit{ratio} := 174.0000000,\ .1400000000$$

```
pivoteqn := eqns[2];
```

$$\mathit{pivoteqn} := t = \frac{15}{146} - \frac{3}{730}\,\mathit{s1} - \frac{21}{1460}\,\mathit{s3} + \frac{5}{292}\,\mathit{s2}$$

```
pivoteqn :=
op(solve(pivoteqn,{pivotvar}));
```

$$pivoteqn := s3 = -\frac{1460}{21}\,t + \frac{50}{7} - \frac{2}{7}\,s1 + \frac{25}{21}\,s2$$

```
eqns2 := [subs(pivoteqn,eqns[1]),pivoteqn,
subs(pivoteqn,eqns[3])];
```

$$eqns2 := [c = \frac{8}{7} + \frac{1}{70}\,s1 + \frac{10}{21}\,t - \frac{1}{105}\,s2,\ s3 = -\frac{1460}{21}\,t + \frac{50}{7} - \frac{2}{7}\,s1 + \frac{25}{21}\,s2,$$
$$h = \frac{17}{7} - \frac{1}{140}\,s1 - \frac{11}{7}\,t + \frac{3}{140}\,s2]$$

```
cost := subs(pivoteqn,cost);
```

$$cost := \frac{705}{7} + \frac{9}{28}\,s1 + \frac{155}{21}\,t + \frac{17}{84}\,s2$$

```
tabular(eqns2,cost);
```

$$\left[\begin{array}{ccccccc}
c & h & s1 & s2 & s3 & t & Z \\
0 & 0 & \frac{9}{28} & \frac{17}{84} & 0 & \frac{155}{21} & \frac{705}{7} \\
1 & 0 & \frac{-1}{70} & \frac{1}{105} & 0 & \frac{-10}{21} & \frac{8}{7} \\
0 & 0 & \frac{2}{7} & \frac{-25}{21} & 1 & \frac{1460}{21} & \frac{50}{7} \\
0 & 1 & \frac{1}{140} & \frac{-3}{140} & 0 & \frac{11}{7} & \frac{17}{7}
\end{array}\right]$$

End of phase 2. The minimum cost diet is 8/7 kg of corn, 17/7 kg hay, 0 tankage for a cost of 705/7 dollars.

```
evalf(");
```

$$\left[\begin{array}{ccccccc}
c & h & s1 & s2 & s3 & t & Z \\
0 & 0 & .3214285714 & .2023809524 & 0 & 7.380952381 & 100.7142857 \\
1. & 0 & -.01428571429 & .009523809524 & 0 & -.4761904762 & 1.142857143 \\
0 & 0 & .2857142857 & -1.190476190 & 1. & 69.52380952 & 7.142857143 \\
0 & 1. & .007142857143 & -.02142857143 & 0 & 1.571428571 & 2.428571429
\end{array}\right]$$

Questions:

1. How low would the price of tankage have to get before it would be used in the pig diet?

2. How low would the amount of vitamins in hay have to get before it becomes economical to use tankage in the diet?

3. Suppose there is only 2 kg of hay per pig available. What is the minimum cost pig diet then?

4. Suppose there is an additional resource of sardine meal costing 33 cents per kg and containing 60, 40, and 15 units of carbo, protein, and vitamins respectively. What is the minimum cost diet then?

11 Exam 1

Work 2 of 3. Due Wednesday Feb 19.

1. It was said in class one day that one could 'make up' a linear programminng problem where if you followed the natural tendency to chose for the pivot variable the one which increases the objective as quickly as possible, you would find it necessary to perform several iterations of the simplex algorithm before the objective is maximized whereas if you fought that natural tendency and instead chose another pivot variable you could maximize the objective in just one iteration of the simplex algorithm.

a) Make up an example with two decision variables and one $<=$ constraint to show this remark is true. Show your example works.

b) Make up an example with two decision variables and three constraints. Show your example works.

c) Make up a general example with 2 decision variables and 'n' constraints.

Argue that your example works.

2. A company makes three lines of tires. Its four-ply biased tires produce \$6 in profit per tire, its fiberglass belted line \$4 per tire, and its radials \$8 a tire. Each type of tire passes through three manufacturing stages as a part of the entire manufacturing process. Each of the three process centers has the following hours of available production time per day: Molding: 12 hours, Curing: 9 hours, and Assembly: 16 hours. The time required in each process to produce 100 tires of each line is as follows: Four-ply tires require 2 hours molding, 3 hours curing and 2 hours assembly ; fiberglass belteds need 2 hours molding, 2 hours curing and 1 hour assembly; radials must have 2 hours molding, 1 hour curing, and 3 hours assembly.

a. Determine the optimum product mix for each day's production, assuming that all tires manufactured are sold.

b. If you had a little more time to give to just one of the processes, which would you choose? Explain your decision in a lot of gory detail.

c. If you were permitted to increase the profit just a little bit per tire of just one of the lines, which one would you choose? Again, give gory details.

3. a) Modify the definition of 'tabular' in the worksheet mymax.mws so that it prints out a table that looks more like the tableau in our text.

b) The worksheet entitled pgfrm2.mws takes the pigfarmer problem and solves it 'by hand', using the simplex algorithm, but without making use of the simplex package in Maple. You will notice as you read through the worksheet that each iteration of the simplex algorithm has a similar form. Automate some of this repetitive work by defining one or more words. Use the worksheet mymax.mws as a guide. Test out your words.

12 Ma 416G Exam 1

12.1 Problem 1

1. It was said in class one day that one could 'make up' a linear programminng problem where if you followed the natural tendency to chose for the pivot variable the one which increases the objective as quickly as possible, you would find it necessary to perform several iterations of the simplex algorithm before the objective is maximized whereas if you fought that natural tendency and instead chose another pivot variable you could maximize the objective in just one iteration of the simplex algorithm.

a) Make up an example with two decision variables and one <= constraint to show this remark is true. Show your example works.

b) Make up an example with two decision variables and three constraints. Show your example works.

c) Make up a general example with 2 decision variables and 'n' constraints. Argue that your example works.

Solution: Take the objective function to be $2\,y + x$. We want to

make a feasible region with n constraints where the optimal corner point lies on the x axis. An easy way to do this would be to choose the corner points first. For a given positive integer n, define $f_n = 1 - (\frac{x}{n})^2$ to be the parabola with y-intercept 1 and x-intercepts n and -n. Choose the $n + 1$ points $p_i = (i,\, f_n(i))$ as i goes from 0 to n. Then together with (0,0) these will form the corner points of our feasible region. Now there are n <= contraints $y - (f_n(i + 1) - f_n(i))\, x \le f_n(i) - i\,(f_n(i + 1) - f_n(i))$ determined by the halfplanes below the lines through the points $p_i,\, p_{i+1}$ as i goes from 0 to n-1. If we introduce n slack variables s[i] and set up the equations with the initial solution x=0, y=0, s[i] = f[n](i)-i*(f[n](i+1)-f[n](i)), i = 1..n, then if we follow the natural tendency to bring y into solution first, then we have started around the feasible region in the clockwise sense toward to optimal point (n,0). It will take n pivots to get there. On the other hand, if we suppress this tendency for the quick profit, we will reach the optimal point in 1 pivot. To illustrate: first define f.

```
f := (x,n)->1-(x/n)^2;
```

Now the equations giving the boundaries of the constraint halfplanes are

```
eqs :=n->{seq( y -( f(i+1,n) -f(i,n))*x
=f(i,n)-i*( f(i+1,n)-f(i,n)),i=0..n-1)};
eqs(5);
```

So we can look at these equations graphically for specific values of n.

```
plots[implicitplot]( eqs(5)
,x=0..9,y=0..1.5);
```

Now put in the slack variables.

```
sleqns :=n-> [seq(s[i+1]=- y +( f(i+1,n)
-f(i,n))*x +f(i,n)-i*( f(i+1,n)-f(i,n)) ,i=0..n-1)] ;
sleqns(5);
obj1 := 2* y+x;
```

To illustrate our argument with n = 5.

```
eqns1 := sleqns(5);
```

Lets load in simplex to make use of ratios and pivot.

```
with(simplex);
```

Giving in to our natural tendencies, choose y as the pivot variable.

```
pivotvar := y;
ratios(eqns1,pivotvar);
pivoteqn := eqns1[1];
eqns2 :=pivot(eqns1,pivotvar,pivoteqn);
obj2 :=subs(eqns2,obj1);
```

So now we are on the merry clockwise chase. To get to the optimal point (5,0) we will need to pivot 5 more times.

```
pivotvar := x;
ratios(eqns2,pivotvar);
pivoteqn := eqns2[2];
eqns3 :=pivot(eqns2,pivotvar,pivoteqn);
obj3 :=subs(eqns3,obj2);
```

Thats one.

```
pivotvar := s[1];
ratios(eqns3,pivotvar);
pivoteqn := eqns3[3];
eqns4 :=pivot(eqns3,pivotvar,pivoteqn);
obj4 :=subs(eqns4,obj3);
```

And two.

```
pivotvar := s[2];
ratios(eqns4,pivotvar);
pivoteqn := eqns4[4];
eqns5 :=pivot(eqns4,pivotvar,pivoteqn);
obj5 :=subs(eqns5,obj4);
```

And 3.

```
pivotvar := s[3];
ratios(eqns5,pivotvar);
pivoteqn := eqns5[5];
eqns6 :=pivot(eqns5,pivotvar,pivoteqn);
obj6 :=subs(eqns6,obj5);
```

And 4.

```
pivotvar := s[4];
ratios(eqns6,pivotvar);
pivoteqn := eqns6[1];
eqns7 :=pivot(eqns6,pivotvar,pivoteqn);
obj7 :=subs(eqns7,obj6);
```

And we are done. Now if we had chosen x as the pivot variable in the first place:

```
pivotvar := x;
ratios(eqns1,pivotvar);
pivoteqn := eqns1[5];
eqns2 :=pivot(eqns1,pivotvar,pivoteqn);
obj2 :=subs(eqns2,obj1);
```

Voila! The optimum is reached in one step.

12.2 Problem 2

A company makes three lines of tires. Its four-ply biased tires produce \$6 in profit per tire, its fiberglass belted line \$4 per tire, and its radials \$8 a tire. Each type of tire passes through three manufacturing stages as a part of the entire manufacturing process. Each of the three process centers has the following hours of available production time per day: Molding: 12 hours, Curing: 9 hours, and Assembly: 16 hours. The time required in each process to produce 100 tires of each line is as follows: Four-ply tires require 2 hours molding, 3 hours curing and 2 hours assembly ; fiberglass belteds need 2 hours molding, 2 hours curing and 1 hour assembly; radials must have 2 hours molding, 1 hour curing, and 3 hours assembly.

a. Determine the optimum product mix for each day's production, assuming that all tires manufactured are sold.

b. If you had a little more time to give to just one of the processes, which would you choose? Explain your decision in a lot of gory detail.

c. If you were permitted to increase the profit just a little bit per tire of just one of the lines, which one would you choose? Again, give gory details.

12.3

12.3.1 Solution: By Maple without the simplex package.

```
restart;
profit := z = 6*x[1] +4*x[2] + 8*x[3];
cnsts := [2*x[1] +2*x[2] +2*x[3] <= 12,
3*x[1] +2*x[2] +x[3] <= 9, 2*x[1] +x[2] +3*x[3] <= 16];
eqns := [s[1] = 12 - 2*x[1] -2*x[2] -2*x[3],
s[2] = 9 - 3*x[1] -2*x[2] -x[3], s[3] = 16 - 2*x[1] - x[2]
-3*x[3]];
```

Making our first current feasable solution as $[x_1, x_2, x_3, s_1, s_2, s_3] = [\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{12}, \mathbf{9}, \mathbf{16}]$ with a profit of $\mathbf{0}$. And lots of room for improvement. Our first switch is x_3 and s_3.

```
pivotvar :=x[3];
pivoteqn := eqns[3];
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
eqns := [subs(neweqn, eqns[1]), subs(neweqn,
eqns[2]),neweqn];
profit := subs(neweqn, profit);
```

Leaving the current feasable solution as $[x_1, x_2, x_3, s_1, s_2, s_3] = [\mathbf{0}, \mathbf{0}, \frac{\mathbf{16}}{\mathbf{3}}, \mathbf{12}, \mathbf{9}, \mathbf{0}]$ with a profit of $\frac{\mathbf{128}}{\mathbf{3}}$. This can still be improved. Let's switch x_2 and s_1.

```
pivotvar :=x[2];
```

How did you decide on the pivot eqn? (Write out the ratios).

```
pivoteqn := eqns[1];
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
```

Opps! The neweqn should replace the 1st equation in the list, not the third. You forgot to change it when you copied it down.

```
eqns := [subs(neweqn, eqns[1]), subs(neweqn,
```

```
eqns[2]),neweqn];
eqns;
eqns := [neweqn,subs(neweqn, eqns[2]),
subs(neweqn, eqns[3]) ];
profit := subs(neweqn, profit);
```

Our current feasable solution is $[x_1, x_2, x_3, s_1, s_2, s_3] = [\mathbf{0}, \mathbf{1}, \mathbf{5}, \mathbf{0}, \mathbf{2}, \mathbf{0}]$ with a profit of 44 What are the units? And both of our remaining variables are negative. So this cannot be improved. (x_1 could be brought into the solution, but it would not increase the profit, it is merely another possible solution.)

But it is worth getting the other optimal corner point. Then any optimal solution will be a convex combination of this two points.

```
pivotvar :=x[1];
```

The ratios are 2,4/3, and 10. so the pivoteqn is

```
pivoteqn := eqns[2];
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
eqns :=
[subs(neweqn,eqns[1]),neweqn,subs(neweqn,eqns[3])];
profit := subs(neweqn,profit);
```

So another optimal solution is to make 4/3 batches of 4 ply tires, 1/3 batch of fiberglass tires and 13/3 batches of radials for a profit of 4400 dollars. Of course, this assumes one can make (and sell) thirds of tires, which isn't likely.

12.4

12.4.1 Part B. Sensitivity Analysis. (right hand sides)

```
restart;
profit := z = 6*x[1] +4*x[2] + 8*x[3];
cnsts := [2*x[1] +2*x[2] +2*x[3] <= 12+alpha,
3*x[1] +2*x[2] +x[3] <= 9+beta, 2*x[1] +x[2] +3*x[3] <= 16+gamma];
eqns := [s[1] = 12+alpha - 2*x[1] -2*x[2]
-2*x[3], s[2] = 9+beta - 3*x[1] -2*x[2] -x[3], s[3] = 16+gamma -
2*x[1] - x[2] -3*x[3]];
```

Making our first current feasible solution as $[x_1, x_2, x_3, s_1, s_2, s_3] = [\mathbf{0}, \mathbf{0}, \mathbf{0}, \mathbf{12}, \mathbf{9}, \mathbf{16}]$ with a profit of $\mathbf{0}$. And lots of room for improvement. Our first switch is x_3 and s_3.

```
pivotvar :=x[3];
pivoteqn := eqns[3];
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
eqns := [subs(neweqn, eqns[1]), subs(neweqn,
eqns[2]),neweqn];
profit := subs(neweqn, profit);
pivotvar :=x[2];
pivoteqn := eqns[1];
```

Leaving the current feasable solution as $[x_1,\, x_2,\, x_3,\, s_1,\, s_2,\, s_3] = [0,\, 0,\, \frac{16}{3},\, 12,\, 9,\, 0]$ with a profit of $\frac{128}{3}$. This can still be improved. Let's switch x_2 and s_1.

```
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
eqns := [ neweqn,subs(neweqn, eqns[2]),
subs(neweqn, eqns[3])];
profit := subs(neweqn, profit);
```

This shows that the change in both α and γ are in the final solution. However, the coefficient of γ is twice as large as coefficient of α. Making the change of γ or Assembly the best choice.

Good reasoning. All the gory details are here. We really should go ahead and make the switch of x1 and s2 to check the analysis at the other corner optimal point.

```
pivotvar :=x[1];
pivoteqn := eqns[2];
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
eqns :=
[subs(neweqn,eqns[1]),neweqn,subs(neweqn,eqns[3])];
profit := subs(neweqn,profit);
```

So the shadow prices remain unchanged and we would increase assembly time.

12.5

12.5.1 Part C. Sensitivity Analysis. (price coefficients)

```
restart;
profit := z = 6*x[1]+alpha*x[1] +4*x[2]
+beta*x[2] + 8*x[3]+gamma*x[3];
cnsts := [2*x[1] +2*x[2] +2*x[3] <= 12,
3*x[1] +2*x[2] +x[3] <= 9, 2*x[1] +x[2] +3*x[3] <= 16];
eqns := [s[1] = 12 -2*x[1] -2*x[2] -2*x[3],
s[2] = 9 -3*x[1] -2*x[2] -x[3], s[3] = 16 -2*x[1] -x[2] -3*x[3]];
```

Making our first current feasable solution as $[x_1,\, x_2,\, x_3,\, s_1,\, s_2,\, s_3] = [0,\, 0,\, 0,\, 12,\, 9,\, 16]$ with a profit of 0. And lots of room for improvement. Our first switch is x_3 and s_3.

```
pivotvar :=x[3];
pivoteqn := eqns[3];
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
eqns := [subs(neweqn, eqns[1]), subs(neweqn,
eqns[2]),neweqn];
profit := subs(neweqn, profit);
pivotvar :=x[2];
pivoteqn := eqns[1];
```

Leaving the current feasable solution as $[x_1,\, x_2,\, x_3,\, s_1,\, s_2,\, s_3] = [0,\, 0,\, \frac{16}{3},\, 12,\, 9,\, 0]$ with a profit of $\frac{128}{3}$. This can still be improved. Let's switch x_2 and s_1.

```
neweqn := pivotvar = solve(pivoteqn,
pivotvar);
```

```
eqns := [neweqn,subs(neweqn, eqns[2]),
subs(neweqn, eqns[3]) ];
profit := subs(neweqn, profit);
```

Assuming the final values of x_1, s_1, s_3 all being $\mathbf{0}$, that alters the final equation to $z = 44{+}\beta{+}\gamma\,5$ making a change in gamma the most beneficial change possible. So, it would be best to add to x_3 the radial tires.

Good analysis. One could do the same analysis at the other optimal corner point. Would you expect any change in your answer there?

12.6 Problem 3

a) Modify the definition of ’tabular’ in the worksheet mymax.mws so that it prints out a table that looks more like the tableau in our text.

b) The worksheet entitled pigfarm2.mws takes the pigfarmer problem and solves it ’by hand’, using the simplex algorithm, but without making use of the simplex package in Maple. You will notice as you read through the worksheet that each iteration of the simplex algorithm has a similar form. Automate some of this repetitive work by defining one or more words. Use the worksheet mymax.mws as a guide. Test out your words.

12.7

12.7.1 Updated tabular

```
restart; with(linalg);
with(simplex);
tabular := proc (f ,obj, v )
local t, A, m,i, n, x,ob,c,Z, vars,zrow;
t := convert(map(lhs-rhs = 0,f),list);
if nargs = 2 then vars := sort(convert(indets(t,name),list))
else vars := v fi;
ob:=subs([seq(h=0,h=vars)],obj);
zrow := linalg[genmatrix]([obj=0],vars,’flag’);
zrow[1,nops(vars)+1] := ob;
A := linalg[’genmatrix’](t,vars,’flag’);
m := linalg[’rowdim’](A); n := linalg[’coldim’](A);
A := stack(zrow,A);
A := stack([op(vars),Z],A);
end:
booktabular := proc (f ,obj, v )
local t, A, m,i, n, x,ob,c,Z, vars,zrow;
#t := convert(map(lhs-rhs = 0,f),list);
t := convert(f,list);
if nargs = 2 then vars :=
sort(convert(indets(t,name),list))
else vars := v fi;
ob:=subs([seq(h=0,h=vars)],obj);
zrow := linalg[genmatrix]([obj=0],vars,’flag’);
zrow[1,nops(vars)+1] := ob;
```

```
A := linalg['genmatrix'](t,vars,'flag');
m := linalg['rowdim'](A); n := linalg['coldim'](A);
A := stack(zrow,A);
A := stack([op(vars),Z],A);
end:
```

And using the example from mymax.mws

```
cnsts := [x1+2*x2<=4,x1+x2<=3];
eqns := setup(cnsts);
z := 2*x1+3*x2;
tabular(eqns,z);
booktabular(eqns,z, [x1,x2,_SL1, _SL2]);
```

12.8

12.8.1 Automated Pigfarmer

Here is a typical minimum cost diet linear programming problem.

Pig problem

```
restart;with(linalg);
```

Problem: A pig farmer has resources of corn, tankage, and hay to make up a pig diet which satisfies certain minimum daily requirements as to the basic nutrients of carbohydrates, proteins, and vitamins. The table showing the minimum amounts of each nutrient needed and the amount of each nutrient in a kilogram of resource and the cost of the resource is given below. Desired is a diet which satisfies the minimum daily requirement at minimum cost.

```
tab := matrix(5,5,
[`Nutrients`,`corn(Kg)`,`tankage(Kg)`,`hay(Kg)`,`mdr`,
`carbo`,90,20,40,200, `protein`,30,80, 60,180,
`vitamins`,10,20,60,150, `cost(.01$)`,35,30,25,``]) ;
```

Setup: Let c, t, and h denote the number of kg of corn, tankage, and hay per pig in the diet. Then the constraints are linear inequalities in these quanitities.

```
cnsts := {90*c + 20*t + 40*h >=200,
30*c + 80*t + 60*h >=180,
10*c + 20*t + 60*h >=150};
```

The cost is a linear function of these quantitities.

```
cost := 35*c + 30*t + 25*h;
```

We can use the simplex package to solve this problem.

```
with(simplex);
sol := minimize(cost,cnsts,NONNEGATIVE);
subs(sol,[cost,cnsts]); evalf(");
```

So, each pig should get no tankage, 17/7 kg of hay, and 8/7 kg of corn each day at at cost of about 1100/7 dollars. The pig gets his minimum carbo and protein, and about 7 units over his minimum vitamins.

```
eqns := [a1 = lhs(cnsts[1])-rhs(cnsts[1])+s1,
a2 = lhs(cnsts[2])-rhs(cnsts[2])+s2,
a3 = lhs(cnsts[3])-rhs(cnsts[3])+s3];
phase1cost := subs(eqns,a1 +a2 +a3);
```

```
tabular (eqns,phase1cost);
pivotvar := h;
pivoteqn := eqns[3];
pivoteqn := op(solve(pivoteqn
,{pivotvar})) ;
eqns2 :=
[subs(pivoteqn,eqns[1]),subs(pivoteqn,eqns[2]),pivoteqn];
findtab (eqns2, phase1cost);
pivotvar := c;
ratios := 100/(250/3),30/20, (5/2)/(1/6);
pivoteqn := eqns2[1];
pivoteqn := op(solve(pivoteqn
,{pivotvar}));
eqns3 :=
[pivoteqn,subs(pivoteqn,eqns2[2]),subs(pivoteqn,eqns2[3])];
findtab (eqns3, (subs(pivoteqn,
phase1cost)));
pivotvar := t;
ratios :=
6/5/(2/25),6/(292/5),23/10/(8/25);
pivoteqn := eqns3[2];
pivoteqn := op(solve(pivoteqn
,{pivotvar}));
eqns4 :=
[subs(pivoteqn,eqns3[1]),pivoteqn,subs(pivoteqn,eqns3[3])];
findtab (eqns4, (subs(pivoteqn,
phase1cost)));
```

End of Phase 1

```
eqns := subs({a1=0,a2=0,a3=0},eqns4);
findtab (eqns, cost);
pivotvar := s3;
ratios :=
87/73/(1./146),21/1460/(15/146.);
pivoteqn := eqns[2];
pivoteqn :=
op(solve(pivoteqn,{pivotvar}));
eqns2 := [subs(pivoteqn,eqns[1]),pivoteqn,
subs(pivoteqn,eqns[3])];
findtab (eqns2, (subs(pivoteqn, cost)));
```

End of phase 2. The minimum cost diet is 8/7 kg of corn, 17/7 kg hay, 0 tankage for a cost of 705/7 dollars.

```
evalf(");
```

12.8.2

12.8.3 Tabular

```
tabular := proc (f ,obj, v )
local t, A, m,i, n, x,ob,c,Z, vars,zrow;
t := convert(map(lhs-rhs = 0,f),list);
if nargs = 2 then vars := sort(convert(indets(t,name),list))
else vars := v fi;
ob:=subs([seq(h=0,h=vars)],obj);
zrow := linalg[genmatrix]([obj=0],vars,'flag');
zrow[1,nops(vars)+1] := ob;
A := linalg['genmatrix'](t,vars,'flag');
m := linalg['rowdim'](A);
n := linalg['coldim'](A);
A := stack(zrow,A);
A := stack([op(vars),Z],A);
end:
```

12.8.4

12.8.5 FindTab

```
findtab:= proc (eqns, z)
local p1cost;
p1cost := subs(eqns, z);
tabular(eqns,p1cost);
end:
```

13 Sensitivity Analysis

Sensitive with four argument takes the original eqns and obj z, the sequence of pivots returned by mymax and the choice of increasing the ith resource by theta. It returns the change in the basic variables and the change in the objective in terms of theta. If the fourth argument is omitted, it takes a modified objective function (say theta has been added to one of the prices.) and returns the change in the final tableau. In either case, what sensitive does is modify the original set of equations, by additing theta to one of the right hand sides or to the coefficient of one of the varialbles in the objective, and the perform the sequence of pivots performed by mymax to optimize the objective. The we can look at the final tableau and solve some inequalitites for theta to get bounds on how much the right hand sides (or coefficients) can change before the optimal solution changes. We need tableau, step and mymax from mymax.mws

```
sensitive := proc(eqns,z,vseq,i)
local n,m,teqns,tz,j,k ;
n := nops(vseq); m := nops(eqns);
tz := z;
if nargs=4 then
teqns := [seq(eqns[j],j=1..i-1),lhs(eqns[i])=rhs(eqns[i])+theta,
seq(eqns[j],j=i+1..m)];
else teqns := eqns fi;
for j from 1 to n do for k from 1 to m do
if lhs(teqns[k])=vseq[j][2]
then tz := subs(solve( teqns[k],{vseq[j][1]}),
tz);
teqns := pivot(teqns,vseq[j][1],teqns[k]);
break; fi; od; od;
[teqns,tz] ;
end:
```

Sample problem:

Maximize $z = 2\,x1 + 3\,x2 + x3$ subject to $x1 + 4\,x2 + 2\,x3 \le 9,\ 3\,x1 + 2\,x2 + x3 \le 6.$ Then find the shadow prices and ranges for the right hand sides. Also determine the ranges for the coefficients of the objective function.

```
z := 2*x1+3*x2+x3;
```

$$z := 2\,x1 + 3\,x2 + x3$$

```
cnsts :=
[x1+4*x2+2*x3<=9,3*x1+2*x2+x3<=6];
```

$$cnsts := [x1 + 4\,x2 + 2\,x3 \le 9,\ 3\,x1 + 2\,x2 + x3 \le 6]$$

```
eqns := setup(cnsts) ;
```

$$eqns := [\_SL1 = 9 - x1 - 4\,x2 - 2\,x3,\ \_SL2 = 6 - 3\,x1 - 2\,x2 - x3]$$

```
sln :=mymax(eqns,z);
```

$$\left[\begin{array}{cccccc} \_SL1 & \_SL2 & x1 & x2 & x3 & Z \\ \frac{-3}{4} & 0 & \frac{5}{4} & 0 & \frac{-1}{2} & \frac{27}{4} \\ \frac{1}{4} & 0 & \frac{1}{4} & 1 & \frac{1}{2} & \frac{9}{4} \\ \frac{-1}{2} & 1 & \frac{5}{2} & 0 & 0 & \frac{3}{2} \end{array}\right]$$

$$\left[\begin{array}{cccccc} \mathit{\_SL1} & \mathit{\_SL2} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\ \frac{-1}{2} & \frac{-1}{2} & 0 & 0 & \frac{-1}{2} & \frac{15}{2} \\ \frac{3}{10} & \frac{-1}{10} & 0 & 1 & \frac{1}{2} & \frac{21}{10} \\ \frac{-1}{5} & \frac{2}{5} & 1 & 0 & 0 & \frac{3}{5} \end{array}\right]$$

$$\mathit{finished: \; no \; pivot \; variable.}$$

$$\mathit{sln} := [[\mathit{x2} = -\frac{3}{10}\mathit{\_SL1} + \frac{21}{10} + \frac{1}{10}\mathit{\_SL2} - \frac{1}{2}\mathit{x3},\ \mathit{x1} = -\frac{2}{5}\mathit{\_SL2} + \frac{3}{5} + \frac{1}{5}\mathit{\_SL1}], \\ -\frac{1}{2}\mathit{\_SL2} + \frac{15}{2} - \frac{1}{2}\mathit{\_SL1} - \frac{1}{2}\mathit{x3},\ [[\mathit{x2}, \mathit{\_SL1}], [\mathit{x1}, \mathit{\_SL2}]]]$$

We had previously written mymax to return not only the final set of equations and the final form of the objective, but also the sequence of pivots that were made to achieve the optimum. So we can tell that first x2 was pivoted with SL1 and then x1 was pivoted with SL1. That is just what senstive needs. So to determine the effect of adding theta to the right side of the first constraint, we make the call

```
sol:= sensitive(eqns,z,sln[3],1) ;
tabular(op(sol),[x1,x2,x3,_SL1,_SL2] );
```

$$\mathit{sol} := [[\mathit{x2} = -\frac{3}{10}\mathit{\_SL1} + \frac{21}{10} + \frac{1}{10}\mathit{\_SL2} + \frac{3}{10}\theta - \frac{1}{2}\mathit{x3},\ \mathit{x1} = -\frac{2}{5}\mathit{\_SL2} + \frac{3}{5} + \frac{1}{5}\mathit{\_SL1} - \frac{1}{5}\theta], \\ -\frac{1}{2}\mathit{\_SL2} + \frac{15}{2} - \frac{1}{2}\mathit{\_SL1} + \frac{1}{2}\theta - \frac{1}{2}\mathit{x3}]$$

$$\left[\begin{array}{cccccc} \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{\_SL1} & \mathit{\_SL2} & Z \\ 0 & 0 & \frac{-1}{2} & \frac{-1}{2} & \frac{-1}{2} & \frac{15}{2} + \frac{1}{2}\theta \\ 0 & 1 & \frac{1}{2} & \frac{3}{10} & \frac{-1}{10} & \frac{21}{10} + \frac{3}{10}\theta \\ 1 & 0 & 0 & \frac{-1}{5} & \frac{2}{5} & -\frac{1}{5}\theta + \frac{3}{5} \end{array}\right]$$

So right away, we can see the effect of increasing the righthand side of constraint 1 by a small amount θ is to increase z by $.5\,\theta$. Thus the shadow price of resource 1 is $\frac{1}{2}$.

To get the range for the right hand side, we can solve the two inequalities $0 \le -\frac{1\theta}{5} + \frac{3}{5}$ and $0 \le \frac{21}{10} + \frac{3\theta}{10}$ for θ to get that it must stay between -7 and 3 in order that the current basic solution remain optimum. This means the resource 1 (currently 9) can range from 2 to 12 without changing the basic optimum solution.

```
sol:= sensitive(eqns,z,sln[3],2) ;
tabular(op(sol),[x1,x2,x3,_SL1,_SL2] );
```

$$\mathit{sol} := [[\mathit{x2} = -\frac{3}{10}\mathit{\_SL1} + \frac{21}{10} + \frac{1}{10}\mathit{\_SL2} - \frac{1}{10}\theta - \frac{1}{2}\mathit{x3},\ \mathit{x1} = -\frac{2}{5}\mathit{\_SL2} + \frac{3}{5} + \frac{1}{5}\mathit{\_SL1} + \frac{2}{5}\theta], \\ -\frac{1}{2}\mathit{\_SL2} + \frac{15}{2} - \frac{1}{2}\mathit{\_SL1} + \frac{1}{2}\theta - \frac{1}{2}\mathit{x3}]$$

$$\left[\begin{array}{cccccc} \mathit{x1} & \mathit{x2} & \mathit{x3} & \mathit{\_SL1} & \mathit{\_SL2} & Z \\ 0 & 0 & \frac{-1}{2} & \frac{-1}{2} & \frac{-1}{2} & \frac{15}{2} + \frac{1}{2}\theta \\ 0 & 1 & \frac{1}{2} & \frac{3}{10} & \frac{-1}{10} & \frac{21}{10} - \frac{1}{10}\theta \\ 1 & 0 & 0 & \frac{-1}{5} & \frac{2}{5} & \frac{2}{5}\theta + \frac{3}{5} \end{array}\right]$$

Exercise: Interpret the above call to sensitive.

Now let's check the sensitivity to a change in the price of x1. Here all we do is add $\theta\, x1$ to the objective function and perform the sequence of pivots.

```
collect(z+theta*x1,x1);
```

$$(2+\theta)\,x1 + 3\,x2 + x3$$

```
sol:= sensitive(eqns,z+theta*x1,sln[3]);
tabular(op(sol),[x1,x2,x3,_SL1,_SL2] );
```

$$sol := [[x2 = -\frac{3}{10}\,\_SL1 + \frac{21}{10} + \frac{1}{10}\,\_SL2 - \frac{1}{2}\,x3,\ x1 = -\frac{2}{5}\,\_SL2 + \frac{3}{5} + \frac{1}{5}\,\_SL1],$$
$$-\frac{1}{2}\,\_SL2 + \frac{15}{2} - \frac{1}{2}\,\_SL1 - \frac{1}{2}\,x3 + \theta\,(-\frac{2}{5}\,\_SL2 + \frac{3}{5} + \frac{1}{5}\,\_SL1)]$$

$$\left[\begin{array}{cccccc} x1 & x2 & x3 & \_SL1 & \_SL2 & Z \\ 0 & 0 & \frac{-1}{2} & -\frac{1}{2}+\frac{1}{5}\,\theta & -\frac{1}{2}-\frac{2}{5}\,\theta & \frac{15}{2}+\frac{3}{5}\,\theta \\ 0 & 1 & \frac{1}{2} & \frac{3}{10} & \frac{-1}{10} & \frac{21}{10} \\ 1 & 0 & 0 & \frac{-1}{5} & \frac{2}{5} & \frac{3}{5} \end{array}\right]$$

Now by inspection we can see that increasing the price of x1 by a small amount theta has the effect of increasing the profit by 3/5 theta. Further more, the ranges on x1 for which the x1 and x2 are the basic variables can be obtained by solving the inequalities $-\frac{1}{2}+\frac{1\,\theta}{5}\le \mathbf{0}$ and $-\frac{1}{2}-\frac{2\,\theta}{5}\le \mathbf{0}$ for theta to get it between -1.25 and 2.5. So the coefficient of x1 (currently 2) can range from .75 to 4.5 without changing the optimal point.

```
sol:= sensitive(eqns,z+theta*x2,sln[3] ) ;
tabular(op(sol),[x1,x2,x3,_SL1,_SL2],4);
```

$$sol := [[x2 = -\frac{3}{10}\,\_SL1 + \frac{21}{10} + \frac{1}{10}\,\_SL2 - \frac{1}{2}\,x3,\ x1 = -\frac{2}{5}\,\_SL2 + \frac{3}{5} + \frac{1}{5}\,\_SL1],$$
$$-\frac{1}{2}\,\_SL2 + \frac{15}{2} - \frac{1}{2}\,\_SL1 - \frac{1}{2}\,x3 + \theta\,(-\frac{3}{10}\,\_SL1 + \frac{21}{10} + \frac{1}{10}\,\_SL2 - \frac{1}{2}\,x3)]$$

$$\left[\begin{array}{cccccc} x1 & x2 & x3 & \_SL1 & \_SL2 & Z \\ 0 & 0 & -\frac{1}{2}-\frac{1}{2}\,\theta & -\frac{1}{2}-\frac{3}{10}\,\theta & -\frac{1}{2}+\frac{1}{10}\,\theta & \frac{15}{2}+\frac{21}{10}\,\theta \\ 0 & 1 & \frac{1}{2} & \frac{3}{10} & \frac{-1}{10} & \frac{21}{10} \\ 1 & 0 & 0 & \frac{-1}{5} & \frac{2}{5} & \frac{3}{5} \end{array}\right]$$

```
sol:= sensitive(eqns,z+theta*x3,sln[3] ) ;
tabular(op(sol),[x1,x2,x3,_SL1,_SL2],4);
```

$$sol := [[x2 = -\frac{3}{10}\,\_SL1 + \frac{21}{10} + \frac{1}{10}\,\_SL2 - \frac{1}{2}\,x3,\ x1 = -\frac{2}{5}\,\_SL2 + \frac{3}{5} + \frac{1}{5}\,\_SL1],$$
$$-\frac{1}{2}\,\_SL2 + \frac{15}{2} - \frac{1}{2}\,\_SL1 - \frac{1}{2}\,x3 + \theta\,x3]$$

$$\left[\begin{array}{cccccc} x1 & x2 & x3 & \_SL1 & \_SL2 & Z \\ 0 & 0 & \theta-\frac{1}{2} & \frac{-1}{2} & \frac{-1}{2} & \frac{15}{2} \\ 0 & 1 & \frac{1}{2} & \frac{3}{10} & \frac{-1}{10} & \frac{21}{10} \\ 1 & 0 & 0 & \frac{-1}{5} & \frac{2}{5} & \frac{3}{5} \end{array}\right]$$

Exercise: Interpret these last two calls to sensitive and then compare all of these calculations with the numbers that mathprog gives you.

14 The revised simplex method

We want to redo the simplex method using matrix multiplication. The main reason for doing this (aside from the fact that matrices are nice) is to be able to do sensitivity analysis in a computationally efficient manner. Our current method involves keeping a record of the pivots that we performed in getting to an optimal corner point solution, and then modifying either a constraint restriction or a price coefficient in the original set of equations, performing the same sequence of pivots, and solving the resulting inequalities we got by looking at the final set of equations to get the ranges. We will see that using matrices greatly reduces the amount of arithmetic. So lets look at a typical problem.

Maximize $2\,x+y$ subject to $x+y\leq 10$ and $3\,x+y\leq 15$.

```
with(linalg):
```

```
Warning, new definition for norm

Warning, new definition for trace
```

First, construct the matrix of the initial tableau.

```
T := matrix(3,6,[1,-2,-1,0,0,0,
0,1,1,1,0,10,
0,3,1,0,1,15]);
```

$$T:=\begin{bmatrix}1&-2&-1&0&0&0\\0&1&1&1&0&10\\0&3&1&0&1&15\end{bmatrix}$$

We choose the third column as the pivot column and the must choose (because of the ratios) the second row as the pivot row. So now construct the elementary gauss matrix to perform the necessary row operations on T.

```
id := diag(1,1,1);
```

$$id:=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}$$

```
multipliers := [1,1,-1];
```

$$multipliers:=[1,\,1,\,-1]$$

```
e1 := augment(
col(id,1),multipliers,col(id,3));
```

$$e1:=\begin{bmatrix}1&1&0\\0&1&0\\0&-1&1\end{bmatrix}$$

To get the tableau after one iteration, multiply e1 by T.

```
T1 := evalm(e1&*T);
```

$$T1:=\begin{bmatrix}1&-1&0&1&0&10\\0&1&1&1&0&10\\0&2&0&-1&1&5\end{bmatrix}$$

There is still a negative coefficient in the top row. Choose the second column and (by necessity) the third row as the pivot row and column. The multipliers are

```
multipliers := [1/2,-1/2,1/2];
```

$$multipliers:=[\frac{1}{2},\,\frac{-1}{2},\,\frac{1}{2}]$$

and the gaussian matrix is

```
e2 := augment(col(id,1..2),multipliers);
```

$$e2 := \begin{bmatrix} 1 & 0 & \frac{1}{2} \\ 0 & 1 & \frac{-1}{2} \\ 0 & 0 & \frac{1}{2} \end{bmatrix}$$

and the new (and final) tableau is

```
T2 := evalm(e2&*T1);
```

$$T2 := \begin{bmatrix} 1 & 0 & 0 & \frac{1}{2} & \frac{1}{2} & \frac{25}{2} \\ 0 & 0 & 1 & \frac{3}{2} & \frac{-1}{2} & \frac{15}{2} \\ 0 & 1 & 0 & \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} \end{bmatrix}$$

The product e2*e1 contains a complete record of the pivots that were made in the simplex algorithm. Note it actually appears in the final tableau– it is the 1st, 4th, and 5th columns of T2.

```
e21 := evalm(e2&*e1);
```

$$e21 := \begin{bmatrix} 1 & \frac{1}{2} & \frac{1}{2} \\ 0 & \frac{3}{2} & \frac{-1}{2} \\ 0 & \frac{-1}{2} & \frac{1}{2} \end{bmatrix}$$

To do sensitivity analysis on the first resouce, all we need to do is multiply e21 by the column [0,10+theta,15] and solve the resulting inequalities for theta.

```
sense1 :=evalm(e21&*( [0,10+theta,15])
);
```

$$sense1 := \left[\frac{25}{2} + \frac{1}{2}\,\theta,\ \frac{15}{2} + \frac{3}{2}\,\theta,\ \frac{5}{2} - \frac{1}{2}\,\theta\right]$$

```
solve({sense1[3]>=0,sense1[2]>=0},theta);
```

So, the right hand side of the first resource can vary between 5 and 15 before the basic variables change in the optimal solution. Notice that to perform this sensitivity analysis, we only need to multiply e21 times the right most column of the original tableau, modified by adding theta to the 2nd entry. This provides a computational advantage over the previous method of performing sensitivity analysis, which involved stepping through the iterations again.

$$\{\theta \leq 5,\ -5 \leq \theta\}$$

Sensitivity analysis on the second resource is done in the same way and is shown below.

Now, sensitivity analysis on the price coefficients is a tiny bit more involved. In order to see what to do, let's add epsilon to the first price coefficient in T and multiply by e21.

```
T[1,2]:= -(2+epsilon);
```

$$T_{1,\,2} := -2 - \varepsilon$$

```
price1_sensitivity := evalm(e21&*T);
```

$$price1\_sensitivity := \begin{bmatrix} 1 & -\varepsilon & 0 & \frac{1}{2} & \frac{1}{2} & \frac{25}{2} \\ 0 & 0 & 1 & \frac{3}{2} & \frac{-1}{2} & \frac{15}{2} \\ 0 & 1 & 0 & \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} \end{bmatrix}$$

The only thing that happens is that an epsilon appears in the first row second column of the tableau. In order to bring x back into solution, we must multiply the bottom row by epsilon and add it to the top row. This row operation can be performed by multiplying on the right by an elementary upper triangular matrix e3.

```
e3 := augment(col(id,1..2),[epsilon,0,1]);
```

$$e3 := \begin{bmatrix} 1 & 0 & \varepsilon \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
p1s := evalm(e3&*price1_sensitivity);
```

$$p1s := \begin{bmatrix} 1 & 0 & 0 & \frac{1}{2} - \frac{1}{2}\varepsilon & \frac{1}{2} + \frac{1}{2}\varepsilon & \frac{25}{2} + \frac{5}{2}\varepsilon \\ 0 & 0 & 1 & \frac{3}{2} & \frac{-1}{2} & \frac{15}{2} \\ 0 & 1 & 0 & \frac{-1}{2} & \frac{1}{2} & \frac{5}{2} \end{bmatrix}$$

Now we see what to do. The shadow prices want to stay nonnegative , else we could increase profit by changing basic variables. So there are two inequalities to solve for epsilon.

```
solve({p1s[1,4]>=0,p1s[1,5]>=0},epsilon);
```

$$\{\varepsilon \leq 1,\ -1 \leq \varepsilon\}$$

This says that the coefficient of x can range all the way from 1 to 3 before the basic variables must change to maintain optimality.

Problem: Carry out the sensitivity analysis on the second price coefficient (the coefficient of y in the objective).

14.1

```
sense2 :=evalm(e21&*( [0,10,15+theta])
);
```

$$sense2 := \left[\frac{25}{2} + \frac{1}{2}\theta,\ \frac{15}{2} - \frac{1}{2}\theta,\ \frac{5}{2} + \frac{1}{2}\theta\right]$$

```
solve({sense1[3]>=0,sense1[2]>=0},theta);
```

$$\{\theta \leq 5,\ -5 \leq \theta\}$$

15 The revised simplex method (automated)

We want to automate the revised simplex method. This involves defining a word 'getpiv' which returns the pivot row and pivot column (ie the pivotequation and pivot variable) and then defining a word 'gauss' which constructs the gauss matrix to perform the exchange. Then, a word 'matmax' can be defined which is similar to 'mymax' from previous worksheets. It performs the simplex method until the objective is maximized, printing the intermediate tableaux as it goes.

```
with(linalg):
```

```
Warning, new definition for norm

Warning, new definition for trace
```

The word gauss defined below takes a column cl and a row number r and returns the gauss matrix which when multiplied times the cl yields the basis column with a 1 in the rth place and 0's elsewhere.

```
gauss := proc(cl,r)
local id,n,multipliers;
n := vectdim(cl);
id := diag(1$n);
multipliers := evalm(-1/cl[r]*cl);
multipliers[r] := 1/cl[r];
augment(col(id,1..r-1),multipliers,col(id,r+1..n));
end;
```

$$\begin{array}{l} gauss := \mathrm{proc}(cl,\ r)\\ \quad \mathrm{local}\,id,\ n,\ multipliers;\\ \qquad n := \mathrm{vectdim}(cl)\ ;\\ \qquad id := \mathrm{diag}(1\ \$\ n)\ ;\\ \qquad multipliers := \mathrm{evalm}(-cl/cl_r)\ ;\\ \qquad multipliers_r := 1/cl_r\ ;\\ \qquad \mathrm{augment}(\mathrm{col}(id,\ 1..r-1),\ multipliers,\ \mathrm{col}(id,\ r+1..n))\\ \quad \mathrm{end} \end{array}$$

Now getpiv takes a tableau T and a list of the column numbers of the nonbasic columns of T, and returns a suitable pivot row and column if one exist, else it returns FAIL.

```
getpiv := proc(T,nbv)
local m, i, tmp, j;
m := coldim(T);
for i in nbv do
tmp := 0,infinity;
if T[1,i]<0 then
for j from 2 to rowdim(T) do
if T[j,i]>0 and T[j,m]/T[j,i]<tmp[2]
then tmp := j,T[j,m]/T[j,i] fi;
od;
fi;
if tmp[1] > 0 then RETURN([tmp[1],i]) fi ;
od;
FAIL; end:
```

Matmax takes a tableau and a vector containing the list of column numbers of the basis for the columns of the tableau (less the right hand column.)

```
matmax := proc( tableau , basic )
local pivot, T, nbv, n, m, b;
```

```
T := evalm(tableau);
n := coldim(T);m := rowdim(T):
b := evalm(basic);
print(evalm(T));
nbv := {seq(i,i=2..n-1)}minus{seq(b[i],i=1..m)};
pivot := getpiv(T,nbv);
while not pivot=FAIL do
b[pivot[1]]:= pivot[2];
T := evalm(gauss(col(T,pivot[2]),pivot[1])&*T);
print(evalm(T));
nbv := {seq(i,i=2..n-1)}minus{seq(b[i],i=1..m)};
pivot := getpiv(T,nbv);
od;
[evalm(T),evalm(b)];
end;
```

```
matmax := proc(tableau, basic)
    localpivot, T, nbv, n, m, b;
        T := evalm(tableau) ;
        n := coldim(T) ;
        m := rowdim(T) ;
        b := evalm(basic) ;
        print(evalm(T)) ;
        nbv := {seq(i, i = 2..n − 1)} \ {seq(b_i, i = 1..m)} ;
        pivot := getpiv(T, nbv) ;
        while not (pivot = FAIL) do
            b_{pivot_1} := pivot_2 ;
            T := evalm((gauss(col(T, pivot_2), pivot_1)) &* T) ;
            print(evalm(T)) ;
            nbv := {seq(i, i = 2..n − 1)} \ {seq(b_i, i = 1..m)} ;
            pivot := getpiv(T, nbv)
        od;
        [evalm(T), evalm(b)]
    end
```

Test it out on this problem: maximize x1 - x2 +2x3 subject to the constraints 2x1- 2x2 + 3x3 <= 5, x1+x2-x3 <=3, and x1-x2+x3 <= 2.

Solution: initialize the tableau.

```
T := matrix(4,8,[1,-1,1,-2,0,0,0,0,
0,2,-2,3,1,0,0,5,
0,1,1,-1,0,1,0,3,
0,1,-1,1,0,0,1,2]);
```

$$T := \begin{bmatrix} 1 & -1 & 1 & -2 & 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 3 & 1 & 0 & 0 & 5 \\ 0 & 1 & 1 & -1 & 0 & 1 & 0 & 3 \\ 0 & 1 & -1 & 1 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Say the word matmax.

```
matmax(T,[1,5,6,7]);
```

$$\begin{bmatrix} 1 & -1 & 1 & -2 & 0 & 0 & 0 & 0 \\ 0 & 2 & -2 & 3 & 1 & 0 & 0 & 5 \\ 0 & 1 & 1 & -1 & 0 & 1 & 0 & 3 \\ 0 & 1 & -1 & 1 & 0 & 0 & 1 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 & 1 & 0 & -2 & 1 \\ 0 & 0 & 2 & -2 & 0 & 1 & -1 & 1 \\ 0 & 1 & -1 & 1 & 0 & 0 & 1 & 2 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & -1 & 3 \\ 0 & 0 & 0 & 1 & 1 & 0 & -2 & 1 \\ 0 & 0 & 2 & 0 & 2 & 1 & -5 & 3 \\ 0 & 1 & -1 & 0 & -1 & 0 & 3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & \frac{1}{3} & \frac{-1}{3} & 0 & \frac{2}{3} & 0 & 0 & \frac{10}{3} \\ 0 & \frac{2}{3} & \frac{-2}{3} & 1 & \frac{1}{3} & 0 & 0 & \frac{5}{3} \\ 0 & \frac{5}{3} & \frac{1}{3} & 0 & \frac{1}{3} & 1 & 0 & \frac{14}{3} \\ 0 & \frac{1}{3} & \frac{-1}{3} & 0 & \frac{-1}{3} & 0 & 1 & \frac{1}{3} \end{bmatrix}$$

$$\begin{bmatrix} 1 & 2 & 0 & 0 & 1 & 1 & 0 & 8 \\ 0 & 4 & 0 & 1 & 1 & 2 & 0 & 11 \\ 0 & 5 & 1 & 0 & 1 & 3 & 0 & 14 \\ 0 & 2 & 0 & 0 & 0 & 1 & 1 & 5 \end{bmatrix}$$

$$\left[\begin{bmatrix} 1 & 2 & 0 & 0 & 1 & 1 & 0 & 8 \\ 0 & 4 & 0 & 1 & 1 & 2 & 0 & 11 \\ 0 & 5 & 1 & 0 & 1 & 3 & 0 & 14 \\ 0 & 2 & 0 & 0 & 0 & 1 & 1 & 5 \end{bmatrix}, [1, 4, 3, 7]\right]$$

So there were three iterations. The objective is maximized at 8 when x1 = 11 and x2 = 14 , s3 = 5.

16 More on Big M and 2 phase

Problem Minimize $2\,x1 + x2 + 3\,x3$ subject to $420 = 5\,x1 + 2\,x2 + 7\,x3$ and $280 \leq 3\,x1 + 2\,x2 + 5\,x3$

```
with(linalg); with(simplex);
```

First set up with 2 artificial and 1 surplus variables.

```
eqns := [a1 = 420-5*x1-2*x2-7*x3,
a2 = 280 -3*x1-2*x2-5*x3 +s1 ];
```

$$eqns := [a1 = 420 - 5\,x1 - 2\,x2 - 7\,x3,\ a2 = 280 - 3\,x1 - 2\,x2 - 5\,x3 + s1]$$

The big M method

Here the objective is modified by adding an indeterminately large multiple of each artificial variable. This will force these variables out of solution if the original problem has a solution. (The original set of constraints could be inconsistent).

```
obj := 2*x1 + x2 + 3*x3 + M*a1 + M*a2;
```

$$obj := 2\,x1 + x2 + 3\,x3 + M\,a1 + a2\,M$$

We can use tabular from mymax to display the equations for making pivoting decisions.

```
tabular(eqns,obj);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ M & M & 0 & 2 & 1 & 3 & 0 \\ 1 & 0 & 0 & 5 & 2 & 7 & 420 \\ 0 & 1 & -1 & 3 & 2 & 5 & 280 \end{bmatrix}$$

Actually, we need to eliminate the basic variables from the objective so that we can decide what the possible pivot variables (entering basic variables) are.

```
obj :=
collect(simplify(subs(eqns,obj)),[x1,x2,x3,s1]);
```

$$obj := (2 - 8\,M)\,x1 + (-4\,M + 1)\,x2 + (-12\,M + 3)\,x3 + 700\,M + M\,s1$$

```
tabular(eqns,obj);
```

$$\begin{bmatrix} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ 0 & 0 & M & 2 - 8\,M & -4\,M + 1 & -12\,M + 3 & 700\,M \\ 1 & 0 & 0 & 5 & 2 & 7 & 420 \\ 0 & 1 & -1 & 3 & 2 & 5 & 280 \end{bmatrix}$$

Now we could bring any of x1, x2 , or x3 into solution. We choose x3. Look at ratios to get leaving basic variable.

```
ratios := 420/7.,280/5.;
```

$$ratio := 60.00000000,\ 56.00000000$$

This says i t's close but we better switch x3 and a2.

```
eqns2 := pivot(eqns,x3,eqns[2]);
```

$$eqns2 := [a1 = 28 - \frac{4}{5}\,x1 + \frac{4}{5}\,x2 + \frac{7}{5}\,a2 - \frac{7}{5}\,s1,\ x3 = -\frac{1}{5}\,a2 + 56 - \frac{3}{5}\,x1 - \frac{2}{5}\,x2 + \frac{1}{5}\,s1]$$

Again, eliminate the basic variables from the objective.

```
obj2 :=
collect(simplify(subs(eqns2,obj)),[x1,x2,a2,s1]);
```

$$obj2 := (\frac{1}{5} - \frac{4}{5}\,M)\,x1 + (\frac{4}{5}\,M - \frac{1}{5})\,x2 + (-\frac{3}{5} + \frac{12}{5}\,M)\,a2 + (-\frac{7}{5}\,M + \frac{3}{5})\,s1 + 168 + 28\,M$$

Make a table to facilitate decisions on the next iteration.

```
tabular(eqns2,obj2);
```

$$\left[\begin{array}{ccccccc} \mathit{a1} & \mathit{a2} & \mathit{s1} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\ 0 & -\frac{3}{5}+\frac{12}{5}M & -\frac{7}{5}M+\frac{3}{5} & \frac{1}{5}-\frac{4}{5}M & \frac{4}{5}M-\frac{1}{5} & 0 & 168+28M \\ 1 & \frac{-7}{5} & \frac{7}{5} & \frac{4}{5} & \frac{-4}{5} & 0 & 28 \\ 0 & \frac{1}{5} & \frac{-1}{5} & \frac{3}{5} & \frac{2}{5} & 1 & 56 \end{array}\right]$$

s1 or x1 could be the entering basic variable, choose x1. Then calculate ratio to get the leaving basic variable.

```
ratios := 28/(4/5), 56/(3/5.);
```

$$\mathit{ratio} := 35,\ 93.33333333$$

This says to pivot on a1, the first equation.

```
eqns3 := pivot(eqns2,x1,eqns2[1]);
```

$$\mathit{eqns3} := [\mathit{x1} = -\frac{5}{4}\,\mathit{a1} + 35 + \mathit{x2} + \frac{7}{4}\,\mathit{a2} - \frac{7}{4}\,\mathit{s1},\ \mathit{x3} = -\frac{5}{4}\,\mathit{a2} + 35 + \frac{3}{4}\,\mathit{a1} - \mathit{x2} + \frac{5}{4}\,\mathit{s1}]$$

```
obj3 := collect(simplify(subs(eqns3,obj2)),[
x2,a1,a2,s1]);
```

$$\mathit{obj3} := (-\frac{1}{4}+M)\,\mathit{a1} + (-\frac{1}{4}+M)\,\mathit{a2} + 175 + \frac{1}{4}\,\mathit{s1}$$

```
tabular(eqns3,obj3);
```

$$\left[\begin{array}{ccccccc} \mathit{a1} & \mathit{a2} & \mathit{s1} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\ -\frac{1}{4}+M & -\frac{1}{4}+M & \frac{1}{4} & 0 & 0 & 0 & 175 \\ \frac{5}{4} & \frac{-7}{4} & \frac{7}{4} & 1 & -1 & 0 & 35 \\ \frac{-3}{4} & \frac{5}{4} & \frac{-5}{4} & 0 & 1 & 1 & 35 \end{array}\right]$$

We are done. All coefficients of nonbasic variables in the objective row are positive, so increasing any of those would increase Z.

Phase 1 - Phase 2 method

Here, the idea is to first minimize the sum of the artificial variables as the first phase. If that minimizes to 0 then both a1 and a2 are zero and we will have a feasible solution to the original problem.

Start of Phase 1

```
aobj := a1 +a2;
```

$$\mathit{aobj} := \mathit{a1} + \mathit{a2}$$

```
aobj := subs(eqns,aobj);
```

$$\mathit{aobj} := 700 - 8\,\mathit{x1} - 4\,\mathit{x2} - 12\,\mathit{x3} + \mathit{s1}$$

```
eqns;
```

$$[\mathit{a1} = 420 - 5\,\mathit{x1} - 2\,\mathit{x2} - 7\,\mathit{x3},\ \mathit{a2} = 280 - 3\,\mathit{x1} - 2\,\mathit{x2} - 5\,\mathit{x3} + \mathit{s1}]$$

```
tabular(eqns,aobj);
```

$$\left[\begin{array}{ccccccc} \mathit{a1} & \mathit{a2} & \mathit{s1} & \mathit{x1} & \mathit{x2} & \mathit{x3} & Z \\ 0 & 0 & 1 & -8 & -4 & -12 & 700 \\ 1 & 0 & 0 & 5 & 2 & 7 & 420 \\ 0 & 1 & -1 & 3 & 2 & 5 & 280 \end{array}\right]$$

So here we could pivot on any of x1, x2, or x3, since increasing any of these would decrease a1+a2. We choose x1.

```
ratios := 420/5.,280/3.;
```

$$\mathit{ratio} := 84.00000000,\ 93.33333333$$

So we want to pivot x1 and a1

```
eqns2 := pivot(eqns,x1,eqns[1]);
```

$$eqns2 := [x1 = -\frac{1}{5}\,a1 + 84 - \frac{2}{5}\,x2 - \frac{7}{5}\,x3,\ a2 = 28 + \frac{3}{5}\,a1 - \frac{4}{5}\,x2 - \frac{4}{5}\,x3 + s1]$$

```
aobj2 := subs(eqns2,aobj);
```

$$aobj2 := 28 + \frac{8}{5}\,a1 - \frac{4}{5}\,x2 - \frac{4}{5}\,x3 + s1$$

```
tabular(eqns2,aobj2);
```

$$\left[\begin{array}{ccccccc} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ \frac{8}{5} & 0 & 1 & 0 & \frac{-4}{5} & \frac{-4}{5} & 28 \\ \frac{1}{5} & 0 & 0 & 1 & \frac{2}{5} & \frac{7}{5} & 84 \\ \frac{-3}{5} & 1 & -1 & 0 & \frac{4}{5} & \frac{4}{5} & 28 \end{array}\right]$$

Now we could pivot on x2 or x3. We choose x2.

```
ratios := 84/(2/5.),28/(4/5.);
```

$$ratio := 210.0000000,\ 35.00000000$$

Pivot on the second equation (a2).

```
eqns3 := pivot(eqns2,x2,eqns2[2]);
```

$$eqns3 := [x1 = -\frac{1}{2}\,a1 + 70 + \frac{1}{2}\,a2 - x3 - \frac{1}{2}\,s1,\ x2 = -\frac{5}{4}\,a2 + 35 + \frac{3}{4}\,a1 - x3 + \frac{5}{4}\,s1]$$

```
aobj3 := subs(eqns3,aobj2);
```

$$aobj3 := a1 + a2$$

```
tabular(eqns3,aobj3);
```

$$\left[\begin{array}{ccccccc} a1 & a2 & s1 & x1 & x2 & x3 & Z \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{-1}{2} & \frac{1}{2} & 1 & 0 & 1 & 70 \\ \frac{-3}{4} & \frac{5}{4} & \frac{-5}{4} & 0 & 1 & 1 & 35 \end{array}\right]$$

We have maximized aobj to 0. This is the end of phase 1. We have a feasible solution to the original problem. x1=70, x2=35. aobj= 0 So set the objective back to the original, restate it in terms of the current nonbasic variables and remove a1 and a2 from the set of nonbasic variables (i.e., set them to 0 in eqns and forget 'em).

```
eqns:=subs({a1=0,a2=0},eqns3);
```

$$eqns := [x1 = 70 - x3 - \frac{1}{2}\,s1,\ x2 = 35 - x3 + \frac{5}{4}\,s1]$$

```
obj := subs(eqns ,2*x1 + x2 + 3*x3) ;
```

$$obj := 175 + \frac{1}{4}\,s1$$

```
tabular(eqns,obj);
```

$$\left[\begin{array}{ccccc} s1 & x1 & x2 & x3 & Z \\ \frac{1}{4} & 0 & 0 & 0 & 175 \\ \frac{1}{2} & 1 & 0 & 1 & 70 \\ \frac{-5}{4} & 0 & 1 & 1 & 35 \end{array}\right]$$

Why are we done? Because increasing s1 results in an increase of obj and increasing x3 results in no change in obj. (Of course it leads you to another corner optimal point. This problem has two corner optimal point solutions.)

17 A Possible Correction

I might have said in class that if you have a feasible set described by a whole bunch (m) of <= constraints in a few (n) variables, then it may require fewer iterations to solve the dual problem, because you would only have a few (n) constraints. Examples show that you can't count on this. Here's one. Take the feasible set constructed in the solution to problem 1 on exam 1.

```
f := (x,n)->1-(x/n)^2;
```

$$f := (x,\, n) \to 1 - \frac{x^2}{n^2}$$

```
ineqs :=proc(n) local i;[seq( y -(
f(i+1,n) -f(i,n))*x <=f(i,n)-i*( f(i+1,n)-f(i,n)),i=0..n-1)] end;
```

```
ineqs := proc(n)
    locali;
        [seq(y − (f(i + 1, n) − f(i, n)) × x ≤ f(i, n) − i × (f(i + 1, n) − f(i, n)), i = 0..n − 1)]
    end
```

```
with(simplex);
```

```
Warning, new definition for basis

Warning, new definition for convexhull

Warning, new definition for cterm

Warning, new definition for display

Warning, new definition for dual

Warning, new definition for feasible

Warning, new definition for maximize

Warning, new definition for pivotvar

Warning, new definition for ratio

Warning, new definition for setup
```

```
[basis, convexhull, cterm, define_zero, display, dual, feasible, maximize, minimize, pivot,
    pivoteqn, pivotvar, ratio, setup, standardize]
```

If we look at the case where there are 7 constraints and take for the objective the middle constraint. Then we are guaranteed that starting from (0,0) we have to pivot 4 times before we get to either optimal corner point.

```
cnsts:= ineqs(7) ;
```

$$cnsts := [y + \frac{1}{49}\,x \le 1,\ y + \frac{3}{49}\,x \le \frac{51}{49},\ y + \frac{5}{49}\,x \le \frac{55}{49},\ y + \frac{1}{7}\,x \le \frac{61}{49},\ y + \frac{9}{49}\,x \le \frac{69}{49},\ y + \frac{11}{49}\,x \le \frac{79}{49},\ y + \frac{13}{49}\,x \le \frac{13}{7}]$$

```
obj := lhs(cnsts[4]);
```

$$obj := y + \frac{1}{7}\,x$$

```
primalprob := obj,cnsts:
infolevel[simplex] :=2;
```

$$\mathit{infolevel}_{\mathit{simplex}} := 2$$

```
soln := maximize(primalprob,NONNEGATIVE);
```

```
simplex/max:   \#\#\#\#\#\# final description \#\#\#\#\#\#
```

$$\begin{aligned}\mathit{eqns}\ &= ,[\_\mathit{SL1} = \frac{12}{49} + 4\,\_\mathit{SL4} - 3\,\_\mathit{SL5},\ \_\mathit{SL2} = \frac{6}{49} + 3\,\_\mathit{SL4} - 2\,\_\mathit{SL5},\\ &\_\mathit{SL3} = \frac{2}{49} + 2\,\_\mathit{SL4} - \_\mathit{SL5},\ \_\mathit{SL6} = -\_\mathit{SL4} + \frac{2}{49} + 2\,\_\mathit{SL5},\ \_\mathit{SL7} = 3\,\_\mathit{SL5} + \frac{6}{49} - 2\,\_\mathit{SL4},\\ &y = -\frac{9}{2}\,\_\mathit{SL4} + \frac{33}{49} + \frac{7}{2}\,\_\mathit{SL5},\ x = -\frac{49}{2}\,\_\mathit{SL5} + 4 + \frac{49}{2}\,\_\mathit{SL4}]\end{aligned}$$

$$\mathit{obj}\ = ,\ -\_\mathit{SL4} + \frac{61}{49}$$

$$\mathit{soln} := \{y = \frac{33}{49},\ x = 4\}$$

Sure enough, there were 4 iterations to get to this. Now in the dual problem there are only two constraints.

```
dualprob :=dual(primalprob,y);
```

$$\begin{aligned}\mathit{dualprob} := \mathit{y1} &+ \frac{51}{49}\,\mathit{y2} + \frac{55}{49}\,\mathit{y3} + \frac{61}{49}\,\mathit{y4} + \frac{69}{49}\,\mathit{y5} + \frac{79}{49}\,\mathit{y6} + \frac{13}{7}\,\mathit{y7}, [\\ &\frac{1}{7} \leq \frac{1}{49}\,\mathit{y1} + \frac{3}{49}\,\mathit{y2} + \frac{5}{49}\,\mathit{y3} + \frac{1}{7}\,\mathit{y4} + \frac{9}{49}\,\mathit{y5} + \frac{11}{49}\,\mathit{y6} + \frac{13}{49}\,\mathit{y7},\\ &1 \leq \mathit{y1} + \mathit{y2} + \mathit{y3} + \mathit{y4} + \mathit{y5} + \mathit{y6} + \mathit{y7}]\end{aligned}$$

```
minimize(dualprob,NONNEGATIVE);
```

```
simplex/max:   \#\#\#\#\#\# final description \#\#\#\#\#\#
```

$$\begin{aligned}\mathit{eqns}\ &= ,[\mathit{y4}\ = -2\,\mathit{y3} - 3\,\mathit{y2} - 4\,\mathit{y1} - \frac{49}{2}\,\_\mathit{SL1} + 1 + \frac{9}{2}\,\_\mathit{SL2} + \mathit{y6} + 2\,\mathit{y7},\\ &\mathit{y5}\ = -\frac{7}{2}\,\_\mathit{SL2} + 3\,\mathit{y1} + \frac{49}{2}\,\_\mathit{SL1} + \mathit{y3} - 2\,\mathit{y6} - 3\,\mathit{y7} + 2\,\mathit{y2}]\\ \mathit{obj}\ &= ,\ -\frac{12}{49}\,\mathit{y1} - \frac{6}{49}\,\mathit{y2} - 4\,\_\mathit{SL1} - \frac{61}{49} - \frac{33}{49}\,\_\mathit{SL2} - \frac{2}{49}\,\mathit{y3} - \frac{2}{49}\,\mathit{y6} - \frac{6}{49}\,\mathit{y7}\end{aligned}$$

$$\{\mathit{y2} = 0,\ \mathit{y1}\ = 0,\ \mathit{y5}\ = 0,\ \mathit{y6}\ = 0,\ \mathit{y7} = 0,\ \mathit{y4}\ = 1,\ \mathit{y3}\ = 0\}$$

Here there were 4 iterations in phase 1 and 3 additional in phase 2 for a total of 7 iterations! The advantage of the dual problem here is that in doing the sensitivity analysis, the matrix $B^{(-1)}$ is only 2by2 in the dual problem and 7by7 in the primal problem, so the arithmetic is much reduced in the dual problem.

18 Project 1: Due Friday before spring break

Solve one of the following problems.

1. Given a set of linear inequalities of the form:

$a\,x + b\,y + c\,z \le d$, we can finding all the corner points of the solution set by solving completely (that is, finding all multiple solutions) a series of linear programming problems. For example, to find the corner points of the tetraheadron $x + y + z \le 10,\ 0 \le x,\ 0 \le y,\ 0 \le z$, we could solve the four problems: 1. Maximize $x + y + z$, subject to $x + y + z \le 10,\ 0 \le x,\ 0 \le y,\ 0 \le z$, and: 2,3, and 4 Minimize w, subject to $x + y + z \le 0,\ x \le 0,\ y \le 0,\ z \le 0$ for $w = x,\ w = y,\ w = z$ respectively.

a) Go ahead and solve these problems, and obtain that the corner points are (0,0,0), (10,0,0), (0,10,0), and (0,0,10). b) Suppose you have a large number, say 15 or so, of inequalities in your set and you suspect that the first constraint is not binding, that is, that it could be eliminated from the set without changing the solution set. How would you verify or disprove this using linear programming? Would it be faster to solve the primal problem or the dual problem?

c) Same setting as in part b). Suppose you suspected that the solution set is empty. How would verify or disprove this using linear programming?

d) Here is a word to generate a sequence of inequalities of the type we have been discussing. Test out your answers to b and c. You can use the simplex package for your investigation.

```
genineq := proc(n)
local f, i;
f := rand(100);
seq((-1)^(f()mod 2)*f()*x +(-1)^(f() mod 2)*f()*y+(-1)^(f() mod
2)*f()*z <= (-1)^(f() mod 2)*f() ,i=1..n) end;
genineq(15);
```

2. Work out the case study at the end of Chapter 4.

3. In the worksheet on sensitivity analysis, we have a word 'sensitive' which is designe to take the output from 'mymax' and do sensitivity analysis on the right hand sides and/or objective coefficients. One of the reason cited for redoing the simplex algorithm using matrices was that sensitivity analysis could be done more efficiently. Carry out that program. Define a word 'matsense' to take the output from 'matmax' and perform sensistivity analysis on the right hand sides and the objective coefficients. Test out your word on some typical maximization problem of your construction.

19 More uses of matmax

We have seen that the matrix form of the simplex algorithm can be used to carry out sensitivity analysis on all coefficients in the problem. It can also be used to investigate the advisability of adding a new product (a new decision variable) or checking the effect of adding a new constraint. There is a difficulty that arises when adding a new constraint that leads to the need for a 'dual simplex algorithm'.

```
with(linalg):
```

So the words are gauss, getpiv and matmax.

```
gauss := proc(cl,r)
local id,n,multipliers;
n := vectdim(cl);
id := diag(1$n);
multipliers := evalm(-1/cl[r]*cl);
multipliers[r] := 1/cl[r];
augment(col(id,1..r-1),multipliers,col(id,r+1..n));
end:
```

Now getpiv takes a tableau T and a list of the column numbers of the nonbasic columns of T, and returns a suitable pivot row and column if one exist, else it returns FAIL.

```
getpiv := proc(T,nbv)
local m, i, tmp, j;
m := coldim(T);
for i in nbv do
tmp := 0,infinity;
if T[1,i]<0 then
for j from 2 to rowdim(T) do
if T[j,i]>0 and T[j,m]/T[j,i]<tmp[2]
then tmp := j,T[j,m]/T[j,i] fi;
od;
fi;
if tmp[1] > 0 then RETURN([tmp[1],i]) fi ;
od;
FAIL; end:
```

Matmax takes a tableau and a vector containing the list of column numbers of the basis for the columns of the tableau (less the right hand column.)

```
   matmax := proc( tableau , basic )
   local pivot, T, nbv, n, m, b;
   T := evalm(tableau);
   n := coldim(T);m := rowdim(T):
   b := evalm(basic);
   print(evalm(T));
   nbv :=
{seq(i,i=2..n-1)}minus{seq(b[i],i=1..m)};
   pivot := getpiv(T,nbv);
   while not pivot=FAIL do
   b[pivot[1]]:= pivot[2];
   T := evalm(gauss(col(T,pivot[2]),pivot[1])&*T);
   print(evalm(T));
   nbv := {seq(i,i=2..n-1)} minus
{seq(b[i],i=1..m)};
   pivot := getpiv(T,nbv);
```

```
od;
[evalm(T),evalm(b)];
end:
```

check this out on this problem

```
T := matrix(4,8,[1,-2,-3,-4,0,0,0,0,
0,2, 2,3,1,0,0,5,
0,1,1, 1,0,1,0,3,
0,1,2 ,1,0,0,1,4]);
```

$$T := \left[\begin{array}{cccccccc} 1 & -2 & -3 & -4 & 0 & 0 & 0 & 0 \\ 0 & 2 & 2 & 3 & 1 & 0 & 0 & 5 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 3 \\ 0 & 1 & 2 & 1 & 0 & 0 & 1 & 4 \end{array}\right]$$

Say the word matmax.

```
Tf :=matmax(T,[1,5,6,7]);
G :=
augment(col(Tf[1],1),col(Tf[1],5),col(Tf[1],6),
col(Tf[1],7));
```

Adding a decision variable

Suppose we wanted to add a new product, which sells for 5 dollars an item. Also, in each constraint, the new product uses 1/2, 2, and 1 units of the resource respectively. To check the profitability, we would simply multiply G by the new column, and get

```
newprod :=evalm(G&*[-5,1/2,2,1]);
?submatrix
```

By inspection, we see that the new item should be brought on line. just augment Tf by newprod

```
Tfn :=
augment(submatrix(Tf[1],1..4,1..7),newprod,col(Tf[1],8));
```

So the pivot column is the 2nd to last one, and the pivot row

```
3/4/(13/8.), (7/4)/(5/8.);
```

is the third one.

```
e1 := gauss(newprod,3);
evalm(e1&*Tfn);
```

It's not yet maximum. Let's go back and feed Tfn to matmax.

```
matmax(Tfn,[1,3,4,6]);
```

Now it's maximum. Make 17/11 of x1, 8/11 of x4 (the new product) and have 19/11 of the bottom resource left over for a maximum of 108/11.

Adding a constraint.

Now what about a new constraint? Suppose it has been legislated that each of the original products have a new (scarce) resource in their makeup according to the inequality

$$2\,\mathit{x1} + x\,2 + 4\,\mathit{x3} \leq 2.$$

How does this change the problem?

Well, we need a new row on our tableau and a new column, since adding a constraint adds a basic variable. We can accomplish this by first adding a column of zeros to Tf, and then adding the newrow to that.

```
Tfnr1 :=
augment(submatrix(Tf[1],1..4,1..7),[0,0,0,0],col(Tf[1],8));
Tfnr2 := stack(Tfnr1,[0,2,1,4,0,0,0,1,2]);
```

Now we have to zero out the entries at the bottom of columns 3 and 4 with two gauss matrices.

```
e1 := gauss(col(Tfnr2,3),4);
Tf4 := evalm(e1 &* Tfnr2);
e2 := gauss(col(Tf4,4),2);
Tf5 := evalm(e2 &* Tf4);
```

But, look. Now we have an infeasible solution, since s4 the new slack variable has the value -7/4 (not permitted). This is where the 'dual simplex algorithm' comes in: We attempt to get all the values of the right hand side non-negative while keeping the values of the 0 row non-negative. We will take s4 out of solution: which nonbasic variable to bring into solution. It would have to be the 2nd or 5th, and by looking at the ratios 5/4/(7/4) and 3/4/(1/4), we see that the first coefficient to become 0 is the 5/4. So the pivot column is the 5th one. Construct the Gauss matrix to make the exchange.

```
d1 := gauss(col(Tf5,5),5);
Tf6 := evalm(d1&*Tf5);
```

So one iteration of the dual simplex algorithm brings us to a feasible (optimal) solution.

20 The farmer's problem: A case problem

The problem

The Smith family farms a section (640 acres) of good Missouri farm land. The Smiths are a healthy, vigorous lot and can work 4000 hours in the winter and spring months, 4500 hours in the summer and fall months. Any hours not spent on the farm labor are worked at the local McDonald at $5 per hour in the winter-spring period and $5.50 per hour in the summer-fall period.

They already own and plan to keep 30 dairy cows worth $35,000 and 2,000 laying hens worth $5,000, and they have a cash fund of $20,000 from which to purchase, at the beginning of the year, additional cows and hens at $1,500 and $3 each respectively. Their cow barn only holds 42 cows, and their henhouse only holds 5,000 hens.

In addition, the Smiths can grow soybeans, corn, and wheat on their farm. Each cow they have during the year requires 2 acres of land for grazing, 1 acre of land planted in corn for feed, and 10 hours per month, while producing a net annual cash income of $850 for the family. Each hen requires .05 acres of land planted in wheat for feed and .05 hours per month, while producing a net annual cash income of $4.25 for the family. In a year when the weather is good for growing crops, an acre of soybeans, corn, or wheat is valued at $70, $60, or $40 respectively, whether it is sold at market or fed to the cows (in the case of corn) or hens (in the case of wheat). Of course, it takes time to grow and harvest crops: 1, .9, and .6 hours per acre in the winter and spring for soybeans, corn, and wheat respectively; 1.4, 1.2, and .7 hours per acre in the summer and fall for soybeans, corn, and wheat.

The net worth of the farm at the end of the coming year is defined as the sum of the net income from the cows and hens plus the net value of the three crops planted for the coming year plus what remains from the investment fund plus the value of the cows and hens at the end of the year minus living expenses of $40,000. We also agreed in class that the income from hours worked off the farm would be added to the net worth.

The problem is to decide on the crop production and cow and hen purchases which will maximize the net worth of the farm at the end of the year.

a) and b) The setup and solution

a) and b)

Decision variables

so = # acres planted in soybeans co = # acres planted in corn wh = # acres planted in wheat
da = number of dairy cows bought at beginning of year
he = number of hens bought at beginning of year
wsh = number of farm hours worked in winter and spring
sfh = number of farm hours worked in summer and fall

So the net worth at the end of the year is

```
with(simplex);
```

```
Warning, new definition for maximize

Warning, new definition for minimize
```

```
[basis, convexhull, cterm, define_zero, display, dual, feasible, maximize, minimize, pivot,
    pivoteqn, pivotvar, ratio, setup, standardize]
```

```
net := 0*so + 60*co + 40*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
```

$$net := 94000. + 60\,co + 40\,wh + 700.\,da + 3.50\,he - 5\,wsh - 5.5\,sfh$$

Now the constraints.

```
constraints := [wsh <= 4000, sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000),
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) = sfh];
```

the hours worked on the farm in summer-fall

$$\begin{aligned}constraints := [&wsh \le 4000,\ sfh \le 4500,\ 1500\,da + 3\,he \le 20000,\ da + 30 \le 42,\\ &he + 2000 \le 5000,\ da + 30 \le co,\ .05\,he + 100. \le wh,\ so + co + wh + 2\,da + 60 \le 640,\\ &so + .9\,co + .6\,wh + 60\,da + 2400. + .30\,he = wsh,\\ &1.4\,so + 1.2\,co + .7\,wh + 60\,da + 2400. + .30\,he = sfh]\end{aligned}$$

Fine. Now find the maximum corner point and the net worth there

```
soln:=maximize(net,constraints,NONNEGATIVE);
```

$$soln := \{co = 480.,\ wh = 100.,\ sfh = 3050.,\ wsh = 2890.,\ da = 0,\ he = 0,\ so = 0\}$$

```
subs(soln,[net,constraints]);
```

$$\begin{aligned}[&96200., [2890. \le 4000,\ 3050. \le 4500,\ 0 \le 20000,\ 30 \le 42,\ 2000 \le 5000,\ 30 \le 480.,\\ &100. \le 100.,\ 640. \le 640,\ 2890. = 2890.,\ 3050. = 3050.]]\end{aligned}$$

So to maximize the net worth of the farm at the end of the year,

the farmer should not invest in any more livestock, grow 30 acres of corn and 100 acres of wheat to feed the cows and chickenns, grow 450 acres of soybeans for sale at the market, and farm out the kids 4000-2937 hours in the winter and spring, 4500-3136 hours in the summer and fall for a maximum net worth of 99,367 dollars.

Another, slightly less profitable, but more 'friendly' solution. (obtained by Kurt Snowden.)

If we do not allow any decision on how many hours to work outside the farm, but rather maximize first the net worth of the farm obtained by work on the farm, and then add to that the addition to the net worth from the surplus hours being worked off the farm, a very different schedule is obtained which is only slightly less profitable.

```
kurtnet := 70*so + 60*co + 40*wh +
850*(da+30) + 4.25*(he+2000) +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
```

$$kurtnet := 70\,so + 60\,co + 40\,wh + 700.\,da + 49300. + 3.50\,he$$

```
kurtconstraints := [
wsh <= 4000,
sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000),
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) = sfh];
```

$$\begin{aligned}&\mathit{kurtconstraints} := [\mathit{wsh} \leq 4000,\ \mathit{sfh} \leq 4500,\ 1500\ \mathit{da} + 3\ \mathit{he} \leq 20000,\ \mathit{da} + 30 \leq 42,\\ &\quad \mathit{he} + 2000 \leq 5000,\ \mathit{da} + 30 \leq \mathit{co},\ .05\ \mathit{he} + 100. \leq \mathit{wh},\ \mathit{so} + \mathit{co} + \mathit{wh} + 2\ \mathit{da} + 60 \leq 640,\\ &\quad \mathit{so} + .9\ \mathit{co} + .6\ \mathit{wh} + 60\ \mathit{da} + 2400. + .30\ \mathit{he} = \mathit{wsh},\\ &\quad 1.4\ \mathit{so} + 1.2\ \mathit{co} + .7\ \mathit{wh} + 60\ \mathit{da} + 2400. + .30\ \mathit{he} = \mathit{sfh}]\end{aligned}$$

```
kurtsoln:=maximize(kurtnet,kurtconstraints,NO
NNEGATIVE);
```

$$\begin{aligned}&\mathit{kurtsoln} :=\\ &\quad \{\mathit{co} = 39.8,\ \mathit{so} = 332.,\ \mathit{sfh} = 4160.,\ \mathit{wh} = 189.,\ \mathit{da} = 9.79,\ \mathit{he} = 1770.,\ \mathit{wsh} = 4000.\}\end{aligned}$$

Here the maximum networth not counting outside hours worked is

```
subs(kurtsoln,[kurtnet,kurtconstraints]);
```

$$\begin{aligned}&[95600., [4000. \leq 4000,\ 4160. \leq 4500,\ 20000. \leq 20000,\ 39.8 \leq 42,\ 3770. \leq 5000,\\ &\quad 39.8 \leq 39.8,\ 189. \leq 189.,\ 641. \leq 640,\ 4000. = 4000.,\ 4160. = 4160.]]\end{aligned}$$

\$95,478.34. We note that 4500-4163.6 hours can be worked outside for a total networth of

```
subs(kurtsoln,kurtnet+5.5*(4500-sfh));
```

$$97100.$$

This is less than the maximum networth we get when hours worked outside is a decision variable, but not not much less. See how much different the schedule is: grow 188.6 acres of wheat, 332 acres of soybeans, 39.8 acres of corn and buy 9.8 cows and 1772.2 hens. All but around 340 hours is worked on the farm, a more 'farmerly' solution for sure.

20.1 c) and d) The six weather scenarios.

The solution obtained above in a) and b) was for good weather, which one doesn't always have. There are five other weather scenarios which occur often enough to worry about: drought, flood, frost, frost_then_dry, flood_then_frost. In these scenarios, the value of an acre of crop is different. For soybeans it is -10, 15, 50, -15, and 10 for each scenario respectively. For corn and wheat respectively , it is -15, 20, 40, -20, 10 and 0, 10, 30, -10, 5 respectively. We can make a word to handle these various cases easily by just copying down the input cells we used to do the good weather scenario, and putting a proc line at the top and an end line at the bottom. Also, we improved the output slightly to make it more readable.

```
scenario := proc(a,b,c)
local net,constraints,soln,wsh,sfh,so,
co,wh,he,da;
net := a*so + b*co + c*wh +
850*(da+30) + 4.25*(he+2000) +
```

```
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
constraints := [
wsh <= 4000,
sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000),
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) = sfh];
soln:=maximize(net,constraints,NONNEGATIVE);
subs(soln,matrix(8,2,[networth,net,
corn, co, soybeans, so,
wheat,wh,cows,da,hens,he,
winterspring,wsh,summerfall,sfh ])); end;
```

```
scenario := proc(a, b, c)
  localnet, constraints, soln, wsh, sfh, so, co, wh, he, da;
    net := a × so + b × co + c × wh + 700. × da + 94000. + 3.50 × he − 5 × wsh − 5.5 × sfh ;
    constraints := [wsh ≤ 4000, sfh ≤ 4500, 1500 × da + 3 × he ≤ 20000, da + 30 ≤ 42,
        he + 2000 ≤ 5000, da + 30 ≤ co, .05 × he + 100. ≤ wh, so + co + wh + 2 × da + 60 ≤ 640,
        so + .9 × co + .6 × wh + 60 × da + 2400. + .30 × he = wsh,
        1.4 × so + 1.2 × co + .7 × wh + 60 × da + 2400. + .30 × he = sfh];
    soln := maximize(net, constraints, NONNEGATIVE) ;
    subs(soln, matrix(8, 2, [networth, net, corn, co, soybeans, so, wheat, wh, cows, da, hens,
        he, winterspring, wsh, summerfall, sfh]))
  end
```

Checking this out on goodweather,

```
infolevel[simplex]:=1;
```

$$infolevel_{simplex} := 1$$

```
goodweather :=scenario(70,60,40);
```

$$goodweather := \begin{bmatrix} networth & 98700. \\ corn & 30.0 \\ soybeans & 450. \\ wheat & 100. \\ cows & 0 \\ hens & 0 \\ winterspring & 2940. \\ summerfall & 3140. \end{bmatrix}$$

Now maximize the networth in each of the other five scenarios.

```
drought :=scenario(-10,-15,0);
flood := scenario(15,20,10);
frost := scenario(50,40,30);
dryandfrost := scenario(-15,-20,-10);
```

```
floodandfrost := scenario(10,10,5);
```

$$drought := \left[\begin{array}{cc} networth & 67800. \\ corn & 42. \\ soybeans & 0 \\ wheat & 134. \\ cows & 12. \\ hens & 667. \\ winterspring & 3440. \\ summerfall & 3460. \end{array}\right]$$

$$flood := \left[\begin{array}{cc} networth & 73500. \\ corn & 422. \\ soybeans & 0 \\ wheat & 134. \\ cows & 12. \\ hens & 667. \\ winterspring & 3780. \\ summerfall & 3920. \end{array}\right]$$

$$frost := \left[\begin{array}{cc} networth & 88700. \\ corn & 30.0 \\ soybeans & 450. \\ wheat & 100. \\ cows & 0 \\ hens & 0 \\ winterspring & 2940. \\ summerfall & 3140. \end{array}\right]$$

$$dryandfrost := \left[\begin{array}{cc} networth & 67100. \\ corn & 42. \\ soybeans & 0 \\ wheat & 100. \\ cows & 12. \\ hens & 0 \\ winterspring & 3220. \\ summerfall & 3240. \end{array}\right]$$

$$floodandfrost := \left[\begin{array}{cc} networth & 70600. \\ corn & 37.3 \\ soybeans & 0 \\ wheat & 250. \\ cows & 7.33 \\ hens & 3000. \\ winterspring & 3920. \\ summerfall & 3960. \end{array}\right]$$

The scenarios rank in order of decreasing maximum networth:
goodweather, frost, flood, flood and frost, drought, dry and frost.

No surprises here. The suggestion is to calculate a 6by6 table of networths for each scenario using all six corner point solutions found and trying to pick the best corner point solution among the six. We can do this from where we are by modifying the original networth expression

```
net := (a,b,c,so,co,wh,da,he,wsh,sfh) ->a*so
+ b*co + c*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
```

$$net := (a, b, c, so, co, wh, da, he, wsh, sfh) \to a\,so + b\,co + c\,wh + 700.\,da + 94000. + 3.50\,he - 5\,wsh - 5.5\,sfh$$

Make a list of the scenarios and cropvalues.

```
scenes := [goodweather,drought, flood, frost,
dryandfrost, floodandfrost];
```

$$scenes := [goodweather, drought, flood, frost, dryandfrost, floodandfrost]$$

```
cropvalues := [[70,60,40],[-10,-15,0],
[15,20,10],[50,40,30],
[-15,-20,-10],[10,10,5]];
```

$$cropvalues := [[70, 60, 40], [-10, -15, 0], [15, 20, 10], [50, 40, 30], [-15, -20, -10], [10, 10, 5]]$$

Now make a sequence of the 36 networths.

```
networths := NULL;
```

$$networths :=$$

Make digits smaller and redefine zero so that the numbers will fit nicely.

```
Digits := 6:define_zero(10^(-4));
```

$$.000100000$$

```
for cr in cropvalues do for sc in scenes do
networths:=networths,subs({co=sc[2,2],so=sc[3,2]
, wh=sc[4,2],da=sc[5,2],he=sc[6,2],wsh=sc[7,2],
sfh=sc[8,2]}, net(cr[1],cr[2],cr[3],so,co,wh,da,he,wsh,sfh)) od od;
mat :=matrix(6,6,[networths]);
```

$$mat := \begin{bmatrix} 99330. & 76385. & 94955. & 99330. & 75000. & 80489. \\ 57080.0 & 67875. & 57944.5 & 57080.0 & 67850. & 67692. \\ 70380. & 70685. & 74055. & 70380. & 70320. & 71497. \\ 88730. & 74205. & 85175. & 88730. & 73160. & 77243. \\ 53680.0 & 66325. & 54494.5 & 53680.0 & 66640. & 65005. \\ 67330.0 & 69595. & 69165. & 67330.0 & 69400. & 69874. \end{bmatrix}$$

In this array of networths, a row represent the networths computed at each corner maximum point for a particular set of cropvalues, and each column represents the networths that occur at that corner point under each of the weather scenarios. We can scan the columns and look for a corner point (ie schedule of planting) which gives the best balance among all the columns. It is pretty clear that the columns with the best balance are the 2nd, 5th, and 6th ones. We could check that by averaging the columns and checking standard deviation in the columns.

```
with(stats);
```

$$[\mathit{anova},\ \mathit{describe},\ \mathit{fit},\ \mathit{importdata},\ \mathit{random},\ \mathit{statevalf},\ \mathit{statplots},\ \mathit{transform}]$$

```
seq(
describe[mean]([seq(mat[i,j],i=1..6)]),j=1..6);
```

$$72755.1,\ 70845.1,\ 72631.8,\ 72755.1,\ 70395.1,\ 71966.8$$

```
seq(
describe[standarddeviation]([seq(mat[i,j],i=1..6)]),j=1..6);
```

$$16360.8,\ 3483.79,\ 14226.6,\ 16360.8,\ 2898.97,\ 5351.14$$

Actually, we see that the deviation from the mean is lowest in the 5th column, so the risk is lower there. But the networth is lower also.

The next parts give a better way to lower risk and increase networth simultaneously, if you have access to weather records.

e) and f) Taking probabilities into account.

The weather records over the past several decades have been examined and empirical probabilities of each weather scenario has been calculated from the data. They are .4, .2, .1 .15, .1 , and .05 for goodweather, drought, flood, frost, droughtandfrost, and floodandfrost respectively. So, new maximum networth is calculated using the 'expected crop values', that is, the weighted average of values across the 6 scenarios.

```
cr := matrix(cropvalues);
probs := [.4,.2,.1,.15,.1,.05];
```

$$cr := \begin{bmatrix} 70 & 60 & 40 \\ -10 & -15 & 0 \\ 15 & 20 & 10 \\ 50 & 40 & 30 \\ -15 & -20 & -10 \\ 10 & 10 & 5 \end{bmatrix}$$

$$\mathit{probs} := [.4,\ .2,\ .1,\ .15,\ .1,\ .05]$$

```
av :=evalm(probs&*cr);
```

$$av := [34.00,\ 27.50,\ 20.75]$$

```
scenario(av[1],av[2],av[3]);
```

$$\begin{bmatrix} \mathit{networth} & 80537.0 \\ \mathit{corn} & 42.0000 \\ \mathit{soybeans} & 414.000 \\ \mathit{wheat} & 100.000 \\ \mathit{cows} & 12. \\ \mathit{hens} & 0 \\ \mathit{winterspring} & 3631.80 \\ \mathit{summerfall} & 3820.00 \end{bmatrix}$$

So, the 'expected maximum net worth' of the farm is \$80,573 with a planting schedule of 414 acres soybeans, no new hens, 12 new cows, 100 a. wheat, 42 a. corn, 3631.8 wsh, and 3820 sfh.

We acually didn't carry out the instruction to the letter, there are six new constraints imposed here: namely, no planting schedule is allowed which reduces the networth of the farm by more than 10,000 under any weather scenario. We calculate the farms networth at the beginning of the year to be 60,000= value of livestock + investment fund. So we want to add in 6 constraints,

one for each cropvalue, and resolve the problem.

```
Digits := 10:define_zero(10^(-7));
```

$$.1000000000\ 10^{-6}$$

```
with(linalg):  with(simplex):

Warning, new definition for basis

Warning, new definition for norm

Warning, new definition for pivot

Warning, new definition for trace

Warning, new definition for basis

Warning, new definition for convexhull

Warning, new definition for cterm

Warning, new definition for display

Warning, new definition for dual

Warning, new definition for feasible

Warning, new definition for maximize

Warning, new definition for pivot

Warning, new definition for pivotvar

Warning, new definition for ratio

Warning, new definition for setup
```

```
net := 34*so + 27.5*co + 20.75*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
constraints := [
70*so + 60 *co + 40*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
-10*so -15 *co + 0*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
15*so + 20 *co + 10*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
```

```
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
50*so + 40 *co + 30*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
-15*so -20 *co -10*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
10*so + 10 *co + 5*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
wsh <= 4000,
sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000),
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) = sfh];
soln:=maximize(net,constraints,NONNEGATIVE);
subs(soln,matrix(8,2,[networth,net,
corn, co, soybeans, so,
wheat,wh,cows,da,hens,he,
winterspring,wsh,summerfall,sfh ]));
```

$$\mathit{net} := 34\,\mathit{so} + 27.5\,\mathit{co} + 20.75\,\mathit{wh} + 700.0\,\mathit{da} + 94000.00 + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh}$$

$$\begin{aligned}
&\mathit{constraints} := [\\
&\quad 60000 \leq 70\,\mathit{so} + 60\,\mathit{co} + 40\,\mathit{wh} + 700.0\,\mathit{da} + 104000.00 + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh},\\
&\quad 60000 \leq -10\,\mathit{so} - 15\,\mathit{co} + 104000.00 + 700.0\,\mathit{da} + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh},\\
&\quad 60000 \leq 15\,\mathit{so} + 20\,\mathit{co} + 10\,\mathit{wh} + 700.0\,\mathit{da} + 104000.00 + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh},\\
&\quad 60000 \leq 50\,\mathit{so} + 40\,\mathit{co} + 30\,\mathit{wh} + 700.0\,\mathit{da} + 104000.00 + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh},\\
&\quad 60000 \leq -15\,\mathit{so} - 20\,\mathit{co} - 10\,\mathit{wh} + 700.0\,\mathit{da} + 104000.00 + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh},\\
&\quad 60000 \leq 10\,\mathit{so} + 10\,\mathit{co} + 5\,\mathit{wh} + 700.0\,\mathit{da} + 104000.00 + 3.50\,\mathit{he} - 5\,\mathit{wsh} - 5.5\,\mathit{sfh},\\
&\quad \mathit{wsh} \leq 4000,\ \mathit{sfh} \leq 4500,\ 1500\,\mathit{da} + 3\,\mathit{he} \leq 20000,\ \mathit{da} + 30 \leq 42,\ \mathit{he} + 2000 \leq 5000,\\
&\quad \mathit{da} + 30 \leq \mathit{co},\ .05\,\mathit{he} + 100.00 \leq \mathit{wh},\ \mathit{so} + \mathit{co} + \mathit{wh} + 2\,\mathit{da} + 60 \leq 640,\\
&\quad \mathit{so} + .9\,\mathit{co} + .6\,\mathit{wh} + 60\,\mathit{da} + 2400.00 + .30\,\mathit{he} = \mathit{wsh},\\
&\quad 1.4\,\mathit{so} + 1.2\,\mathit{co} + .7\,\mathit{wh} + 60\,\mathit{da} + 2400.00 + .30\,\mathit{he} = \mathit{sfh}]
\end{aligned}$$

$$\begin{aligned}
&\mathit{soln} := \{\mathit{wh} = 100.0000000,\ \mathit{sfh} = 3820.000000,\ \mathit{so} = 414.0000000,\ \mathit{co} = 42.00000000,\\
&\quad \mathit{da} = 12.00000000,\ \mathit{he} = 0,\ \mathit{wsh} = 3631.800000\}
\end{aligned}$$

$$\begin{bmatrix} \mathit{networth} & 80537.00000 \\ \mathit{corn} & 42.00000000 \\ \mathit{soybeans} & 414.0000000 \\ \mathit{wheat} & 100.0000000 \\ \mathit{cows} & 12.00000000 \\ \mathit{hens} & 0 \\ \mathit{winterspring} & 3631.800000 \\ \mathit{summerfall} & 3820.000000 \end{bmatrix}$$

Why don't I find it surprizing that those six constraints were not binding. All that work, we get the same solution!

20.2 g) and h) Some sensitivity analysis.

Use a shadow price in f) to decide the answer to this question:

Would it increase the networth to borrow money at 10% to purchase cows or hens beyond what can be obtained with the 20,000 from the investment fund?

Solution: First we have to go back and resolve with the infolevel set to 2, so that we can see the final set of equations.

```
infolevel[simplex]:=2;
```

$$\mathit{infolevel}_{\mathit{simplex}} := 2$$

```
maximize(net ,constraints,NONNEGATIVE);
```

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{aligned}
\mathit{eqns} = {}&,[\_\mathit{SL1} = 24000.00 + 60.3333333334\ \mathit{so} + 51.5000000000\ \mathit{co}\\
&+ 34.6666666666\ \mathit{wh} - 2.16666666663\ \mathit{sfh} + \mathit{he} + 200.\ \mathit{da}, \_\mathit{SL2} = 24000.00\\
&- 19.6666666666\ \mathit{so} - 23.5000000000\ \mathit{co} - 5.33333333333\ \mathit{wh} - 2.16666666663\ \mathit{sfh}\\
&+ \mathit{he} + 200.\ \mathit{da}, \_\mathit{SL3} = 24000.00 + 5.33333333337\ \mathit{so} + 11.5000000000\ \mathit{co}\\
&+ 4.66666666667\ \mathit{wh} - 2.16666666663\ \mathit{sfh} + \mathit{he} + 200.\ \mathit{da}, \_\mathit{SL4} = 24000.00\\
&+ 40.3333333334\ \mathit{so} + 31.5000000000\ \mathit{co} + 24.6666666666\ \mathit{wh} - 2.16666666663\ \mathit{sfh}\\
&+ \mathit{he} + 200.\ \mathit{da}, \_\mathit{SL5} = 24000.00 - 24.6666666666\ \mathit{so} - 28.5000000000\ \mathit{co}\\
&- 15.3333333334\ \mathit{wh} - 2.16666666663\ \mathit{sfh} + \mathit{he} + 200.\ \mathit{da}, \_\mathit{SL6} = 24000.00\\
&+ .33333333337\ \mathit{so} + 1.5000000000\ \mathit{co} - .33333333333\ \mathit{wh} - 2.16666666663\ \mathit{sfh} + \mathit{he}\\
&+ 200.\ \mathit{da}, \_\mathit{SL7} = 12000. + 5.06666666667\ \mathit{so} + 4.30000000000\ \mathit{co}\\
&+ 2.43333333333\ \mathit{wh} - 4.33333333333\ \mathit{sfh} + \mathit{he} + 200.\ \mathit{da}, \_\mathit{SL8} = 12500.\\
&- 4.33333333333\ \mathit{sfh} + \mathit{he} + 4.66666666667\ \mathit{so} + 4.\ \mathit{co} + 2.33333333333\ \mathit{wh} + 200.\ \mathit{da},\\
&\_\mathit{SL9} = 52000. - 700.\ \mathit{da} + 18.6666666667\ \mathit{so} + 16.\ \mathit{co} + 9.33333333332\ \mathit{wh}\\
&- 13.3333333333\ \mathit{sfh} + \mathit{he}, \_\mathit{SL10} = 8012. + 199.\ \mathit{da} + \mathit{he} + 4.66666666667\ \mathit{so} + 4.\ \mathit{co}\\
&+ 2.33333333333\ \mathit{wh} - 3.33333333333\ \mathit{sfh}, \_\mathit{SL11} = 19000. + 9.33333333334\ \mathit{so} + 8.\ \mathit{co}\\
&+ 4.66666666666\ \mathit{wh} + 400.\ \mathit{da} - 6.66666666666\ \mathit{sfh} + \mathit{he}, \_\mathit{SL12} = 7970. + 199.\ \mathit{da}\\
&+ 5.\ \mathit{co} + \mathit{he} + 4.66666666667\ \mathit{so} + 2.33333333333\ \mathit{wh} - 3.33333333333\ \mathit{sfh}, \_\mathit{SL13} =\\
&8300.00 + 4.90000000000\ \mathit{so} + 4.20\ \mathit{co} + 3.45000000000\ \mathit{wh} + 210.00\ \mathit{da}\\
&- 3.50000000000\ \mathit{sfh} + \mathit{he}, \_\mathit{SL14} = 8580. + 3.66666666667\ \mathit{so} + 3.\ \mathit{co}\\
&+ 1.33333333333\ \mathit{wh} + 198.\ \mathit{da} + \mathit{he} - 3.33333333333\ \mathit{sfh}, \_\mathit{AR} = \mathit{he}\\
&+ 4.66666666667\ \mathit{so} + 4.\ \mathit{co} + 2.33333333333\ \mathit{wh} + 200.\ \mathit{da} - 3.33333333333\ \mathit{sfh}\\
&+ 8000., \mathit{wsh} = 4.26666666667\ \mathit{so} + 3.70000000000\ \mathit{co} + 2.23333333333\ \mathit{wh}\\
&- 2.33333333333\ \mathit{sfh} + \mathit{he} + 200.\ \mathit{da} + 8000.]
\end{aligned}$$

$$\begin{aligned}\mathit{obj} \ &= , -\mathit{he} - 4.66666666667\ \mathit{so} - 4.\ \mathit{co} - 2.33333333333\ \mathit{wh} - 200.\ \mathit{da}\\ &+ 3.33333333333\ \mathit{sfh} - 8000.\end{aligned}$$

$$\mathit{pivot\ variable}\ = ,\ \mathit{sfh}$$

$$\begin{aligned}\mathit{pivot\ equation}\ &= , \mathit{\_SL13} = 8300.00 + 4.90000000000\ \mathit{so} + 4.20\ \mathit{co} + 3.45000000000\ \mathit{wh}\\ &+ 210.00\ \mathit{da} - 3.50000000000\ \mathit{sfh} + \mathit{he}\end{aligned}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{aligned}\mathit{eqns}\ = , [\mathit{\_SL1} &= 18861.9047620 + 57.3000000001\ \mathit{so} + 48.9000000000\ \mathit{co}\\
&+ 32.5309523809\ \mathit{wh} + .619047619037\ \mathit{\_SL13} + 70.000000002\ \mathit{da} + .380952380963\ \mathit{he},\\
&\mathit{\_SL2} = 18861.9047620 - 22.6999999999\ \mathit{so} - 26.1000000000\ \mathit{co} - 7.46904761901\ \mathit{wh}\\
&+ .619047619037\ \mathit{\_SL13} + 70.000000002\ \mathit{da} + .380952380963\ \mathit{he}, \mathit{\_SL3} =\\
&18861.9047620 + 2.30000000009\ \mathit{so} + 8.90000000004\ \mathit{co} + 2.53095238099\ \mathit{wh}\\
&+ .619047619037\ \mathit{\_SL13} + 70.000000002\ \mathit{da} + .380952380963\ \mathit{he}, \mathit{\_SL4} =\\
&18861.9047620 + 37.3000000001\ \mathit{so} + 28.9000000000\ \mathit{co} + 22.5309523809\ \mathit{wh}\\
&+ .619047619037\ \mathit{\_SL13} + 70.000000002\ \mathit{da} + .380952380963\ \mathit{he}, \mathit{\_SL5} =\\
&18861.9047620 - 27.6999999999\ \mathit{so} - 31.1000000000\ \mathit{co} - 17.4690476191\ \mathit{wh}\\
&+ .619047619037\ \mathit{\_SL13} + 70.000000002\ \mathit{da} + .380952380963\ \mathit{he}, \mathit{\_SL6} =\\
&18861.9047620 - 2.69999999991\ \mathit{so} - 1.09999999996\ \mathit{co} - 2.46904761901\ \mathit{wh}\\
&+ .619047619037\ \mathit{\_SL13} + 70.000000002\ \mathit{da} + .380952380963\ \mathit{he}, \mathit{\_SL7} = 1723.8095238\\
&- .99999999999\ \mathit{so} - .90000000000\ \mathit{co} - 1.83809523809\ \mathit{wh} + 1.23809523809\ \mathit{\_SL13}\\
&- 60.000000000\ \mathit{da} - .23809523809\ \mathit{he}, \mathit{\_SL8} = 2223.8095238 + 1.23809523809\ \mathit{\_SL13}\\
&- 1.39999999999\ \mathit{so} - 1.20000000000\ \mathit{co} - 1.93809523809\ \mathit{wh} - 60.000000000\ \mathit{da}\\
&- .23809523809\ \mathit{he}, \mathit{\_SL9} = 20380.9523810 - 1500.00000000\ \mathit{da} + .1\,10^{-9}\ \mathit{so}\\
&- 3.80952380948\ \mathit{wh} + 3.80952380951\ \mathit{\_SL13} - 2.80952380951\ \mathit{he}, \mathit{\_SL10} =\\
&107.23809524 - 1.000000000\ \mathit{da} + .047619047621\ \mathit{he} + .1\,10^{-10}\ \mathit{so} - .95238095238\ \mathit{wh}\\
&+ .952380952379\ \mathit{\_SL13}, \mathit{\_SL11} = 3190.4761905 + .2\,10^{-10}\ \mathit{so} + .1\,10^{-10}\ \mathit{co}\\
&- 1.90476190476\ \mathit{wh} + 1.90476190476\ \mathit{\_SL13} - .90476190476\ \mathit{he}, \mathit{\_SL12} =\\
&65.23809524 - 1.000000000\ \mathit{da} + 1.00000000000\ \mathit{co} + .047619047621\ \mathit{he} + .1\,10^{-10}\ \mathit{so}\\
&- .95238095238\ \mathit{wh} + .952380952379\ \mathit{\_SL13}, \mathit{sfh} = -.285714285714\ \mathit{\_SL13}\\
&+ 2371.42857143 + 1.40000000000\ \mathit{so} + 1.20000000000\ \mathit{co} + .985714285714\ \mathit{wh}\\
&+ 60.\ \mathit{da} + .285714285714\ \mathit{he}, \mathit{\_SL14} = 675.23809524 - .99999999999\ \mathit{so}\\
&- 1.00000000000\ \mathit{co} - 1.95238095238\ \mathit{wh} - 2.000000000\ \mathit{da} + .047619047621\ \mathit{he}\\
&+ .952380952379\ \mathit{\_SL13}, \mathit{\_AR} = .047619047621\ \mathit{he} + .1\,10^{-10}\ \mathit{so} - .95238095238\ \mathit{wh}\\
&+ .952380952379\ \mathit{\_SL13} + 95.23809524, \mathit{wsh} = 1.00000000001\ \mathit{so} + .90000000000\ \mathit{co}\\
&- .06666666667\ \mathit{wh} + .666666666665\ \mathit{\_SL13} + 2466.66666667 + 60.000000000\ \mathit{da}\\
&+ .333333333335\ \mathit{he}]\end{aligned}$$

$$\begin{aligned}\mathit{obj}\ &= , -.047619047621\ \mathit{he} - .1\,10^{-10}\ \mathit{so} + .95238095238\ \mathit{wh} - .952380952379\ \mathit{\_SL13}\\ &- 95.23809524\end{aligned}$$

$$\mathit{pivot\ variable}\ = ,\ \mathit{wh}$$

$$\begin{aligned}\mathit{pivot\ equation}\ &= , \mathit{\_SL12} = 65.23809524 - 1.000000000\ \mathit{da} + 1.00000000000\ \mathit{co}\\ &+ .047619047621\ \mathit{he} + .1\,10^{-10}\ \mathit{so} - .95238095238\ \mathit{wh} + .952380952379\ \mathit{\_SL13}\end{aligned}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{aligned}\mathit{eqns}\ = , [\mathit{\_SL1} &= 21090.2750002 + 57.3000000004\ \mathit{so} + 83.0574999999\ \mathit{co}\\
&- 34.1574999999\ \mathit{\_SL12} + 35.8425000021\ \mathit{da} + 2.00750000007\ \mathit{he}\\
&+ 33.1499999999\ \mathit{\_SL13}, \mathit{\_SL2} = 18350.2750001 - 22.7000000000\ \mathit{so}\\
&- 33.9425000000\ \mathit{co} + 7.84249999996\ \mathit{\_SL12} + 77.8425000020\ \mathit{da} + .007499999997\ \mathit{he}\\
&- 6.84999999996\ \mathit{\_SL13}, \mathit{\_SL3} = 19035.2750001 + 2.30000000012\ \mathit{so}\\
&+ 11.5575000001\ \mathit{co} - 2.65750000004\ \mathit{\_SL12} + 67.3425000020\ \mathit{da} + .507500000018\ \mathit{he}\end{aligned}$$

$$\begin{aligned}
&+3.15000000003\ \mathit{\_SL13},\mathit{\_SL4}=20405.2750001+37.3000000003\ \mathit{so}\\
&+52.5574999999\ \mathit{co}-23.6574999999\ \mathit{\_SL12}+46.3425000021\ \mathit{da}+1.50750000005\ \mathit{he}\\
&+23.1499999999\ \mathit{\_SL13},\mathit{\_SL5}=17665.2750001-27.7000000001\ \mathit{so}\\
&-49.4425000001\ \mathit{co}+18.3425000001\ \mathit{\_SL12}+88.3425000021\ \mathit{da}-.492500000029\ \mathit{he}\\
&-16.8500000001\ \mathit{\_SL13},\mathit{\_SL6}=18692.7750001-2.69999999994\ \mathit{so}\\
&-3.69249999992\ \mathit{co}+2.59249999996\ \mathit{\_SL12}+72.5925000020\ \mathit{da}+.257500000007\ \mathit{he}\\
&-1.84999999997\ \mathit{\_SL13},\mathit{\_SL7}=1597.89999999-1.00000000001\ \mathit{so}\\
&-2.82999999999\ \mathit{co}+1.92999999999\ \mathit{\_SL12}-58.0700000000\ \mathit{da}-.329999999998\ \mathit{he}\\
&-.60000000000\ \mathit{\_SL13},\mathit{\_SL8}=2091.04999999-.70000000000\ \mathit{\_SL13}\\
&-1.40000000001\ \mathit{so}-3.23499999999\ \mathit{co}+2.03499999999\ \mathit{\_SL12}-57.9650000000\ \mathit{da}\\
&-.334999999999\ \mathit{he},\mathit{\_SL9}=20120.0000000-1496.00000000\ \mathit{da}\\
&+.600000000005\ 10^{-10}\ \mathit{so}+3.99999999995\ \mathit{\_SL12}-3.99999999995\ \mathit{co}\\
&-2.99999999999\ \mathit{he}+.3\ 10^{-10}\ \mathit{\_SL13},\mathit{\_SL10}=\\
&42.0000000000-.1\ 10^{-11}\ \mathit{da}+.1\ 10^{-22}\ \mathit{so}+.999999999999\ \mathit{\_SL12}-.999999999999\ \mathit{co},\\
&\mathit{\_SL11}=3060.00000002-1.99999999999\ \mathit{co}+2.00000000000\ \mathit{\_SL12}\\
&+2.00000000000\ \mathit{da}-1.00000000000\ \mathit{he},\mathit{wh}=-1.05000000000\ \mathit{\_SL12}\\
&+68.5000000021-1.05000000000\ \mathit{da}+1.05000000000\ \mathit{co}+.0500000000021\ \mathit{he}\\
&+.105000000000\ 10^{-10}\ \mathit{so}+.999999999999\ \mathit{\_SL13},\mathit{sfh}=.699999999999\ \mathit{\_SL13}\\
&+2438.95000000+1.40000000001\ \mathit{so}+2.23500000000\ \mathit{co}-1.03500000000\ \mathit{\_SL12}\\
&+58.9650000000\ \mathit{da}+.335000000002\ \mathit{he},\mathit{\_SL14}=541.499999998-1.00000000001\ \mathit{so}\\
&-3.05000000000\ \mathit{co}+2.05000000000\ \mathit{\_SL12}+.05000000000\ \mathit{da}-.0500000000021\ \mathit{he}\\
&-1.00000000000\ \mathit{\_SL13},\mathit{\_AR}=.1\ 10^{-22}\ \mathit{so}+.999999999999\ \mathit{\_SL12}+30.0000000000\\
&+.999999999999\ \mathit{da}-.999999999999\ \mathit{co},\mathit{wsh}=1.00000000001\ \mathit{so}\\
&+.829999999997\ \mathit{co}+.0700000000035\ \mathit{\_SL12}+2462.10000000+60.0700000000\ \mathit{da}\\
&+.330000000001\ \mathit{he}+.599999999995\ \mathit{\_SL13}]
\end{aligned}$$

$$\begin{aligned}
\mathit{obj}\ =\ &,-.1\ 10^{-22}\ \mathit{so}-.999999999999\ \mathit{\_SL12}-30.0000000000-.999999999999\ \mathit{da}\\
&+.999999999999\ \mathit{co}
\end{aligned}$$

$$\mathit{pivot\ variable}\ =\ ,\ \mathit{co}$$

$$\begin{aligned}
\mathit{pivot\ equation}\ =\ &,\mathit{\_AR}=.1\ 10^{-22}\ \mathit{so}+.999999999999\ \mathit{\_SL12}+30.0000000000\\
&+.999999999999\ \mathit{da}-.999999999999\ \mathit{co}
\end{aligned}$$

```
simplex/max:   \#\#\#\#\#\# final description \#\#\#\#\#\#
```

$$\begin{aligned}
\mathit{eqns}\ =\ &,[\mathit{\_SL1}=23582.0000002+57.3000000004\ \mathit{so}-83.0574999999\ \mathit{\_AR}\\
&+48.9000000000\ \mathit{\_SL12}+118.900000002\ \mathit{da}+2.00750000007\ \mathit{he}\\
&+33.1499999999\ \mathit{\_SL13},\mathit{\_SL2}=17332.0000001-22.7000000000\ \mathit{so}\\
&+33.9425000000\ \mathit{\_AR}-26.1000000000\ \mathit{\_SL12}+43.9000000020\ \mathit{da}\\
&+.007499999997\ \mathit{he}-6.84999999996\ \mathit{\_SL13},\mathit{\_SL3}=19382.0000001\\
&+2.30000000012\ \mathit{so}-11.5575000001\ \mathit{\_AR}+8.90000000006\ \mathit{\_SL12}\\
&+78.9000000021\ \mathit{da}+.507500000018\ \mathit{he}+3.15000000003\ \mathit{\_SL13},\mathit{\_SL4}=\\
&21982.0000001+37.3000000003\ \mathit{so}-52.5574999999\ \mathit{\_AR}+28.9000000000\ \mathit{\_SL12}\\
&+98.9000000020\ \mathit{da}+1.50750000005\ \mathit{he}+23.1499999999\ \mathit{\_SL13},\mathit{\_SL5}=\\
&16182.0000001-27.7000000001\ \mathit{so}+49.4425000001\ \mathit{\_AR}-31.1000000000\ \mathit{\_SL12}\\
&+38.9000000020\ \mathit{da}-.492500000029\ \mathit{he}-16.8500000001\ \mathit{\_SL13},\mathit{\_SL6}=\\
&18582.0000001-2.69999999994\ \mathit{so}+3.69249999992\ \mathit{\_AR}-1.09999999996\ \mathit{\_SL12}\\
&+68.9000000021\ \mathit{da}+.257500000007\ \mathit{he}-1.84999999997\ \mathit{\_SL13},\mathit{\_SL7}=\\
&1512.99999999-1.00000000001\ \mathit{so}+2.82999999999\ \mathit{\_AR}-.90000000000\ \mathit{\_SL12}\\
&-60.9000000000\ \mathit{da}-.329999999998\ \mathit{he}-.60000000000\ \mathit{\_SL13},\mathit{\_SL8}=\\
&1993.99999999-.70000000000\ \mathit{\_SL13}-1.40000000001\ \mathit{so}+3.23499999999\ \mathit{\_AR}\\
&-1.20000000000\ \mathit{\_SL12}-61.2000000000\ \mathit{da}-.334999999999\ \mathit{he},\mathit{\_SL9}=\\
&20000.0000000-1500.00000000\ \mathit{da}+.600000000005\ 10^{-10}\ \mathit{so}+3.99999999995\ \mathit{\_AR}\\
&-2.99999999999\ \mathit{he}+.3\ 10^{-10}\ \mathit{\_SL13},\\
&\mathit{\_SL10}=12.0000000000-1.00000000000\ \mathit{da}+.1\ 10^{-34}\ \mathit{so}+.999999999999\ \mathit{\_AR},\mathit{\_SL11}\\
&=3000.00000002+1.99999999999\ \mathit{\_AR}-.199999999999\ 10^{-22}\ \mathit{so}+.1\ 10^{-10}\ \mathit{\_SL12}\\
&+.1\ 10^{-10}\ \mathit{da}-1.00000000000\ \mathit{he},\mathit{wh}=100.000000002-1.05000000000\ \mathit{\_AR}
\end{aligned}$$

$$\begin{aligned}
&+ .105000000000\;10^{-10}\,\mathit{so} + .0500000000021\,\mathit{he} + .999999999999\,\mathit{\_SL13},\ \mathit{sfh} = \\
&.699999999999\,\mathit{\_SL13} + 2506.00000000 + 1.40000000001\,\mathit{so} - 2.23500000000\,\mathit{\_AR} \\
&+ 1.20000000000\,\mathit{\_SL12} + 61.2000000000\,\mathit{da} + .335000000002\,\mathit{he},\ \mathit{\_SL14} = \\
&449.999999998 - 1.00000000001\,\mathit{so} + 3.05000000000\,\mathit{\_AR} - 1.00000000000\,\mathit{\_SL12} \\
&- 3.00000000000\,\mathit{da} - .0500000000021\,\mathit{he} - 1.00000000000\,\mathit{\_SL13}, \\
&\mathit{co} = -1.00000000000\,\mathit{\_AR} + .100000000000\;10^{-22}\,\mathit{so} + \mathit{\_SL12} + 30.0000000000 + \mathit{da}, \\
&\mathit{wsh} = 1.00000000001\,\mathit{so} - .829999999997\,\mathit{\_AR} + .900000000001\,\mathit{\_SL12} \\
&+ 2487.00000000 + 60.9000000000\,\mathit{da} + .330000000001\,\mathit{he} + .599999999995\,\mathit{\_SL13}]
\end{aligned}$$

$$\mathit{obj}\ = ,\ -.1\;10^{-34}\,\mathit{so} - .999999999999\,\mathit{\_AR}$$

```
maximize:   eqns =
```

$$\begin{aligned}
&[\mathit{\_SL1} = 23582.0000002 + 57.3000000004\,\mathit{so} - 83.0574999999\,\mathit{\_AR} \\
&+ 48.9000000000\,\mathit{\_SL12} + 118.900000002\,\mathit{da} + 2.00750000007\,\mathit{he} \\
&+ 33.1499999999\,\mathit{\_SL13},\ \mathit{\_SL2} = 17332.0000001 - 22.7000000000\,\mathit{so} \\
&+ 33.9425000000\,\mathit{\_AR} - 26.1000000000\,\mathit{\_SL12} + 43.9000000020\,\mathit{da} \\
&+ .007499999997\,\mathit{he} - 6.84999999996\,\mathit{\_SL13},\ \mathit{\_SL3} = 19382.0000001 \\
&+ 2.30000000012\,\mathit{so} - 11.5575000001\,\mathit{\_AR} + 8.90000000006\,\mathit{\_SL12} \\
&+ 78.9000000021\,\mathit{da} + .507500000018\,\mathit{he} + 3.15000000003\,\mathit{\_SL13},\ \mathit{\_SL4} = \\
&21982.0000001 + 37.3000000003\,\mathit{so} - 52.5574999999\,\mathit{\_AR} + 28.9000000000\,\mathit{\_SL12} \\
&+ 98.9000000020\,\mathit{da} + 1.50750000005\,\mathit{he} + 23.1499999999\,\mathit{\_SL13},\ \mathit{\_SL5} = \\
&16182.0000001 - 27.7000000001\,\mathit{so} + 49.4425000001\,\mathit{\_AR} - 31.1000000000\,\mathit{\_SL12} \\
&+ 38.9000000020\,\mathit{da} - .492500000029\,\mathit{he} - 16.8500000001\,\mathit{\_SL13},\ \mathit{\_SL6} = \\
&18582.0000001 - 2.69999999994\,\mathit{so} + 3.69249999992\,\mathit{\_AR} - 1.09999999996\,\mathit{\_SL12} \\
&+ 68.9000000021\,\mathit{da} + .257500000007\,\mathit{he} - 1.84999999997\,\mathit{\_SL13},\ \mathit{\_SL7} = \\
&1512.99999999 - 1.00000000001\,\mathit{so} + 2.82999999999\,\mathit{\_AR} - .90000000000\,\mathit{\_SL12} \\
&- 60.9000000000\,\mathit{da} - .329999999998\,\mathit{he} - .60000000000\,\mathit{\_SL13},\ \mathit{\_SL8} = \\
&1993.99999999 - .70000000000\,\mathit{\_SL13} - 1.40000000001\,\mathit{so} + 3.23499999999\,\mathit{\_AR} \\
&- 1.20000000000\,\mathit{\_SL12} - 61.2000000000\,\mathit{da} - .334999999999\,\mathit{he},\ \mathit{\_SL9} = \\
&20000.0000000 - 1500.00000000\,\mathit{da} + .600000000005\;10^{-10}\,\mathit{so} + 3.99999999995\,\mathit{\_AR} \\
&- 2.99999999999\,\mathit{he} + .3\;10^{-10}\,\mathit{\_SL13}, \\
&\mathit{\_SL10} = 12.0000000000 - 1.00000000000\,\mathit{da} + .1\;10^{-34}\,\mathit{so} + .999999999999\,\mathit{\_AR},\ \mathit{\_SL11} \\
&= 3000.00000002 + 1.99999999999\,\mathit{\_AR} - .199999999999\;10^{-22}\,\mathit{so} + .1\;10^{-10}\,\mathit{\_SL12} \\
&+ .1\;10^{-10}\,\mathit{da} - 1.00000000000\,\mathit{he},\ \mathit{wh} = 100.000000002 - 1.05000000000\,\mathit{\_AR} \\
&+ .105000000000\;10^{-10}\,\mathit{so} + .0500000000021\,\mathit{he} + .999999999999\,\mathit{\_SL13},\ \mathit{sfh} = \\
&.699999999999\,\mathit{\_SL13} + 2506.00000000 + 1.40000000001\,\mathit{so} - 2.23500000000\,\mathit{\_AR} \\
&+ 1.20000000000\,\mathit{\_SL12} + 61.2000000000\,\mathit{da} + .335000000002\,\mathit{he},\ \mathit{\_SL14} = \\
&449.999999998 - 1.00000000001\,\mathit{so} + 3.05000000000\,\mathit{\_AR} - 1.00000000000\,\mathit{\_SL12} \\
&- 3.00000000000\,\mathit{da} - .0500000000021\,\mathit{he} - 1.00000000000\,\mathit{\_SL13}, \\
&\mathit{co} = -1.00000000000\,\mathit{\_AR} + .100000000000\;10^{-22}\,\mathit{so} + \mathit{\_SL12} + 30.0000000000 + \mathit{da}, \\
&\mathit{wsh} = 1.00000000001\,\mathit{so} - .829999999997\,\mathit{\_AR} + .900000000001\,\mathit{\_SL12} \\
&+ 2487.00000000 + 60.9000000000\,\mathit{da} + .330000000001\,\mathit{he} + .599999999995\,\mathit{\_SL13}]
\end{aligned}$$

```
maximize:   trans =
                                        {}

simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{aligned}
&\mathit{eqns}\ = ,[\mathit{\_SL1} = 23582.0000002 + 57.3000000004\,\mathit{so} + 48.9000000000\,\mathit{\_SL12} \\
&+ 118.900000002\,\mathit{da} + 2.00750000007\,\mathit{he} + 33.1499999999\,\mathit{\_SL13},\ \mathit{\_SL2} = \\
&17332.0000001 - 22.7000000000\,\mathit{so} - 26.1000000000\,\mathit{\_SL12} + 43.9000000020\,\mathit{da} \\
&+ .007499999997\,\mathit{he} - 6.84999999996\,\mathit{\_SL13},\ \mathit{\_SL3} = 19382.0000001 \\
&+ 2.30000000012\,\mathit{so} + 8.90000000006\,\mathit{\_SL12} + 78.9000000021\,\mathit{da} + .507500000018\,\mathit{he} \\
&+ 3.15000000003\,\mathit{\_SL13},\ \mathit{\_SL4} = 21982.0000001 + 37.3000000003\,\mathit{so} \\
&+ 28.9000000000\,\mathit{\_SL12} + 98.9000000020\,\mathit{da} + 1.50750000005\,\mathit{he}
\end{aligned}$$

$$+ 23.1499999999 \,\mathit{\_SL13}, \mathit{\_SL5} = 16182.0000001 - 27.7000000001 \,\mathit{so}$$
$$- 31.1000000000 \,\mathit{\_SL12} + 38.9000000020 \,\mathit{da} - .492500000029 \,\mathit{he}$$
$$- 16.8500000001 \,\mathit{\_SL13}, \mathit{\_SL6} = 18582.0000001 - 2.69999999994 \,\mathit{so}$$
$$- 1.09999999996 \,\mathit{\_SL12} + 68.9000000021 \,\mathit{da} + .257500000007 \,\mathit{he}$$
$$- 1.84999999997 \,\mathit{\_SL13}, \mathit{\_SL7} = 1512.99999999 - 1.00000000001 \,\mathit{so}$$
$$- .90000000000 \,\mathit{\_SL12} - 60.9000000000 \,\mathit{da} - .329999999998 \,\mathit{he}$$
$$- .60000000000 \,\mathit{\_SL13}, \mathit{\_SL8} = 1993.99999999 - .70000000000 \,\mathit{\_SL13}$$
$$- 1.40000000001 \,\mathit{so} - 1.20000000000 \,\mathit{\_SL12} - 61.2000000000 \,\mathit{da} - .334999999999 \,\mathit{he},$$
$$\mathit{\_SL9} = 20000.0000000 - 1500.00000000 \,\mathit{da} + .600000000005 \; 10^{-10} \,\mathit{so}$$
$$- 2.99999999999 \,\mathit{he} + .3 \; 10^{-10} \,\mathit{\_SL13},$$
$$\mathit{\_SL10} = 12.0000000000 - 1.00000000000 \,\mathit{da} + .1 \; 10^{-34} \,\mathit{so}, \mathit{\_SL11} = 3000.00000002$$
$$- .199999999999 \; 10^{-22} \,\mathit{so} + .1 \; 10^{-10} \,\mathit{\_SL12} + .1 \; 10^{-10} \,\mathit{da} - 1.00000000000 \,\mathit{he}, \mathit{wh} =$$
$$100.000000002 + .105000000000 \; 10^{-10} \,\mathit{so} + .0500000000021 \,\mathit{he} + .999999999999 \,\mathit{\_SL13}$$
$$, \mathit{sfh} = .699999999999 \,\mathit{\_SL13} + 2506.00000000 + 1.40000000001 \,\mathit{so}$$
$$+ 1.20000000000 \,\mathit{\_SL12} + 61.2000000000 \,\mathit{da} + .335000000002 \,\mathit{he}, \mathit{\_SL14} =$$
$$449.999999998 - 1.00000000001 \,\mathit{so} - 1.00000000000 \,\mathit{\_SL12} - 3.00000000000 \,\mathit{da}$$
$$- .0500000000021 \,\mathit{he} - 1.00000000000 \,\mathit{\_SL13},$$
$$\mathit{co} = 30.0000000000 + .100000000000 \; 10^{-22} \,\mathit{so} + \mathit{\_SL12} + \mathit{da}, \mathit{wsh} = 1.00000000001 \,\mathit{so}$$
$$+ 2487.00000000 + .900000000001 \,\mathit{\_SL12} + 60.9000000000 \,\mathit{da} + .330000000001 \,\mathit{he}$$
$$+ .599999999995 \,\mathit{\_SL13}]$$

$$\mathit{obj} \;=\; , 21.3000000001 \,\mathit{so} + 70682.0000000 + 16.4000000000 \,\mathit{\_SL12} + 86.400000000 \,\mathit{da}$$
$$+ 1.04500000002 \,\mathit{he} + 13.9000000000 \,\mathit{\_SL13}$$

$$\mathit{pivot\ variable} \;=\; , \mathit{\_SL13}$$

$$\mathit{pivot\ equation} \;=\; , \mathit{\_SL14} = 449.999999998 - 1.00000000001 \,\mathit{so} - 1.00000000000 \,\mathit{\_SL12}$$
$$- 3.00000000000 \,\mathit{da} - .0500000000021 \,\mathit{he} - 1.00000000000 \,\mathit{\_SL13}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\mathit{eqns} \;=\; , [\mathit{\_SL1} = 38499.5000001 + 24.1500000002 \,\mathit{so} + 15.7500000001 \,\mathit{\_SL12}$$
$$+ 19.4500000023 \,\mathit{da} + .35000000001 \,\mathit{he} - 33.1499999999 \,\mathit{\_SL14}, \mathit{\_SL2} =$$
$$14249.5000001 - 15.8500000000 \,\mathit{so} - 19.2500000000 \,\mathit{\_SL12} + 64.4500000019 \,\mathit{da}$$
$$+ .350000000009 \,\mathit{he} + 6.84999999996 \,\mathit{\_SL14}, \mathit{\_SL3} = 20799.5000001$$
$$- .84999999994 \,\mathit{so} + 5.75000000003 \,\mathit{\_SL12} + 69.4500000020 \,\mathit{da} + .350000000010 \,\mathit{he}$$
$$- 3.15000000003 \,\mathit{\_SL14}, \mathit{\_SL4} = 32399.5000000 + 14.1500000002 \,\mathit{so}$$
$$+ 5.7500000001 \,\mathit{\_SL12} + 29.4500000023 \,\mathit{da} + .35000000001 \,\mathit{he}$$
$$- 23.1499999999 \,\mathit{\_SL14}, \mathit{\_SL5} = 8599.50000009 - 10.8499999998 \,\mathit{so}$$
$$- 14.2499999999 \,\mathit{\_SL12} + 89.4500000023 \,\mathit{da} + .350000000011 \,\mathit{he}$$
$$+ 16.8500000001 \,\mathit{\_SL14}, \mathit{\_SL6} = 17749.5000001 - .84999999995 \,\mathit{so}$$
$$+ .75000000001 \,\mathit{\_SL12} + 74.4500000020 \,\mathit{da} + .350000000009 \,\mathit{he}$$
$$+ 1.84999999997 \,\mathit{\_SL14}, \mathit{\_SL7} = 1242.99999999 - .400000000004 \,\mathit{so}$$
$$- .30000000000 \,\mathit{\_SL12} - 59.1000000000 \,\mathit{da} - .299999999997 \,\mathit{he}$$
$$+ .60000000000 \,\mathit{\_SL14}, \mathit{\_SL8} = 1678.99999999 + .70000000000 \,\mathit{\_SL14}$$
$$- .700000000003 \,\mathit{so} - .50000000000 \,\mathit{\_SL12} - 59.1000000000 \,\mathit{da} - .299999999998 \,\mathit{he},$$
$$\mathit{\_SL9} = 20000.0000000 - 1500.00000000 \,\mathit{da} + .300000000002 \; 10^{-10} \,\mathit{so}$$
$$- 2.99999999999 \,\mathit{he} - .3 \; 10^{-10} \,\mathit{\_SL14} - .3 \; 10^{-10} \,\mathit{\_SL12},$$
$$\mathit{\_SL10} = 12.0000000000 - 1.00000000000 \,\mathit{da} + .1 \; 10^{-34} \,\mathit{so}, \mathit{\_SL11} = 3000.00000002$$
$$- .199999999999 \; 10^{-22} \,\mathit{so} + .1 \; 10^{-10} \,\mathit{\_SL12} + .1 \; 10^{-10} \,\mathit{da} - 1.00000000000 \,\mathit{he}, \mathit{wh} =$$
$$550.000000000 - 1.00000000000 \,\mathit{so} + .1 \; 10^{-12} \,\mathit{he} - .999999999999 \,\mathit{\_SL14}$$
$$- .999999999999 \,\mathit{\_SL12} - 3.00000000000 \,\mathit{da}, \mathit{sfh} = -.699999999999 \,\mathit{\_SL14}$$
$$+ 2821.00000000 + .700000000004 \,\mathit{so} + .500000000001 \,\mathit{\_SL12} + 59.1000000000 \,\mathit{da}$$
$$+ .300000000001 \,\mathit{he}, \mathit{\_SL13} = -1. \,\mathit{\_SL14} + 449.999999998 - 1.00000000001 \,\mathit{so}$$
$$- 1. \,\mathit{\_SL12} - 3. \,\mathit{da} - .0500000000021 \,\mathit{he},$$
$$\mathit{co} = 30.0000000000 + .100000000000 \; 10^{-22} \,\mathit{so} + \mathit{\_SL12} + \mathit{da}, \mathit{wsh} = .400000000009 \,\mathit{so}$$

$$\begin{gathered}
+\, 2757.00000000 + .300000000006\ \mathit{\_SL12} + 59.1000000000\ \mathit{da} + .300000000000\ \mathit{he} \\
-\, .599999999995\ \mathit{\_SL14}]
\end{gathered}$$

$$\begin{gathered}
\mathit{obj}\ =\, , 7.4000000000\ \mathit{so} + 76937.0000000 + 2.5000000000\ \mathit{\_SL12} + 44.7000000000\ \mathit{da} \\
+\, .349999999991\ \mathit{he} - 13.9000000000\ \mathit{\_SL14}
\end{gathered}$$

$$\mathit{pivot\ variable}\ =\, ,\ \mathit{da}$$

$$\mathit{pivot\ equation}\ =\, ,\ \mathit{\_SL10} = 12.0000000000 - 1.00000000000\ \mathit{da} + .1\ 10^{-34}\ \mathit{so}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{gathered}
\mathit{eqns}\ =\, , [\mathit{\_SL1} = 38732.9000001 + 24.1500000002\ \mathit{so} + 15.7500000001\ \mathit{\_SL12} \\
-\, 19.4500000023\ \mathit{\_SL10} + .35000000001\ \mathit{he} - 33.1499999999\ \mathit{\_SL14}, \mathit{\_SL2} = \\
15022.9000001 - 15.8500000000\ \mathit{so} - 19.2500000000\ \mathit{\_SL12} - 64.4500000019\ \mathit{\_SL10} \\
+\, .350000000009\ \mathit{he} + 6.84999999996\ \mathit{\_SL14}, \mathit{\_SL3} = 21632.9000001 \\
-\, .84999999994\ \mathit{so} + 5.75000000003\ \mathit{\_SL12} - 69.4500000020\ \mathit{\_SL10} \\
+\, .350000000010\ \mathit{he} - 3.15000000003\ \mathit{\_SL14}, \mathit{\_SL4} = 32752.9000000 \\
+\, 14.1500000002\ \mathit{so} + 5.7500000001\ \mathit{\_SL12} - 29.4500000023\ \mathit{\_SL10} \\
+\, .35000000001\ \mathit{he} - 23.1499999999\ \mathit{\_SL14}, \mathit{\_SL5} = 9672.90000012 \\
-\, 10.8499999998\ \mathit{so} - 14.2499999999\ \mathit{\_SL12} - 89.4500000023\ \mathit{\_SL10} \\
+\, .350000000011\ \mathit{he} + 16.8500000001\ \mathit{\_SL14}, \mathit{\_SL6} = 18642.9000001 \\
-\, .84999999995\ \mathit{so} + .75000000001\ \mathit{\_SL12} - 74.4500000020\ \mathit{\_SL10} \\
+\, .350000000009\ \mathit{he} + 1.84999999997\ \mathit{\_SL14}, \mathit{\_SL7} = 533.799999990 \\
-\, .400000000004\ \mathit{so} - .30000000000\ \mathit{\_SL12} + 59.1000000000\ \mathit{\_SL10} \\
-\, .299999999997\ \mathit{he} + .60000000000\ \mathit{\_SL14}, \mathit{\_SL8} = 969.799999990 \\
+\, .70000000000\ \mathit{\_SL14} - .700000000003\ \mathit{so} - .50000000000\ \mathit{\_SL12} \\
+\, 59.1000000000\ \mathit{\_SL10} - .299999999998\ \mathit{he}, \mathit{\_SL9} = 2000.0000000 \\
+\, 1500.00000000\ \mathit{\_SL10} + .300000000002\ 10^{-10}\ \mathit{so} - 2.99999999999\ \mathit{he} - .3\ 10^{-10}\ \mathit{\_SL14} \\
-\, .3\ 10^{-10}\ \mathit{\_SL12},\ \mathit{da} = -1.\ \mathit{\_SL10} + 12. + .100000000000\ 10^{-34}\ \mathit{so}, \mathit{\_SL11} = \\
3000.00000002 - .199999999999\ 10^{-22}\ \mathit{so} + .1\ 10^{-10}\ \mathit{\_SL12} - .1\ 10^{-10}\ \mathit{\_SL10} \\
-\, 1.00000000000\ \mathit{he}, \mathit{wh} = 514.000000000 - 1.00000000000\ \mathit{so} + .1\ 10^{-12}\ \mathit{he} \\
-\, .999999999999\ \mathit{\_SL14} - .999999999999\ \mathit{\_SL12} + 3.00000000000\ \mathit{\_SL10}, \mathit{sfh} = \\
-.699999999999\ \mathit{\_SL14} + 3530.20000000 + .700000000004\ \mathit{so} + .500000000001\ \mathit{\_SL12} \\
-\, 59.1000000000\ \mathit{\_SL10} + .300000000001\ \mathit{he}, \mathit{\_SL13} = -1.\ \mathit{\_SL14} + 413.999999998 \\
-\, 1.00000000001\ \mathit{so} - 1.\ \mathit{\_SL12} + 3.\ \mathit{\_SL10} - .0500000000021\ \mathit{he}, \\
\mathit{co} = 42.0000000000 + .100000000000\ 10^{-22}\ \mathit{so} + \mathit{\_SL12} - 1.\ \mathit{\_SL10}, \mathit{wsh} = \\
.400000000009\ \mathit{so} + 3466.20000000 + .300000000006\ \mathit{\_SL12} - 59.1000000000\ \mathit{\_SL10} \\
+\, .300000000000\ \mathit{he} - .599999999995\ \mathit{\_SL14}]
\end{gathered}$$

$$\begin{gathered}
\mathit{obj}\ =\, , 7.4000000000\ \mathit{so} + 77473.4000000 + 2.5000000000\ \mathit{\_SL12} \\
-\, 44.7000000000\ \mathit{\_SL10} + .349999999991\ \mathit{he} - 13.9000000000\ \mathit{\_SL14}
\end{gathered}$$

$$\mathit{pivot\ variable}\ =\, ,\ \mathit{he}$$

$$\begin{gathered}
\mathit{pivot\ equation}\ =\, , \mathit{\_SL9} = 2000.0000000 + 1500.00000000\ \mathit{\_SL10} \\
+\, .300000000002\ 10^{-10}\ \mathit{so} - 2.99999999999\ \mathit{he} - .3\ 10^{-10}\ \mathit{\_SL14} - .3\ 10^{-10}\ \mathit{\_SL12}
\end{gathered}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{gathered}
\mathit{eqns}\ =\, , [\mathit{\_SL1} = 38966.2333334 + 24.1500000002\ \mathit{so} + 15.7500000001\ \mathit{\_SL12} \\
+\, 155.550000004\ \mathit{\_SL10} - .116666666670\ \mathit{\_SL9} - 33.1499999999\ \mathit{\_SL14}, \mathit{\_SL2} = \\
15256.2333334 - 15.8500000000\ \mathit{so} - 19.2500000000\ \mathit{\_SL12} + 110.550000003\ \mathit{\_SL10} \\
-\, .116666666670\ \mathit{\_SL9} + 6.84999999996\ \mathit{\_SL14}, \mathit{\_SL3} = 21866.2333334 \\
-\, .849999999936\ \mathit{so} + 5.75000000003\ \mathit{\_SL12} + 105.550000004\ \mathit{\_SL10} \\
-\, .116666666670\ \mathit{\_SL9} - 3.15000000003\ \mathit{\_SL14}, \mathit{\_SL4} = 32986.2333333 \\
+\, 14.1500000002\ \mathit{so} + 5.75000000010\ \mathit{\_SL12} + 145.550000004\ \mathit{\_SL10}
\end{gathered}$$

$$\begin{gathered}
- .116666666670\ \mathit{\_SL9} - 23.1499999999\ \mathit{\_SL14}, \mathit{\_SL5} = 9906.23333346 \\
- 10.8499999998\ \mathit{so} - 14.2499999999\ \mathit{\_SL12} + 85.5500000037\ \mathit{\_SL10} \\
- .116666666671\ \mathit{\_SL9} + 16.8500000001\ \mathit{\_SL14}, \mathit{\_SL6} = 18876.2333334 \\
- .849999999946\ \mathit{so} + .750000000006\ \mathit{\_SL12} + 100.550000003\ \mathit{\_SL10} \\
- .116666666670\ \mathit{\_SL9} + 1.84999999997\ \mathit{\_SL14}, \mathit{\_SL7} = 333.799999991 \\
- .400000000007\ \mathit{so} - .299999999997\ \mathit{\_SL12} - 90.8999999990\ \mathit{\_SL10} \\
+ .0999999999992\ \mathit{\_SL9} + .600000000003\ \mathit{\_SL14}, \mathit{\_SL8} = 769.799999991 \\
+ .700000000003\ \mathit{\_SL14} - .700000000006\ \mathit{so} - .499999999997\ \mathit{\_SL12} \\
- 90.9000000000\ \mathit{\_SL10} + .0999999999995\ \mathit{\_SL9}, \mathit{he} = -.333333333334\ \mathit{\_SL9} \\
+ 666.666666669 + 500.000000002\ \mathit{\_SL10} + .100000000001\ 10^{-10}\ \mathit{so} \\
- .100000000000\ 10^{-10}\ \mathit{\_SL14} - .100000000000\ 10^{-10}\ \mathit{\_SL12}, \\
\mathit{da} = -1.\ \mathit{\_SL10} + 12. + .100000000000\ 10^{-34}\ \mathit{so}, \mathit{\_SL11} = 2333.33333335 \\
- .100000000001\ 10^{-10}\ \mathit{so} + .200000000000\ 10^{-10}\ \mathit{\_SL12} - 500.000000002\ \mathit{\_SL10} \\
+ .333333333334\ \mathit{\_SL9} + .100000000000\ 10^{-10}\ \mathit{\_SL14}, \mathit{wh} = 514.000000000 \\
- 1.00000000000\ \mathit{so} - .333333333334\ 10^{-13}\ \mathit{\_SL9} + 3.00000000005\ \mathit{\_SL10} \\
- .999999999999\ \mathit{\_SL14} - .999999999999\ \mathit{\_SL12}, \mathit{sfh} = -.700000000002\ \mathit{\_SL14} \\
+ 3730.20000000 + .700000000007\ \mathit{so} + .499999999998\ \mathit{\_SL12} + 90.9000000010\ \mathit{\_SL10} \\
- .100000000001\ \mathit{\_SL9}, \mathit{\_SL13} = -.999999999999\ \mathit{\_SL14} + 380.666666663 \\
- 1.00000000001\ \mathit{so} - .999999999999\ \mathit{\_SL12} - 22.0000000012\ \mathit{\_SL10} \\
+ .0166666666674\ \mathit{\_SL9}, \\
\mathit{co} = 42.0000000000 + .100000000000\ 10^{-22}\ \mathit{so} + \mathit{\_SL12} - 1.\ \mathit{\_SL10}, \mathit{wsh} = \\
.400000000012\ \mathit{so} + 3666.20000000 + .300000000003\ \mathit{\_SL12} + 90.9000000010\ \mathit{\_SL10} \\
- .100000000000\ \mathit{\_SL9} - .599999999998\ \mathit{\_SL14}]
\end{gathered}$$

$$\begin{gathered}
\mathit{obj} \ = , 7.40000000000\ \mathit{so} + 77706.7333333 + 2.50000000000\ \mathit{\_SL12} \\
+ 130.299999996\ \mathit{\_SL10} - .116666666664\ \mathit{\_SL9} - 13.9000000000\ \mathit{\_SL14} \\
\mathit{pivot\ variable} \ = , \mathit{\_SL10}
\end{gathered}$$

$$\begin{gathered}
\mathit{pivot\ equation} \ = , \mathit{\_SL7} = 333.799999991 - .400000000007\ \mathit{so} - .299999999997\ \mathit{\_SL12} \\
- 90.8999999990\ \mathit{\_SL10} + .0999999999992\ \mathit{\_SL9} + .600000000003\ \mathit{\_SL14}
\end{gathered}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{gathered}
\mathit{eqns} \ = , [\mathit{\_SL1} = 39537.4389440 + 23.4655115513\ \mathit{so} + 15.2366336635\ \mathit{\_SL12} \\
- 1.71122112217\ \mathit{\_SL7} + .054455445545\ \mathit{\_SL9} - 32.1232673266\ \mathit{\_SL14}, \mathit{\_SL2} = \\
15662.1914192 - 16.3364686469\ \mathit{so} - 19.6148514852\ \mathit{\_SL12} - 1.21617161720\ \mathit{\_SL7} \\
+ .004950495049\ \mathit{\_SL9} + 7.57970297029\ \mathit{\_SL14}, \mathit{\_SL3} = 22253.8305831 \\
- 1.31446644661\ \mathit{so} + 5.40165016503\ \mathit{\_SL12} - 1.16116611667\ \mathit{\_SL7} \\
- .000550055005\ \mathit{\_SL9} - 2.45330033003\ \mathit{\_SL14}, \mathit{\_SL4} = 33520.7172717 \\
+ 13.5095159518\ \mathit{so} + 5.26963696378\ \mathit{\_SL12} - 1.60121012107\ \mathit{\_SL7} \\
+ .043454345435\ \mathit{\_SL9} - 22.1892739273\ \mathit{\_SL14}, \mathit{\_SL5} = 10220.3872389 \\
- 11.2264576456\ \mathit{so} - 14.5323432342\ \mathit{\_SL12} - .941144114460\ \mathit{\_SL7} \\
- .0225522552259\ \mathit{\_SL9} + 17.4146864688\ \mathit{\_SL14}, \mathit{\_SL6} = 19245.4697470 \\
- 1.29246424640\ \mathit{so} + .418151815177\ \mathit{\_SL12} - 1.10616061610\ \mathit{\_SL7} \\
- .006050605061\ \mathit{\_SL9} + 2.51369636964\ \mathit{\_SL14}, \mathit{\_SL10} = -.0110011001101\ \mathit{\_SL7} \\
+ 3.67216721666 - .00440044004413\ \mathit{so} - .00330033003301\ \mathit{\_SL12} \\
+ .00110011001100\ \mathit{\_SL9} + .00660066006611\ \mathit{\_SL14}, \mathit{\_SL8} = 435.999999997 \\
+ .099999999994\ \mathit{\_SL14} - .299999999995\ \mathit{so} - .199999999996\ \mathit{\_SL12} \\
+ 1.00000000001\ \mathit{\_SL7} - .4\ 10^{-12}\ \mathit{\_SL9}, \mathit{he} = .216721672168\ \mathit{\_SL9} + 2502.75027501 \\
- 5.50055005507\ \mathit{\_SL7} - 2.20022002206\ \mathit{so} - 1.65016501652\ \mathit{\_SL12} \\
+ 3.30033003306\ \mathit{\_SL14}, \mathit{da} = .0110011001101\ \mathit{\_SL7} + 8.32783278334 \\
+ .00440044004413\ \mathit{so} + .00330033003301\ \mathit{\_SL12} - .00110011001100\ \mathit{\_SL9} \\
- .00660066006611\ \mathit{\_SL14}, \mathit{\_SL11} = 497.24972501 + 2.20022002206\ \mathit{so} \\
+ 1.65016501653\ \mathit{\_SL12} + 5.50055005507\ \mathit{\_SL7} - .216721672168\ \mathit{\_SL9} \\
- 3.30033003306\ \mathit{\_SL14}, \mathit{wh} = 525.016501650 - 1.01320132013\ \mathit{so}
\end{gathered}$$

$$\begin{gathered}
+ .00330033003303\ \_SL9 - .0330033003309\ \_SL7 - 1.00990099010\ \_SL12 \\
- .980198019800\ \_SL14, sfh = -.099999999986\ \_SL14 + 4064.00000000 \\
+ .299999999991\ so + .199999999994\ \_SL12 - 1.00000000002\ \_SL7, \_SL13 = \\
-1.14521452146\ \_SL14 + 299.878987892 - .903190319034\ so - .927392739269\ \_SL12 \\
+ .242024202435\ \_SL7 - .0075357535759\ \_SL9, co = 38.3278327833 \\
+ .00440044004413\ so + 1.00330033003\ \_SL12 + .0110011001101\ \_SL7 \\
- .00110011001100\ \_SL9 - .00660066006611\ \_SL14, wsh = -.4\,10^{-11}\ so \\
+ 4000.00000000 - .1\,10^{-11}\ \_SL12 - 1.00000000002\ \_SL7 + .1\,10^{-11}\ \_SL9 \\
+ .18\,10^{-10}\ \_SL14]
\end{gathered}$$

$$\begin{gathered}
obj\ = , 6.82662266227\ so + 78185.2167216 + 2.06996699671\ \_SL12 \\
- 1.43344334430\ \_SL7 + .026677667765\ \_SL9 - 13.0399339934\ \_SL14
\end{gathered}$$

$$pivot\ variable\ = , \_SL9$$

$$\begin{gathered}
pivot\ equation\ = , \_SL11 = 497.24972501 + 2.20022002206\ so + 1.65016501653\ \_SL12 \\
+ 5.50055005507\ \_SL7 - .216721672168\ \_SL9 - 3.30033003306\ \_SL14
\end{gathered}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$\begin{gathered}
eqns\ = , [\_SL1 = 39662.3824028 + 24.0183587142\ so + 15.6512690357\ \_SL12 \\
- .32910321493\ \_SL7 - .251269035534\ \_SL11 - 32.9525380709\ \_SL14, \_SL2 = \\
15673.5499155 - 16.2862098139\ so - 19.5771573605\ \_SL12 - 1.09052453474\ \_SL7 \\
- .0228426395915\ \_SL11 + 7.50431472081\ \_SL14, \_SL3 = 22252.5685280 \\
- 1.32005076138\ so + 5.39746192895\ \_SL12 - 1.17512690360\ \_SL7 \\
+ .00253807106367\ \_SL11 - 2.44492385787\ \_SL14, \_SL4 = 33620.4196277 \\
+ 13.9506768192\ so + 5.60050761431\ \_SL12 - .49830795266\ \_SL7 \\
- .200507614214\ \_SL11 - 22.8510152283\ \_SL14, \_SL5 = 10168.6429782 \\
- 11.4554145515\ so - 14.7040609136\ \_SL12 - 1.51353637908\ \_SL7 \\
+ .104060913707\ \_SL11 + 17.7581218276\ \_SL14, \_SL6 = 19231.5871405 \\
- 1.35389170895\ so + .372081218265\ \_SL12 - 1.25972927247\ \_SL7 \\
+ .0279187817280\ \_SL11 + 2.60583756346\ \_SL14, \_SL10 = .0169204737734\ \_SL7 \\
+ 6.19627749589 + .00676818950947\ so + .00507614213208\ \_SL12 \\
- .00507614213195\ \_SL11 - .0101522842640\ \_SL14, \_SL8 = 435.999999996 \\
+ .100000000000\ \_SL14 - .299999999999\ so - .199999999999\ \_SL12 \\
+ 1.00000000000\ \_SL7 + .184568527918\,10^{-11}\ \_SL11, \\
he = -.999999999999\ \_SL11 + 3000.00000002 + .1\,10^{-10}\ \_SL12 + .1\,10^{-10}\ \_SL14, da = \\
-.0169204737734\ \_SL7 + 5.80372250411 - .00676818950947\ so \\
- .00507614213208\ \_SL12 + .00507614213195\ \_SL11 + .0101522842640\ \_SL14, \_SL9 \\
= -4.61421319795\ \_SL11 + 2294.41624382 + 10.1522842642\ so \\
+ 7.61421319807\ \_SL12 + 25.3807106601\ \_SL7 - 15.2284263961\ \_SL14, wh = \\
532.588832488 - .979695431469\ so - .0152284263960\ \_SL11 - .984771573605\ \_SL12 \\
+ .0507614213203\ \_SL7 - 1.03045685279\ \_SL14, sfh = -.099999999986\ \_SL14 \\
+ 4064.00000000 + .299999999991\ so + .199999999994\ \_SL12 - 1.00000000002\ \_SL7, \\
\_SL13 = -1.03045685279\ \_SL14 + 282.588832478 - .979695431482\ so \\
- .984771573604\ \_SL12 + .050761421319\ \_SL7 + .0347715736064\ \_SL11, co = \\
35.8037225041 - .00676818950947\ so + .994923857865\ \_SL12 - .0169204737734\ \_SL7 \\
+ .00507614213195\ \_SL11 + .0101522842640\ \_SL14, wsh = .61522842642\,10^{-11}\ so \\
+ 4000.00000000 + .661421319807\,10^{-11}\ \_SL12 - .999999999995\ \_SL7 \\
- .461421319795\,10^{-11}\ \_SL11 + .27715736039\,10^{-11}\ \_SL14]
\end{gathered}$$

$$\begin{gathered}
obj\ = , 7.09746192893\ so + 78246.4263959 + 2.27309644670\ \_SL12 \\
- .756345177670\ \_SL7 - .123096446692\ \_SL11 - 13.4461928934\ \_SL14
\end{gathered}$$

$$pivot\ variable\ = , so$$

$$\begin{gathered}
pivot\ equation\ = , \_SL13 = -1.03045685279\ \_SL14 + 282.588832478 \\
- .979695431482\ so - .984771573604\ \_SL12 + .050761421319\ \_SL7 \\
+ .0347715736064\ \_SL11
\end{gathered}$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$
\begin{aligned}
&\mathit{eqns} \;=\; ,[\_\mathit{SL1} = 46590.3721933 - 24.5161485318\,\_\mathit{SL13} - 58.2153713296\,\_\mathit{SL14}\\
&\quad - 8.4915371327\,\_\mathit{SL12} + .91537132982\,\_\mathit{SL7} + .601196027687\,\_\mathit{SL11}, \_\mathit{SL2} =\\
&\quad 10975.8644217 + 16.6237478410\,\_\mathit{SL13} + 24.6343696026\,\_\mathit{SL14}\\
&\quad - 3.2065630400\,\_\mathit{SL12} - 1.93436960280\,\_\mathit{SL7} - .600876511258\,\_\mathit{SL11}, \_\mathit{SL3} =\\
&\quad 21871.8056996 + 1.34740932637\,\_\mathit{SL13} - 1.05647668400\,\_\mathit{SL14}\\
&\quad + 6.72435233157\,\_\mathit{SL12} - 1.24352331611\,\_\mathit{SL7} - .0443134715061\,\_\mathit{SL11}, \_\mathit{SL4} =\\
&\quad 37644.4309152 - 14.2398100174\,\_\mathit{SL13} - 37.5245250431\,\_\mathit{SL14}\\
&\quad - 8.42245250429\,\_\mathit{SL12} + .224525043135\,\_\mathit{SL7} + .294632987946\,\_\mathit{SL11}, \_\mathit{SL5} =\\
&\quad 6864.37910228 + 11.6928324695\,\_\mathit{SL13} + 29.8070811744\,\_\mathit{SL14}\\
&\quad - 3.1892918827\,\_\mathit{SL12} - 2.10708117448\,\_\mathit{SL7} - .302517271175\,\_\mathit{SL11}, \_\mathit{SL6} =\\
&\quad 18841.0630398 + 1.38195164073\,\_\mathit{SL13} + 4.02987910187\,\_\mathit{SL14}\\
&\quad + 1.73298791015\,\_\mathit{SL12} - 1.32987910195\,\_\mathit{SL7} - .0201338514680\,\_\mathit{SL11}, \_\mathit{SL10} =\\
&\quad .0172711571677\,\_\mathit{SL7} + 8.14853195172 - .00690846286710\,\_\mathit{SL13}\\
&\quad - .0172711571677\,\_\mathit{SL14} - .00172711571675\,\_\mathit{SL12} - .00483592400686\,\_\mathit{SL11}, \_\mathit{SL8}\\
&\quad = 349.466321244 + .415544041446\,\_\mathit{SL14} + .306217616576\,\_\mathit{SL13}\\
&\quad + .101554404142\,\_\mathit{SL12} + .984455958550\,\_\mathit{SL7} - .0106476683926\,\_\mathit{SL11},\\
&\quad \mathit{he} = -.999999999999\,\_\mathit{SL11} + 3000.00000002 + .1\,10^{-10}\,\_\mathit{SL12} + .1\,10^{-10}\,\_\mathit{SL14}, \mathit{da} =\\
&\quad -.0172711571677\,\_\mathit{SL7} + 3.85146804828 + .00690846286710\,\_\mathit{SL13}\\
&\quad + .0172711571677\,\_\mathit{SL14} + .00172711571675\,\_\mathit{SL12} + .00483592400686\,\_\mathit{SL11}, \_\mathit{SL9}\\
&\quad = -4.25388601031\,\_\mathit{SL11} + 5222.79792756 - 10.3626943007\,\_\mathit{SL13}\\
&\quad - 25.9067357516\,\_\mathit{SL14} - 2.59067357513\,\_\mathit{SL12} + 25.9067357515\,\_\mathit{SL7}, \mathit{wh} =\\
&\quad 250.000000014 + .999999999986\,\_\mathit{SL13} - .1\,10^{-10}\,\_\mathit{SL14} - .14\,10^{-10}\,\_\mathit{SL12}\\
&\quad + .19\,10^{-11}\,\_\mathit{SL7} - .0500000000019\,\_\mathit{SL11}, \mathit{sfh} = -.415544041424\,\_\mathit{SL14}\\
&\quad + 4150.53367875 - .306217616568\,\_\mathit{SL13} - .101554404139\,\_\mathit{SL12}\\
&\quad - .984455958570\,\_\mathit{SL7} + .0106476683941\,\_\mathit{SL11}, \mathit{so} = -1.02072538859\,\_\mathit{SL13}\\
&\quad - 1.05181347149\,\_\mathit{SL14} + 288.445595842 - 1.00518134714\,\_\mathit{SL12}\\
&\quad + .0518134715013\,\_\mathit{SL7} + .0354922279813\,\_\mathit{SL11}, \mathit{co} = 33.8514680483\\
&\quad + .00690846286710\,\_\mathit{SL13} + .0172711571677\,\_\mathit{SL14} + 1.00172711571\,\_\mathit{SL12}\\
&\quad - .0172711571677\,\_\mathit{SL7} + .00483592400686\,\_\mathit{SL11}, \mathit{wsh} =\\
&\quad -.627979274629\,10^{-11}\,\_\mathit{SL13} - .369948186562\,10^{-11}\,\_\mathit{SL14} + 4000.00000000\\
&\quad + .43005181339\,10^{-12}\,\_\mathit{SL12} - .999999999995\,\_\mathit{SL7} - .439585492224\,10^{-11}\,\_\mathit{SL11}]
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{obj} \;=\; ,-7.24455958541\,\_\mathit{SL13} - 20.9113989636\,\_\mathit{SL14} + 80293.6580310\\
&\quad - 4.86113989630\,\_\mathit{SL12} - .388601036284\,\_\mathit{SL7} + .128808290178\,\_\mathit{SL11}
\end{aligned}
$$

$$\mathit{pivot\ variable} \;=\; ,\ \_\mathit{SL11}$$

$$
\begin{aligned}
&\mathit{pivot\ equation} \;=\; ,\_\mathit{SL9} = -4.25388601031\,\_\mathit{SL11} + 5222.79792756\\
&\quad - 10.3626943007\,\_\mathit{SL13} - 25.9067357516\,\_\mathit{SL14} - 2.59067357513\,\_\mathit{SL12}\\
&\quad + 25.9067357515\,\_\mathit{SL7}
\end{aligned}
$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$
\begin{aligned}
&\mathit{eqns} \;=\; ,[\_\mathit{SL1} = 47328.5032480 - 25.9806942753\,\_\mathit{SL13} - 61.8767356883\,\_\mathit{SL14}\\
&\quad - 8.85767356857\,\_\mathit{SL12} + 4.57673568852\,\_\mathit{SL7} - .141328664245\,\_\mathit{SL9}, \_\mathit{SL2} =\\
&\quad 10238.1256600 + 18.0875152254\,\_\mathit{SL13} + 28.2937880635\,\_\mathit{SL14}\\
&\quad - 2.84062119392\,\_\mathit{SL12} - 5.59378806364\,\_\mathit{SL7} + .141253552588\,\_\mathit{SL9}, \_\mathit{SL3} =\\
&\quad 21817.3989039 + 1.45535931786\,\_\mathit{SL13} - .786601705272\,\_\mathit{SL14}\\
&\quad + 6.75133982944\,\_\mathit{SL12} - 1.51339829484\,\_\mathit{SL7} + .0104171741788\,\_\mathit{SL9}, \_\mathit{SL4} =\\
&\quad 38006.1727567 - 14.9575517664\,\_\mathit{SL13} - 39.3188794155\,\_\mathit{SL14}\\
&\quad - 8.60188794153\,\_\mathit{SL12} + 2.01887941556\,\_\mathit{SL7} - .0692620787751\,\_\mathit{SL9}, \_\mathit{SL5} =\\
&\quad 6492.95716641 + 12.4297807550\,\_\mathit{SL13} + 31.6494518881\,\_\mathit{SL14}
\end{aligned}
$$

$$
\begin{aligned}
&- 3.00505481134\ \_SL12 - 3.94945188813\ \_SL7 + .0711155095464\ \_SL9, \_SL6 = \\
&18816.3432806 + 1.43099878194\ \_SL13 + 4.15249695491\ \_SL14 \\
&+ 1.74524969545\ \_SL12 - 1.45249695499\ \_SL7 + .00473304912713\ \_SL9, \_SL10 = \\
&-.0121802679660\ \_SL7 + 2.21112464470 + .00487210718650\ \_SL13 \\
&+ .0121802679661\ \_SL14 + .00121802679660\ \_SL12 + .00113682501015\ \_SL9, \_SL8 = \\
&336.393422657 + .480389768564\ \_SL14 + .332155907424\ \_SL13 \\
&+ .108038976854\ \_SL12 + .919610231432\ \_SL7 + .00250304506675\ \_SL9, he = \\
&.235079171745\ \_SL9 + 1772.22898902 + 2.43605359325\ \_SL13 \\
&+ 6.09013398309\ \_SL14 + .609013398311\ \_SL12 - 6.09013398306\ \_SL7, da = \\
&.0121802679660\ \_SL7 + 9.78887535530 - .00487210718650\ \_SL13 \\
&- .0121802679661\ \_SL14 - .00121802679660\ \_SL12 - .00113682501015\ \_SL9, \_SL11 \\
&= -.235079171745\ \_SL9 + 1227.77101100 - 2.43605359325\ \_SL13 \\
&- 6.09013398309\ \_SL14 - .609013398302\ \_SL12 + 6.09013398307\ \_SL7, wh = \\
&188.611449462 + 1.12180267965\ \_SL13 + .304506699156\ \_SL14 \\
&+ .0304506699023\ \_SL12 - .304506699163\ \_SL7 + .0117539585877\ \_SL9, sfh = \\
&-.480389768552\ \_SL14 + 4163.60657734 - .332155907419\ \_SL13 \\
&- .108038976852\ \_SL12 - .919610231443\ \_SL7 - .00250304506710\ \_SL9, so = \\
&-1.10718635810\ \_SL13 - 1.26796589525\ \_SL14 + 332.021924473 \\
&- 1.02679658952\ \_SL12 + .267965895265\ \_SL7 - .00834348355723\ \_SL9, co = \\
&39.7888753553 - .00487210718650\ \_SL13 - .0121802679661\ \_SL14 \\
&+ .998781973197\ \_SL12 + .0121802679660\ \_SL7 - .00113682501015\ \_SL9, wsh = \\
&.442874543241\ 10^{-11}\ \_SL13 + .230718635811\ 10^{-10}\ \_SL14 + 3999.99999999 \\
&+ .310718635803\ 10^{-11}\ \_SL12 - 1.00000000002\ \_SL7 + .103337393423\ 10^{-11}\ \_SL9]
\end{aligned}
$$

$$
\begin{aligned}
obj\ =\ , &-7.55834348354\ \_SL13 - 21.6958587089\ \_SL14 + 80451.8051157 \\
&- 4.93958587083\ \_SL12 + .395858709030\ \_SL7 - .0302801461689\ \_SL9
\end{aligned}
$$

$$pivot\ variable\ =\ ,\ \_SL7$$

$$
\begin{aligned}
pivot\ equation\ =\ ,\ \_SL10 = &-.0121802679660\ \_SL7 + 2.21112464470 \\
&+ .00487210718650\ \_SL13 + .0121802679661\ \_SL14 + .00121802679660\ \_SL12 \\
&+ .00113682501015\ \_SL9
\end{aligned}
$$

```
simplex/max:   \#\#\#\#\#\#   [\_AR, \_SL6, \_SL14, \_SL1, \_SL2,
\_SL5, \_SL9, \_SL10, \_SL11, \_SL13, co, wh, da, he, wsh, sfh, so,
\_SL12, \_SL3, \_SL7, \_SL4, \_SL8]   \#\#\#\#\#\#
```

$$
\begin{aligned}
eqns\ =\ ,[\_SL1 = {}&48159.3333333 - 24.1499999999\ \_SL13 - 57.2999999997\ \_SL14 \\
&- 8.39999999972\ \_SL12 - 375.750000024\ \_SL10 + .285833333346\ \_SL9, \_SL2 = \\
&9222.66666688 + 15.8499999999\ \_SL13 + 22.6999999998\ \_SL14 \\
&- 3.40000000028\ \_SL12 + 459.250000021\ \_SL10 - .380833333347\ \_SL9, \_SL3 = \\
&21542.6666668 + .849999999912\ \_SL13 - 2.30000000013\ \_SL14 \\
&+ 6.59999999996\ \_SL12 + 124.250000005\ \_SL10 - .130833333338\ \_SL9, \_SL4 = \\
&38372.6666666 - 14.1500000002\ \_SL13 - 37.2999999999\ \_SL14 \\
&- 8.39999999997\ \_SL12 - 165.750000016\ \_SL10 + .119166666676\ \_SL9, \_SL5 = \\
&5776.00000034 + 10.8499999997\ \_SL13 + 27.6999999999\ \_SL14 \\
&- 3.40000000015\ \_SL12 + 324.250000013\ \_SL10 - .297500000010\ \_SL9, \_SL6 = \\
&18552.6666667 + .849999999932\ \_SL13 + 2.69999999991\ \_SL14 \\
&+ 1.59999999995\ \_SL12 + 119.250000004\ \_SL10 - .130833333338\ \_SL9, \_SL7 = \\
&-82.0999999993\ \_SL10 + 181.533333328 + .400000000008\ \_SL13 \\
&+ 1.00000000001\ \_SL14 + .100000000000\ \_SL12 + .0933333333325\ \_SL9, \_SL8 = \\
&503.333333331 + 1.40000000001\ \_SL14 + .700000000004\ \_SL13 \\
&+ .199999999997\ \_SL12 - 75.4999999999\ \_SL10 + .0883333333330\ \_SL9, he = \\
&-.333333333336\ \_SL9 + 666.66666666 - .2\ 10^{-10}\ \_SL13 - .3\ 10^{-10}\ \_SL14 \\
&+ .5\ 10^{-11}\ \_SL12 + 500.000000005\ \_SL10, \\
&da = -1.00000000000\ \_SL10 + 12.0000000000, \_SL11 = .333333333336\ \_SL9 \\
&+ 2333.33333336 + .3\ 10^{-10}\ \_SL13 + .4\ 10^{-10}\ \_SL14 + .5\ 10^{-11}\ \_SL12 \\
&- 500.000000006\ \_SL10, wh = 133.333333342 + .999999999982\ \_SL13 \\
&- .10\ 10^{-10}\ \_SL14 - .140\ 10^{-10}\ \_SL12 + 25.0000000011\ \_SL10 - .0166666666673\ \_SL9,
\end{aligned}
$$

$$\begin{aligned}
&\mathit{sfh} = -1.40000000000\ \_\mathit{SL14} + 3996.66666666 - .700000000004\ \_\mathit{SL13}\\
&- .199999999996\ \_\mathit{SL12} + 75.5000000008\ \_\mathit{SL10} - .0883333333343\ \_\mathit{SL9}, \mathit{so} =\\
&-.999999999992\ \_\mathit{SL13} - .999999999982\ \_\mathit{SL14} + 380.666666659\\
&- .999999999994\ \_\mathit{SL12} - 22.0000000011\ \_\mathit{SL10} + .0166666666673\ \_\mathit{SL9},\\
&\mathit{co} = 42.0000000000 + .999999999994\ \_\mathit{SL12} - 1.00000000000\ \_\mathit{SL10}, \mathit{wsh} =\\
&-.400000000012\ \_\mathit{SL13} - 1.00000000001\ \_\mathit{SL14} + 3818.46666666\\
&- .0999999999989\ \_\mathit{SL12} + 82.1000000009\ \_\mathit{SL10} - .0933333333334\ \_\mathit{SL9}]
\end{aligned}$$

$$\begin{aligned}
\mathit{obj}\ = {}&, -7.39999999993\ \_\mathit{SL13} - 21.2999999999\ \_\mathit{SL14} + 80523.6666667\\
&- 4.89999999993\ \_\mathit{SL12} - 32.5000000111\ \_\mathit{SL10} + .0066666666736\ \_\mathit{SL9}
\end{aligned}$$

$$\mathit{pivot\ variable}\ = ,\ \_\mathit{SL9}$$

$$\begin{aligned}
\mathit{pivot\ equation}\ = {}&, \mathit{he} = -.333333333336\ \_\mathit{SL9} + 666.66666666 - .2\,10^{-10}\ \_\mathit{SL13}\\
&- .3\,10^{-10}\ \_\mathit{SL14} + .5\,10^{-11}\ \_\mathit{SL12} + 500.000000005\ \_\mathit{SL10}
\end{aligned}$$

```
simplex/max:   \#\#\#\#\#\# final description \#\#\#\#\#\#
```

$$\begin{aligned}
\mathit{eqns}\ = {}&, [\_\mathit{SL1} = 48731.0000000 - 24.1499999999\ \_\mathit{SL13} - 57.2999999997\ \_\mathit{SL14}\\
&- 8.39999999972\ \_\mathit{SL12} + 52.999999995\ \_\mathit{SL10} - .857500000032\ \mathit{he}, \_\mathit{SL2} =\\
&8461.00000020 + 15.8499999999\ \_\mathit{SL13} + 22.6999999998\ \_\mathit{SL14}\\
&- 3.40000000029\ \_\mathit{SL12} - 112.000000000\ \_\mathit{SL10} + 1.14250000003\ \mathit{he}, \_\mathit{SL3} =\\
&21281.0000001 + .849999999920\ \_\mathit{SL13} - 2.30000000012\ \_\mathit{SL14}\\
&+ 6.59999999996\ \_\mathit{SL12} - 72.000000002\ \_\mathit{SL10} + .392500000011\ \mathit{he}, \_\mathit{SL4} =\\
&38610.9999999 - 14.1500000002\ \_\mathit{SL13} - 37.2999999999\ \_\mathit{SL14}\\
&- 8.39999999997\ \_\mathit{SL12} + 12.999999998\ \_\mathit{SL10} - .357500000026\ \mathit{he}, \_\mathit{SL5} =\\
&5181.00000033 + 10.8499999997\ \_\mathit{SL13} + 27.6999999999\ \_\mathit{SL14}\\
&- 3.40000000015\ \_\mathit{SL12} - 122.000000002\ \_\mathit{SL10} + .892500000024\ \mathit{he}, \_\mathit{SL6} =\\
&18291.0000000 + .849999999940\ \_\mathit{SL13} + 2.69999999992\ \_\mathit{SL14}\\
&+ 1.59999999995\ \_\mathit{SL12} - 77.000000003\ \_\mathit{SL10} + .392500000011\ \mathit{he}, \_\mathit{SL7} =\\
&57.8999999997\ \_\mathit{SL10} + 368.199999989 + .400000000002\ \_\mathit{SL13}\\
&+ 1.00000000000\ \_\mathit{SL14} + .100000000001\ \_\mathit{SL12} - .279999999996\ \mathit{he}, \_\mathit{SL8} =\\
&679.999999993 + 1.40000000000\ \_\mathit{SL14} + .699999999999\ \_\mathit{SL13}\\
&+ .199999999998\ \_\mathit{SL12} + 57.0000000001\ \_\mathit{SL10} - .264999999997\ \mathit{he}, \_\mathit{SL9} =\\
&-2.99999999998\ \mathit{he} + 1999.99999996 - .599999999995\,10^{-10}\ \_\mathit{SL13}\\
&- .899999999993\,10^{-10}\ \_\mathit{SL14} + .149999999999\,10^{-10}\ \_\mathit{SL12} + 1500.00000000\ \_\mathit{SL10},\\
&\mathit{da} = -1.00000000000\ \_\mathit{SL10} + 12.0000000000, \_\mathit{SL11}\ = -1.00000000000\ \mathit{he}\\
&+ 3000.00000002 + .100000000000\ 10^{-10}\ \_\mathit{SL13} + .100000000000\ 10^{-10}\ \_\mathit{SL14}\\
&+ .100000000000\ 10^{-10}\ \_\mathit{SL12} - .2\,10^{-8}\ \_\mathit{SL10}, \mathit{wh} = 100.000000008\\
&+ .999999999983\ \_\mathit{SL13} - .849999999995\,10^{-11}\ \_\mathit{SL14} - .142500000000\ 10^{-10}\ \_\mathit{SL12}\\
&+ .1\,10^{-9}\ \_\mathit{SL10} + .0500000000016\ \mathit{he}, \mathit{sfh} = -1.39999999999\ \_\mathit{SL14} + 3820.00000000\\
&- .699999999999\ \_\mathit{SL13} - .199999999997\ \_\mathit{SL12} - 57.0000000002\ \_\mathit{SL10}\\
&+ .265000000001\ \mathit{he}, \mathit{so} = -.999999999993\ \_\mathit{SL13} - .999999999984\ \_\mathit{SL14}\\
&+ 413.999999993 - .999999999994\ \_\mathit{SL12} + 2.9999999999\ \_\mathit{SL10} - .0500000000016\ \mathit{he}\\
&, \mathit{co} = 42.0000000000 + .999999999994\ \_\mathit{SL12} - 1.00000000000\ \_\mathit{SL10}, \mathit{wsh} =\\
&-.400000000006\ \_\mathit{SL13} - 1.00000000000\ \_\mathit{SL14} + 3631.80000000\\
&- .100000000000\ \_\mathit{SL12} - 57.8999999991\ \_\mathit{SL10} + .279999999998\ \mathit{he}]
\end{aligned}$$

$$\begin{aligned}
\mathit{obj}\ = {}&, -7.39999999993\ \_\mathit{SL13} - 21.2999999999\ \_\mathit{SL14} + 80537.0000000\\
&- 4.89999999993\ \_\mathit{SL12} - 22.5000000007\ \_\mathit{SL10} - .0200000000207\ \mathit{he}
\end{aligned}$$

$$\begin{aligned}
&\{\mathit{wh} = 100.0000000,\ \mathit{sfh} = 3820.000000,\ \mathit{so} = 414.0000000,\ \mathit{co} = 42.00000000,\\
&\mathit{da} = 12.00000000,\ \mathit{he} = 0,\ \mathit{wsh} = 3631.800000\}
\end{aligned}$$

```
infolevel[simplex]:=1;
```

$$\mathit{infolevel}_{\mathit{simplex}} := 1$$

The final objective function looks like this:

By inspection of the final objective function, we see that the nonbasic slack variables in the objective belong to cow limit, sl10, and corn sl12, wheat sl13, and farm size sl14. The shadow price for the cow limit is the increase in the networth by increasing the cow limit from 12 to 13. This turns out to be $22.50, the negative of the coefficient of sl10 in the final set of equations. But the cost of the money for the year is .1*1500 = $150, so it would be disadvantageous to borrow the money to buy cows.

Don't borrow the money.

To make sure that our assertion that the marginal increase in the net worth by buying one more cow is 22.5, we can simply remaximize with the 12 limit replaced by 13 and calculate the

change in networth.

```
net := 34*so + 27.5*co + 20.75*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
constraints := [
70*so + 60 *co + 40*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
-10*so -15 *co + 0*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
15*so + 20 *co + 10*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
50*so + 40 *co + 30*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
-15*so -20 *co -10*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
10*so + 10 *co + 5*wh +
850*(da+30) + 4.25*(he+2000) +
(4000-wsh)*5+(4500-sfh)*5.5 +
20000 - 1500*da - 3*he +
```

```
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000+10000>=60000,
wsh <= 4000,
sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42 +1,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000) ,
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) =
sfh];
soln:=maximize(net,constraints,NONNEGATIVE);
subs(soln,matrix(8,2,[networth,net,
corn, co, soybeans, so,
wheat,wh,cows,da,hens,he,
winterspring,wsh,summerfall,sfh ]));
```

$$net := 34\ so + 27.5\ co + 20.75\ wh + 700.0\ da + 94000.00 + 3.50\ he - 5\ wsh - 5.5\ sfh$$

$$\begin{array}{l}
constraints := [\\
\quad 60000 \leq 70\ so + 60\ co + 40\ wh + 700.0\ da + 104000.00 + 3.50\ he - 5\ wsh - 5.5\ sfh,\\
\quad 60000 \leq -10\ so - 15\ co + 104000.00 + 700.0\ da + 3.50\ he - 5\ wsh - 5.5\ sfh,\\
\quad 60000 \leq 15\ so + 20\ co + 10\ wh + 700.0\ da + 104000.00 + 3.50\ he - 5\ wsh - 5.5\ sfh,\\
\quad 60000 \leq 50\ so + 40\ co + 30\ wh + 700.0\ da + 104000.00 + 3.50\ he - 5\ wsh - 5.5\ sfh,\\
\quad 60000 \leq -15\ so - 20\ co - 10\ wh + 700.0\ da + 104000.00 + 3.50\ he - 5\ wsh - 5.5\ sfh,\\
\quad 60000 \leq 10\ so + 10\ co + 5\ wh + 700.0\ da + 104000.00 + 3.50\ he - 5\ wsh - 5.5\ sfh,\\
\quad wsh \leq 4000,\ sfh \leq 4500,\ 1500\ da + 3\ he \leq 20000,\ da + 30 \leq 43,\ he + 2000 \leq 5000,\\
\quad da + 30 \leq co,\ .05\ he + 100.00 \leq wh,\ so + co + wh + 2\ da + 60 \leq 640,\\
\quad so + .9\ co + .6\ wh + 60\ da + 2400.00 + .30\ he = wsh,\\
\quad 1.4\ so + 1.2\ co + .7\ wh + 60\ da + 2400.00 + .30\ he = sfh]
\end{array}$$

$$\begin{array}{l}
soln := \{wh = 100.0000000,\ he = 0,\ sfh = 3877.000000,\ da = 13.00000000,\\
\quad co = 43.00000000,\ wsh = 3689.700000,\ so = 411.0000000\}
\end{array}$$

$$\left[\begin{array}{cc}
networth & 80559.50000\\
corn & 43.00000000\\
soybeans & 411.0000000\\
wheat & 100.0000000\\
cows & 13.00000000\\
hens & 0\\
winterspring & 3689.700000\\
summerfall & 3877.000000
\end{array}\right]$$

Sure enough, the networth increased by 59.5-37 =22.5

h) Use sensitivity analysis to identify the most sensitive coefficients of the objective function in part f).

```
with(linalg):with(simplex):
```

```
Warning, new definition for basis

Warning, new definition for pivot

Warning, new definition for basis
```

```
Warning, new definition for convexhull

Warning, new definition for cterm

Warning, new definition for display

Warning, new definition for dual

Warning, new definition for feasible

Warning, new definition for maximize

Warning, new definition for pivot

Warning, new definition for pivotvar

Warning, new definition for ratio

Warning, new definition for setup
```

We need gauss, getpiv and matmax from a previous worksheet. Getpiv and matmax are modified to exclude selected columns from being pivot columns.

```
gauss := proc(cl,r)
local id,n,multipliers;
n := vectdim(cl);
id := diag(1$n);
multipliers := evalm(-1/cl[r]*cl);
multipliers[r] := 1/cl[r];
augment(col(id,1..r-1),multipliers,col(id,r+1..n));
end:
```

Now getpiv takes a tableau T and a list of the column numbers of the nonbasic columns of T, and returns a suitable pivot row and column if one exist, else it returns FAIL.

```
getpiv := proc(T,nbv,exclusions)
local m, i, tmp, j;
m := coldim(T);
for i in {op(nbv)}minus exclusions do
tmp := 0,infinity;
if T[1,i]<0 then
for j from 2 to rowdim(T) do
if T[j,i]>0 and T[j,m]/T[j,i]<tmp[2]
then tmp := j,T[j,m]/T[j,i] fi;
od;
fi;
if tmp[1] > 0 then RETURN([tmp[1],i]) fi ;
od;
FAIL; end:
```

Matmax takes a tableau and a vector containing the list of column numbers of the basis for the columns of the tableau (less the right hand column.)

```
matmax := proc( tableau , basic,exclusions )
local pivot, T, nbv, n, m, b;
T := evalm(tableau);
n := coldim(T);m := rowdim(T):
b := evalm(basic);
```

```
   print(evalm(T));
   nbv := {seq(i,i=2..n-1)}minus{seq(b[i],i=1..m)};
   pivot := getpiv(T,nbv,exclusions );
   while not pivot=FAIL do
   b[pivot[1]]:= pivot[2];
   T := evalm(gauss(col(T,pivot[2]),pivot[1])&*T);
   print(evalm(T));
   nbv := {seq(i,i=2..n-1)} minus
{seq(b[i],i=1..m)};
   pivot := getpiv(T,nbv,exclusions );
   od;
   [evalm(T),evalm(b)];
   end:
```

Now we want to calculate the sensitivity ranges on the values of soybeans, corn, wheat, cows, hens, winterspring hours and fallsummer hours. So here is the problem set up with wsh and sfh eliminated as decision variables.

```
net := (34 )*so + 27.5*co + 20.75*wh +
850*(da+30) + 4.25*(he+2000) +
(1600-(1 *so + .9*co + .6*wh + 60*da + .3*he))*5+ (2100-(1.4*so +
1.2*co + .7*wh +60*da + .3*he))*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000 ;
```

$$net := 16.40\ co + 13.90\ wh + 70.0\ da + .35\ he + 21.30\ so + 68800.00$$

```
eqns := [z-net+68800=68800,
1500*da + 3*he + sl1 = 20000,
da + sl2 = 12,
he +sl3 = 3000,
co - da - sp1 + ar1 = 30,
wh - .05*he -sp2 + ar2 = 100,
so + co + wh + 2*da + sl4 = 580,
so + .9*co + .6*wh + 60*da + .3*he +sl5 = 1600,
1.4* so + 1.2*co + .7*wh + 60*da + .3*he +sl6 = 2100];
```

$$\begin{aligned}eqns := [&z - 16.40\ co - 13.90\ wh - 70.0\ da - .35\ he - 21.30\ so = 68800,\\ &1500\ da + 3\ he + sl1 = 20000,\ da + sl2 = 12,\ he + sl3 = 3000,\ co - da - sp1 + ar1 = 30,\\ &wh - .05\ he - sp2 + ar2 = 100,\ so + co + wh + 2\ da + sl4 = 580,\\ &so + .9\ co + .6\ wh + 60\ da + .3\ he + sl5 = 1600,\\ &1.4\ so + 1.2\ co + .7\ wh + 60\ da + .3\ he + sl6 = 2100]\end{aligned}$$

Since we have artificial variables, we have to have a phase 1 objective, to maximize the negative of the sum of the artificials.

```
artz := z+( convert([seq(
solve(eqns[i],ar.(i-4)) ,i=5..6)],'+'))=0;
```

$$artz := z - co + da + sp1 + 130. - 1.\ wh + .05000000000\ he + sp2 = 0$$

genmatrix is a nice word from linalg that enables us to construct the matrix to feed to matmax.

```
T := genmatrix([artz,op(eqns)],
[z,so,co,wh,da,he,sl1,sl2,sl3,ar1,ar2,sl4,sl5,sl6,sp1,sp2],flag);
```

$$T := \begin{bmatrix} 1, 0, -1, -1., 1, .05000000000, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, -130. \\ 1, -21.30, -16.40, -13.90, -70.0, -.35, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 68800. \\ 0, 0, 0, 0, 1500, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20000 \\ 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 12 \\ 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3000 \\ 0, 0, 1, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 30 \\ 0, 0, 0, 1, 0, -.05, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 100. \\ 0, 1, 1, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 580 \\ 0, 1, .9, .6, 60, .3, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1600. \\ 0, 1.4, 1.2, .7, 60, .3, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2100. \end{bmatrix}$$

We will keep a copy of the original problem matrix.

```
orig := delrows(T,1..1);
```

$$orig := \begin{bmatrix} 1, -21.30, -16.40, -13.90, -70.0, -.35, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 68800. \\ 0, 0, 0, 0, 1500, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20000 \\ 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 12 \\ 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3000 \\ 0, 0, 1, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 30 \\ 0, 0, 0, 1, 0, -.05, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 100. \\ 0, 1, 1, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 580 \\ 0, 1, .9, .6, 60, .3, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1600. \\ 0, 1.4, 1.2, .7, 60, .3, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2100. \end{bmatrix}$$

Now feed T to matmax with no exclusions.

```
soln1
:=matmax(T,[1,7,8,9,10,11,12,13,14,14],{});
```

$$\begin{bmatrix} 1, 0, -1, -1., 1, .05000000000, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, -130. \\ 1, -21.30, -16.40, -13.90, -70.0, -.35, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 68800. \\ 0, 0, 0, 0, 1500, 3, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 20000 \\ 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 12 \\ 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 3000 \\ 0, 0, 1, 0, -1, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 0, 30 \\ 0, 0, 0, 1, 0, -.05, 0, 0, 0, 0, 1, 0, 0, 0, 0, -1, 100. \\ 0, 1, 1, 1, 2, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 580 \\ 0, 1, .9, .6, 60, .3, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1600. \\ 0, 1.4, 1.2, .7, 60, .3, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 2100. \end{bmatrix}$$

```
[1 , 0 , 0 , −1. , 0 , .05000000000 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 1 , −100.]
    [1 , −21.30 , 0 , −13.90 , −86.40 , −.35 , 0 , 0 , 0 , 16.40 , 0 , 0 , 0 , 0 , −16.40 , 0 , 69292.00]
    [0 , 0 , 0 , 0 , 1500 , 3 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 20000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , −1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 30]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 1 , 3 , 0 , 0 , 0 , 0 , −1 , 0 , 1 , 0 , 0 , 1 , 0 , 550]
    [0 , 1 , 0 , .6 , 60.9 , .3 , 0 , 0 , 0 , −.9 , 0 , 0 , 1 , 0 , .9 , 0 , 1573.0]
    [0 , 1.4 , 0 , .7 , 61.2 , .3 , 0 , 0 , 0 , −1.2 , 0 , 0 , 0 , 1 , 1.2 , 0 , 2064.0]
```

```
[1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1. , 0 , 0 , 0 , 0 , 0 , 0]
    [1 , −21.30 , 0 , 0 , −86.40 , −1.0450 , 0 , 0 , 0 , 16.40 , 13.90 , 0 , 0 , 0 , −16.40 , −13.90 ,
    70682.00]
    [0 , 0 , 0 , 0 , 1500 , 3 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 20000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , −1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 30]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 0 , 3 , .05 , 0 , 0 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 450.]
    [0 , 1 , 0 , 0 , 60.9 , .330 , 0 , 0 , 0 , −.9 , −.6 , 0 , 1 , 0 , .9 , .6 , 1513.0]
    [0 , 1.4 , 0 , 0 , 61.2 , .335 , 0 , 0 , 0 , −1.2 , −.7 , 0 , 0 , 1 , 1.2 , .7 , 1994.0]

soln1 := [
    [1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 1 , 1. , 0 , 0 , 0 , 0 , 0 , 0]
    [1 , −21.30 , 0 , 0 , −86.40 , −1.0450 , 0 , 0 , 0 , 16.40 , 13.90 , 0 , 0 , 0 , −16.40 , −13.90 ,
    70682.00]
    [0 , 0 , 0 , 0 , 1500 , 3 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 20000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , −1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 30]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 0 , 3 , .05 , 0 , 0 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 450.]
    [0 , 1 , 0 , 0 , 60.9 , .330 , 0 , 0 , 0 , −.9 , −.6 , 0 , 1 , 0 , .9 , .6 , 1513.0]
    [0 , 1.4 , 0 , 0 , 61.2 , .335 , 0 , 0 , 0 , −1.2 , −.7 , 0 , 0 , 1 , 1.2 , .7 , 1994.0],
    [1, 7, 8, 9, 10, 3, 4, 13, 14, 14]]
```

At this point phase 1 is successfully completed. So we get rid of the top row.

```
T :=delrows(soln1[1],1..1);
```

```
T :=
    [1 , −21.30 , 0 , 0 , −86.40 , −1.0450 , 0 , 0 , 0 , 16.40 , 13.90 , 0 , 0 , 0 , −16.40 , −13.90 ,
    70682.00]
    [0 , 0 , 0 , 0 , 1500 , 3 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 20000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , −1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 30]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 0 , 3 , .05 , 0 , 0 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 450.]
    [0 , 1 , 0 , 0 , 60.9 , .330 , 0 , 0 , 0 , −.9 , −.6 , 0 , 1 , 0 , .9 , .6 , 1513.0]
    [0 , 1.4 , 0 , 0 , 61.2 , .335 , 0 , 0 , 0 , −1.2 , −.7 , 0 , 0 , 1 , 1.2 , .7 , 1994.0]
```

Now feed this matrix to matmax, with the exclusion of the 10 and 11 columns for consideration as basic variables.

```
phase2
:=matmax(T,[1,3,4,7,8,9,10,11,12],{10,11});
```

```
[1 , −21.30 , 0 , 0 , −86.40 , −1.0450 , 0 , 0 , 0 , 16.40 , 13.90 , 0 , 0 , 0 , −16.40 , −13.90 ,
    70682.00]
    [0 , 0 , 0 , 0 , 1500 , 3 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 20000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , −1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 30]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 0 , 3 , .05 , 0 , 0 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 450.]
```

```
        [0 , 1 , 0 , 0 , 60.9 , .330 , 0 , 0 , 0 , −.9 , −.6 , 0 , 1 , 0 , .9 , .6 , 1513.0]
        [0 , 1.4 , 0 , 0 , 61.2 , .335 , 0 , 0 , 0 , −1.2 , −.7 , 0 , 0 , 1 , 1.2 , .7 , 1994.0]

[1 , 0 , 0 , 0 , −22.50 , .0200 , 0 , 0 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 , 80267.00]
    [0 , 0 , 0 , 0 , 1500 , 3 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 20000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , −1 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 30]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 0 , 3 , .05 , 0 , 0 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 450.]
    [0 , 0 , 0 , 0 , 57.9 , .280 , 0 , 0 , 0 , .1 , .4 , −1 , 1 , 0 , −.1 , −.4 , 1063.0]
    [0 , 0 , 0 , 0 , 57.0 , .265 , 0 , 0 , 0 , .2 , .7 , −1.4 , 0 , 1 , −.2 , −.7 , 1364.0]

[1 , 0 , 0 , 0 , 0 , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 , 80537.00]
    [0 , 0 , 0 , 0 , 0 , 3 , 1 , −1500 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000]
    [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
    [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
    [0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 42]
    [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
    [0 , 1 , 0 , 0 , 0 , .05 , 0 , −3 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1 , 1 , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1 , −.2 , −.7 , 680.0]

  phase2 := [
        [1 , 0 , 0 , 0 , 0 , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 , 80537.00]
        [0 , 0 , 0 , 0 , 0 , 3 , 1 , −1500 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000]
        [0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12]
        [0 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000]
        [0 , 0 , 1 , 0 , 0 , 0 , 0 , 1 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 0 , 42]
        [0 , 0 , 0 , 1 , 0 , −.05 , 0 , 0 , 0 , 0 , 1 , 0 , 0 , 0 , 0 , −1 , 100.]
        [0 , 1 , 0 , 0 , 0 , .05 , 0 , −3 , 0 , −1 , −1 , 1 , 0 , 0 , 1 , 1 , 414.]
        [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1 , 1 , 0 , −.1 , −.4 , 368.2]
        [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1 , −.2 , −.7 , 680.0],
        [1, 3, 5, 7, 8, 9, 2, 11, 12]]
```

So we have arrived at the optimum planning schedule previously obtained. But we are in a position to do the sensitivity analysis, because Tf contains a copy of the product g of the gauss matrices in the original basic columns.

```
Tf := evalf(phase2[1]);
```

```
  Tf :=
        [1. , 0 , 0 , 0 , 0 , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 , 80537.00]
        [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
        [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
        [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
        [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
        [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
        [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
        [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
        [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

We can extract g from Tf by deleting columns. (in two bunches).

```
g := delcols(Tf, 2..6 );
```

$$g := \begin{bmatrix} 1. & 0 & 22.50 & 0 & -4.90 & -7.40 & 21.30 & 0 & 0 & 4.90 & 7.40 & 80537.00 \\ 0 & 1. & -1500. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2000. \\ 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12. \\ 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3000. \\ 0 & 0 & 1. & 0 & 1. & 0 & 0 & 0 & 0 & -1. & 0 & 42. \\ 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & -1. & 100. \\ 0 & 0 & -3. & 0 & -1. & -1. & 1. & 0 & 0 & 1. & 1. & 414. \\ 0 & 0 & -57.9 & 0 & .1 & .4 & -1. & 1. & 0 & -.1 & -.4 & 368.2 \\ 0 & 0 & -57.0 & 0 & .2 & .7 & -1.4 & 0 & 1. & -.2 & -.7 & 680.0 \end{bmatrix}$$

```
g := delcols(g,10..12);
```

$$g := \begin{bmatrix} 1. & 0 & 22.50 & 0 & -4.90 & -7.40 & 21.30 & 0 & 0 \\ 0 & 1. & -1500. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 1. & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & -3. & 0 & -1. & -1. & 1. & 0 & 0 \\ 0 & 0 & -57.9 & 0 & .1 & .4 & -1. & 1. & 0 \\ 0 & 0 & -57.0 & 0 & .2 & .7 & -1.4 & 0 & 1. \end{bmatrix}$$

Multiply g by the original tableau to check that we get the final tableau.

```
evalm(g&*orig);
```

```
[1., 0, 0, 0, 0, .0200, 0, 22.50, 0, −4.90, −7.40, 21.30, 0, 0, 4.90, 7.40, 80537.00]
    [0, 0, 0, 0, 0, 3., 1., −1500., 0, 0, 0, 0, 0, 0, 0, 0, 2000.]
    [0, 0, 0, 0, 1., 0, 0, 1., 0, 0, 0, 0, 0, 0, 0, 0, 12.]
    [0, 0, 0, 0, 0, 1., 0, 0, 1., 0, 0, 0, 0, 0, 0, 0, 3000.]
    [0, 0, 1., 0, 0, 0, 0, 1., 0, 1., 0, 0, 0, 0, −1., 0, 42.]
    [0, 0, 0, 1., 0, −.05, 0, 0, 0, 0, 1., 0, 0, 0, 0, −1., 100.]
    [0, 1., 0, 0, 0, .05, 0, −3., 0, −1., −1., 1., 0, 0, 1., 1., 414.]
    [0, 0, 0, 0, 0, .280, 0, −57.9, 0, .1, .4, −1., 1., 0, −.1, −.4, 368.2]
    [0, 0, 0, 0, 0, .265, 0, −57.0, 0, .2, .7, −1.4, 0, 1., −.2, −.7, 680.0]
```

Good. We are ready to roll.

Sensitivity Ranges

The method for getting the range is the same for each value: We add theta to the value in the original tableau, and multiply by g. The resulting tableau needs to be multiplied by a gauss matrix to zero out the top of the basic column. Then we solve the inequalities we get by required that the tops of all the nonbasic columns be nonnegative (rightmost column excluded). The range for theta is obtained, and we add the original value to get the range on the value.

The function v defined below will be useful

```
v := (t1,t2,t3,t4,t5,t6,t7) ->(34+t1)*so +
(27.5+t2)*co + (20.75+t3)*wh +
(850+t4)*(da+30) + (4.25+t5)*(he+2000) +
(1600-(1 *so + .9*co + .6*wh + 60*da + .3*he))*(5+t6)+ (2100-(1.4*so
+ 1.2*co + .7*wh +60*da + .3*he))*(5.5+t7) +
20000 - 1500*da - 3*he +
```

```
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000 ;
```

$$\begin{array}{l} v := (\mathit{t1},\ \mathit{t2},\ \mathit{t3},\ \mathit{t4},\ \mathit{t5},\ \mathit{t6},\ \mathit{t7}) \to (34 + \mathit{t1})\, \mathit{so} + (27.5 + \mathit{t2})\, \mathit{co} + (20.75 + \mathit{t3})\, \mathit{wh} \\ \quad + (850 + \mathit{t4})\, (\mathit{da} + 30) + (4.25 + \mathit{t5})\, (\mathit{he} + 2000) \\ \quad + (1600 - \mathit{so} - .9\, \mathit{co} - .6\, \mathit{wh} - 60\, \mathit{da} - .3\, \mathit{he})\, (5 + \mathit{t6}) \\ \quad + (2100 - 1.4\, \mathit{so} - 1.2\, \mathit{co} - .7\, \mathit{wh} - 60\, \mathit{da} - .3\, \mathit{he})\, (5.5 + \mathit{t7}) + 15250.00 - 150.0\, \mathit{da} \\ \quad - .75\, \mathit{he} \end{array}$$

Also, a form of scenario modified to put any values will be useful for checking purposes.

```
scenario := proc(a,b,c,d,e,f,g)
local net,constraints,soln,wsh,sfh,so,
co,wh,he,da;
net := a*so + b*co + c*wh +
d*(da+30) + e*(he+2000) +
(4000-wsh)*f+(4500-sfh)*g +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
constraints := [
wsh <= 4000,
sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000),
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) = sfh];
soln:=maximize(net,constraints,NONNEGATIVE);
subs(soln,matrix(8,2,[networth,net,
corn, co, soybeans, so,
wheat,wh,cows,da,hens,he,
winterspring,wsh,summerfall,sfh ])); end;
```

$$\begin{array}{l} \mathit{scenario} := \mathrm{proc}(a,\ b,\ c,\ d,\ e,\ f,\ g) \\ \quad \mathrm{local}\mathit{net},\ \mathit{constraints},\ \mathit{soln},\ \mathit{wsh},\ \mathit{sfh},\ \mathit{so},\ \mathit{co},\ \mathit{wh},\ \mathit{he},\ \mathit{da}; \\ \qquad \mathit{net} := a \times \mathit{so} + b \times \mathit{co} + c \times \mathit{wh} + d \times (\mathit{da} + 30) + e \times (\mathit{he} + 2000) + (4000 - \mathit{wsh}) \times f \\ \qquad\quad + (4500 - \mathit{sfh}) \times g + 15250.00 - 150.0 \times \mathit{da} - .75 \times \mathit{he}; \\ \qquad \mathit{constraints} := [\mathit{wsh} \le 4000,\ \mathit{sfh} \le 4500,\ 1500 \times \mathit{da} + 3 \times \mathit{he} \le 20000,\ \mathit{da} + 30 \le 42, \\ \qquad\quad \mathit{he} + 2000 \le 5000,\ \mathit{da} + 30 \le \mathit{co},\ .05 \times \mathit{he} + 100.00 \le \mathit{wh},\ \mathit{so} + \mathit{co} + \mathit{wh} + 2 \times \mathit{da} + 60 \le 640, \\ \qquad\quad \mathit{so} + .9 \times \mathit{co} + .6 \times \mathit{wh} + 60 \times \mathit{da} + 2400.00 + .30 \times \mathit{he} = \mathit{wsh}, \\ \qquad\quad 1.4 \times \mathit{so} + 1.2 \times \mathit{co} + .7 \times \mathit{wh} + 60 \times \mathit{da} + 2400.00 + .30 \times \mathit{he} = \mathit{sfh}]; \\ \qquad \mathit{soln} := \mathrm{maximize}(\mathit{net},\ \mathit{constraints},\ \mathit{NONNEGATIVE}); \\ \qquad \mathrm{subs}(\mathit{soln}, \mathrm{matrix}(8,\ 2,\ [\mathit{networth},\ \mathit{net},\ \mathit{corn},\ \mathit{co},\ \mathit{soybeans},\ \mathit{so},\ \mathit{wheat},\ \mathit{wh},\ \mathit{cows},\ \mathit{da},\ \mathit{hens}, \\ \qquad\quad \mathit{he},\ \mathit{winterspring},\ \mathit{wsh},\ \mathit{summerfall},\ \mathit{sfh}])) \\ \quad \mathrm{end} \end{array}$$

The function showgt shows the final modified tableau. We have to inspect this manually to decide the appropriate gauss matrix to premultiply by.

```
showgt := proc(n)
local net, eqns, T;
net := v(seq(0,i=1..n-1),theta,seq(0,i=n+1..7));
```

```
eqns := [z-net+68800=68800,
1500*da + 3*he + sl1 = 20000,
da + sl2 = 12,
he +sl3 = 3000,
co - da - sp1 + ar1 = 30,
wh - .05*he -sp2 + ar2 = 100,
so + co + wh + 2*da + sl4 = 580,
so + .9*co + .6*wh + 60*da + .3*he +sl5 = 1600,
1.4* so + 1.2*co + .7*wh + 60*da + .3*he +sl6 = 2100];
T := genmatrix(eqns,
[z,so,co,wh,da,he,sl1,sl2,sl3,ar1,ar2,sl4,sl5,sl6,sp1,sp2],flag);
evalm(g&*T); end;
```

```
showgt := proc(n)
    localnet, eqns, T;
        net := v(seq(0, i = 1..n − 1), θ, seq(0, i = n + 1..7)) ;
        eqns := [z − net + 68800 = 68800, 1500 × da + 3 × he + sl1 = 20000, da + sl2 = 12,
            he + sl3 = 3000, co − da − sp1 + ar1 = 30, wh − .05 × he − sp2 + ar2 = 100,
            so + co + wh + 2 × da + sl4 = 580, so + .9 × co + .6 × wh + 60 × da + .3 × he + sl5 = 1600,
            1.4 × so + 1.2 × co + .7 × wh + 60 × da + .3 × he + sl6 = 2100];
        T :=
            genmatrix(eqns, [z, so, co, wh, da, he, sl1, sl2, sl3, ar1, ar2, sl4, sl5, sl6, sp1, sp2], flag);
        evalm(g &* T)
    end
```

Now we are ready to go.

sensitivity range on value of soybeans

```
ok := showgt(1);
```

```
ok :=
    [1. , −1. θ , 0 , 0 , 0 , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 ,
    80537.00]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
ineqns :=
row(evalm(gauss(col(ok,2),7)&*ok),1);
sol :=
solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>  ;
```

```
ineqns := [1., 0, 0, 0, 0, .0200 + .05 θ, 0, 22.50 − 3. θ, 0, −4.90 − 1. θ, −7.40 − 1. θ,
    21.30 + 1. θ, 0, 0, 4.90 + 1. θ, 7.40 + 1. θ, 80537.00 + 414. θ]
                    sol := {−.4000000000 ≤ θ, θ ≤ 7.500000000}
```

```
seq(sl + '34<=34',sl=sol);
```

```
                    33.60000000 ≤ θ + 34, θ + 34 ≤ 41.50000000
```

So the value of an acre of soybeans can vary from 33.6 to 38.9 to keep the same planting schedule. To check this, we can use scenario.

```
scenario(33.5,27.5,20.75,850,4.25,5.,5.5);
```

```
[ networth      80333.33333 ]
[ corn          42.00000000 ]
[ soybeans      380.6666667 ]
[ wheat         133.3333333 ]
[ cows              12.     ]
[ hens          666.6666667 ]
[ winterspring  3818.466667 ]
[ summerfall    3996.666667 ]
```

```
scenario(33.6,27.5,20.75,850,4.25,5.,5.5);
```

```
[ networth      80371.40000 ]
[ corn          42.00000000 ]
[ soybeans      414.0000000 ]
[ wheat         100.0000000 ]
[ cows              12.     ]
[ hens               0      ]
[ winterspring  3631.800000 ]
[ summerfall    3820.000000 ]
```

sensitivity range on value of corn

This shows what planting schedule to change to if soybeans drop down below 33.6 (and all other values remain).

```
ok := showgt(2);
```

```
ok :=
    [1. , 0 , −1.θ , 0 , 0 , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 ,
    80537.00]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
ineqns :=
row(evalm(gauss(col(ok,3),5)&*ok),1);
sol :=
solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>  ;
```

```
ineqns := [1., 0, 0, 0, 0, .0200, 0, 22.50 + 1.θ, 0, −4.90 + 1.θ, −7.40, 21.30, 0, 0,
    4.90 − 1.θ, 7.40, 80537.00 + 42.θ]
                    sol := {−22.50000000 ≤ θ, θ ≤ 4.900000000}
```

```
seq(sl + '27.5<=27.5',sl=sol);
```

```
5.00000000 ≤ 27.5 + θ, 27.5 + θ ≤ 32.40000000
```

So the value of an acre of corn can vary from 5.00 to 32.4 to keep the same planting schedule. What happens at the upper end is, we switch from soybeans to corn.

```
scenario(34,32.4,20.75,850,4.25,5,5.5);
```

```
[ networth      80742.80000 ]
[ corn          456.0000000 ]
[ soybeans      0           ]
[ wheat         100.0000000 ]
[ cows          12.00000000 ]
[ hens          0           ]
[ winterspring  3590.400000 ]
[ summerfall    3737.200000 ]
```

sensitivity range on value of wheat

```
ok := showgt(3);
```

```
ok :=
    [1. , 0 , 0 , −1. θ , 0 , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 ,
    80537.00]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
  ineqns :=
  row(evalm(gauss(col(ok,4),6)&*ok),1);
  sol :=
  solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>   ;
```

```
ineqns := [1., 0, 0, 0, 0, .0200 − .05 θ, 0, 22.50, 0, −4.90, −7.40 + 1. θ, 21.30, 0, 0, 4.90,
    7.40 − 1. θ, 80537.00 + 100. θ]
                              sol := {θ ≤ .4000000000}
```

```
seq(sl + '20.75<=20.75',sl=sol);
```

```
20.75 + θ ≤ 21.15000000
```

So the value of an acre of wheat can be anything up to 21.15 to keep the same planting schedule.

sensitivity range on value of a cow

```
ok := showgt(4);
```

```
ok :=
    [1. , 0 , 0 , 0 , −1. θ , .0200 , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 ,
    80537.00 + 30. θ]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
```

```
        [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
        [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
        [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
        [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
        [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
  ineqns :=
  row(evalm(gauss(col(ok,5),3)&*ok),1);
  sol :=
  solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>  ;
 ineqns := [1., 0, 0, 0, 0, .0200, 0, 22.50 + 1. θ, 0, −4.90, −7.40, 21.30, 0, 0, 4.90, 7.40,
     80537.00 + 42. θ]
                          sol := {−22.50000000 ≤ θ}
  seq(sl + '850<=850',sl=sol);
                            827.5000000 ≤ θ + 850
```

The value of a cow can go all the way down to 827.5 before it becomes non profitable to buy new cows.

sensitivity range on value of a hen

```
  ok := showgt(5);
    ok :=
       [1. , 0 , 0 , 0 , 0 , .0200 − 1. θ , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 , 7.40 ,
       80537.00 + 2000. θ]
       [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
       [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
       [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
       [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
       [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
       [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
       [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
       [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
  ineqns := row(ok,1);
  sol :=
  solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>  ;
 ineqns := [1., 0, 0, 0, 0, .0200 − 1. θ, 0, 22.50, 0, −4.90, −7.40, 21.30, 0, 0, 4.90, 7.40,
     80537.00 + 2000. θ]
                          sol := {θ ≤ .02000000000}
  seq(sl + '4.25<=4.25',sl=sol);
                            4.25 + θ ≤ 4.270000000
```

If the value of a chicken goes up to 4.27 (by only 2 cents) it will become profitable to buy more chickens.

sensitivity range on value of winterspring hour

Here we will need to zero out the tops of five basic columns because wsh is implicitiy a function of those 5 variables.

```
ok := showgt(6);
```

```
ok :=
    [1. , 1. θ , .9 θ , .6 θ , 60. θ , .0200 + .3 θ , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 , 4.90 ,
    7.40 , 80537.00 + 1600. θ]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
ok := evalm(gauss(col(ok,2),7)&*ok);
```

```
ok :=
    [1. , 0 , .9 θ , .6 θ , 60. θ , .0200 + .25 θ , 0 , 22.50 + 3. θ , 0 , −4.90 + 1. θ , −7.40 + 1. θ ,
    21.30 − 1. θ , 0 , 0 , 4.90 − 1. θ , 7.40 − 1. θ , 80537.00 + 1186. θ]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
ok := evalm(gauss(col(ok,3),5)&*ok);
```

```
ok :=
    [1. , 0 , 0 , .6 θ , 60. θ , .0200 + .25 θ , 0 , 22.50 + 2.1 θ , 0 , −4.90 + .1 θ , −7.40 + 1. θ ,
    21.30 − 1. θ , 0 , 0 , 4.90 − .1 θ , 7.40 − 1. θ , 80537.00 + 1148.2 θ]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
ok := evalm(gauss(col(ok,4),6)&*ok);
```

```
ok :=
    [1. , 0 , 0 , 0 , 60. θ , .0200 + .280 θ , 0 , 22.50 + 2.1 θ , 0 , −4.90 + .1 θ , −7.40 + .4 θ ,
    21.30 − 1. θ , 0 , 0 , 4.90 − .1 θ , 7.40 − .4 θ , 80537.00 + 1088.2 θ]
    [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
    [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
    [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
    [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
    [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
    [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
    [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
    [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
ok := evalm(gauss(col(ok,5),3)&*ok);

  ok :=
     [1. , 0 , 0 , 0 , 0 , .0200 + .280 θ , 0 , 22.50 − 57.9 θ , 0 , −4.90 + .1 θ , −7.40 + .4 θ ,
     21.30 − 1. θ , 0 , 0 , 4.90 − .1 θ , 7.40 − .4 θ , 80537.00 + 368.2 θ]
     [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
     [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
     [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
     [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
     [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
     [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
     [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
     [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
```

```
   ineqns := row(ok,1);
   sol :=
   solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>  ;
  ineqns := [1., 0, 0, 0, 0, .0200 + .280 θ, 0, 22.50 − 57.9 θ, 0, −4.90 + .1 θ, −7.40 + .4 θ,
      21.30 − 1. θ, 0, 0, 4.90 − .1 θ, 7.40 − .4 θ, 80537.00 + 368.2 θ]
                       sol := {−.07142857143 ≤ θ, θ ≤ .3886010363}
   seq(sl + '5<=5',sl=sol);
                       4.928571429 ≤ θ + 5, θ + 5 ≤ 5.388601036
```

So the value of an hour of winter spring time can vary from 5.39 to 4.93 to keep the same planting schedule.

To get the summer fall hour value:

```
ok := showgt(7);

  ok :=
     [1. , 1.4 θ , 1.2 θ , .7 θ , 60. θ , .0200 + .3 θ , 0 , 22.50 , 0 , −4.90 , −7.40 , 21.30 , 0 , 0 ,
     4.90 , 7.40 , 80537.00 + 2100. θ]
     [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
     [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
     [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
     [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
     [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
     [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
     [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
     [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
ok := evalm(gauss(col(ok,2),7)&*ok):
ok := evalm(gauss(col(ok,3),5)&*ok):
ok := evalm(gauss(col(ok,4),6)&*ok):
ok := evalm(gauss(col(ok,5),3)&*ok);

  ok :=
     [1. , 0 , 0 , 0 , 0 , .0200 + .265 θ , 0 , 22.50 − 57.0 θ , 0 , −4.90 + .2 θ , −7.40 + .7 θ ,
     21.30 − 1.4 θ , 0 , 0 , 4.90 − .2 θ , 7.40 − .7 θ , 80537.00 + 680.0 θ]
     [0 , 0 , 0 , 0 , 0 , 3. , 1. , −1500. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 2000.]
     [0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 12.]
     [0 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 1. , 0 , 0 , 0 , 0 , 0 , 0 , 0 , 3000.]
     [0 , 0 , 1. , 0 , 0 , 0 , 0 , 1. , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 0 , 42.]
```

```
          [0 , 0 , 0 , 1. , 0 , −.05 , 0 , 0 , 0 , 0 , 1. , 0 , 0 , 0 , 0 , −1. , 100.]
          [0 , 1. , 0 , 0 , 0 , .05 , 0 , −3. , 0 , −1. , −1. , 1. , 0 , 0 , 1. , 1. , 414.]
          [0 , 0 , 0 , 0 , 0 , .280 , 0 , −57.9 , 0 , .1 , .4 , −1. , 1. , 0 , −.1 , −.4 , 368.2]
          [0 , 0 , 0 , 0 , 0 , .265 , 0 , −57.0 , 0 , .2 , .7 , −1.4 , 0 , 1. , −.2 , −.7 , 680.0]
  ineqns := row(ok ,1);
  sol :=
  solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
>   ;
   ineqns := [1., 0, 0, 0, 0, .0200 + .265 θ, 0, 22.50 − 57.0 θ, 0, −4.90 + .2 θ, −7.40 + .7 θ,
       21.30 − 1.4 θ, 0, 0, 4.90 − .2 θ, 7.40 − .7 θ, 80537.00 + 680.0 θ]
                           sol := {θ ≤ .3947368421, −.07547169811 ≤ θ}
  seq(sl + '5.5<=5.5',sl=sol);
                           θ + 5.5 ≤ 5.894736842, 5.424528302 ≤ θ + 5.5
```

So the value of an hour of summer fall time can vary from 5.43 to 4.89 to keep the same planting schedule.

The most sensitive (in absolute terms) is the value of hen.

21 Parametric programming example

In the case problem from project 1, we f ound that the yearly income from a hen was the most sensitive coefficient of all of the coefficients of the objective function in that when the income went up from 4.25 to 4.8 there was a large change in 'planting schedule' used to maximize net worth (although this change resulted in only a 25 dollar increase in net worth). What we want to do here is get the net income as a function of this coefficient, which we will call theta, because that seems to be what everybody is calling it.

```
with(linalg):with(simplex):
```

```
Warning, new definition for norm

Warning, new definition for trace

Warning, new definition for basis

Warning, new definition for maximize

Warning, new definition for minimize

Warning, new definition for pivot
```

We need gauss, getpiv and matmax from a previous worksheet. Getpiv and matmax are modified to exclude selected columns from being pivot columns.

```
gauss := proc(cl,r)
local id,n,multipliers;
n := vectdim(cl);
id := diag(1$n);
multipliers := evalm(-1/cl[r]*cl);
multipliers[r] := 1/cl[r];
augment(col(id,1..r-1),multipliers,col(id,r+1..n));
end:
```

Now getpiv takes a tableau T and a list of the column numbers of the nonbasic columns of T, and returns a suitable pivot row and column if one exist, else it returns FAIL. We have modified getpiv so that selection columns can be excluded from consideration.

```
getpiv := proc(T,nbv,exclusions)
local m, i, tmp, j;
m := coldim(T);
for i in {op(nbv)}minus exclusions do
tmp := 0,infinity;
if T[1,i]<0 then
for j from 2 to rowdim(T) do
if T[j,i]>0 and T[j,m]/T[j,i]<tmp[2]
then tmp := j,T[j,m]/T[j,i] fi;
od;
fi;
if tmp[1] > 0 then RETURN([tmp[1],i]) fi ;
od;
FAIL; end:
```

Matmax takes a tableau and a vector containing the list of column numbers of the basis for the columns of the tableau (less the right hand column.) We have modified matmax so that we can omit selected columns from consideration.

```
   matmax := proc( tableau , basic,exclusions )
   local pivot, T, nbv, n, m, b;
   T := evalm(tableau);
   n := coldim(T);m := rowdim(T):
   b := evalm(basic);
   print(evalm(T));
   nbv := {seq(i,i=2..n-1)}minus{seq(b[i],i=1..m)};
   pivot := getpiv(T,nbv,exclusions );
   while not pivot=FAIL do
   b[pivot[1]]:= pivot[2];
   T := evalm(gauss(col(T,pivot[2]),pivot[1])&*T);
   print(evalm(T));
   nbv := {seq(i,i=2..n-1)} minus
{seq(b[i],i=1..m)};
   pivot := getpiv(T,nbv,exclusions );
   od;
   [evalm(T),evalm(b)];
   end:
```

So here is the problem set up with wsh and sfh eliminated as decision variables.

```
net := (34 )*so + 27.5*co + 20.75*wh +
850*(da+30) + 4.25*(he+2000) +
(1600-(1 *so + .9*co + .6*wh + 60*da + .3*he))*5+ (2100-(1.4*so +
1.2*co + .7*wh +60*da + .3*he))*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000 ;
```

$$\mathit{net} := 21.30\,\mathit{so} + 16.40\,\mathit{co} + 13.90\,\mathit{wh} + 70.0\,\mathit{da} + .35\,\mathit{he} + 68800.00$$

```
eqns := [z-net+68800=68800,
1500*da + 3*he + sl1 = 20000,
da + sl2 = 12,
he +sl3 = 3000,
co - da - sp1 + ar1 = 30,
wh - .05*he -sp2 + ar2 = 100,
so + co + wh + 2*da + sl4 = 580,
so + .9*co + .6*wh + 60*da + .3*he +sl5 = 1600,
1.4* so + 1.2*co + .7*wh + 60*da + .3*he +sl6 = 2100];
```

$$\begin{aligned}\mathit{eqns} := [&z - 21.30\,\mathit{so} - 16.40\,\mathit{co} - 13.90\,\mathit{wh} - 70.0\,\mathit{da} - .35\,\mathit{he} = 68800,\\ &1500\,\mathit{da} + 3\,\mathit{he} + \mathit{sl1} = 20000,\ \mathit{da} + \mathit{sl2} = 12,\ \mathit{he} + \mathit{sl3} = 3000,\ \mathit{co} - \mathit{da} - \mathit{sp1} + \mathit{ar1} = 30,\\ &\mathit{wh} - .05\,\mathit{he} - \mathit{sp2} + \mathit{ar2} = 100,\ \mathit{so} + \mathit{co} + \mathit{wh} + 2\,\mathit{da} + \mathit{sl4} = 580,\\ &\mathit{so} + .9\,\mathit{co} + .6\,\mathit{wh} + 60\,\mathit{da} + .3\,\mathit{he} + \mathit{sl5} = 1600,\\ &1.4\,\mathit{so} + 1.2\,\mathit{co} + .7\,\mathit{wh} + 60\,\mathit{da} + .3\,\mathit{he} + \mathit{sl6} = 2100]\end{aligned}$$

Since we have artificial variables, we have to have a phase 1 objective, to maximize the negative of the sum of the artificials.

```
artz := z+( convert([seq(
solve(eqns[i],ar.(i-4)) ,i=5..6)],`+`))=0;
```

$$\mathit{artz} := z - \mathit{co} + \mathit{da} + \mathit{sp1} + 130. - 1.\,\mathit{wh} + .05000000000\,\mathit{he} + \mathit{sp2} = 0$$

genmatrix is a nice word from linalg that enables us to construct the matrix to feed to matmax.

```
T := genmatrix([artz,op(eqns)],
[z,so,co,wh,da,he,sl1,sl2,sl3,ar1,ar2,sl4,sl5,sl6,sp1,sp2],flag);
```

We will keep a copy of the original problem matrix.

```
orig := delrows(T,1..1);
```

Now feed T to matmax with no exclusions.

```
soln1
:=matmax(T,[1,7,8,9,10,11,12,13,14,14],{});
```

At this point phase 1 is successfully completed. So we get rid of the top row.

```
T :=delrows(soln1[1],1..1);
```

Now feed this matrix to matmax, with the exclusion of the 10 and 11 columns for consideration as basic variables.

```
phase2 :=matmax(T,[1, 7, 8, 9, 12, 3, 4, 13,
14],{10,11});
```

So we have arrived at the optimum planning schedule previously obtained. But we are in a position to do the parametric because Tf contains a copy of the product g of the gauss matrices in the original basic columns.

```
Tf := evalf(phase2[1]);
```

```
Tf :=
    [1., 0, 0, 0, 0, .0200, 0, 22.50, 0, −4.90, −7.40, 21.30, 0, 0, 4.90, 7.40, 80537.00]
    [0, 0, 0, 0, 0, 3., 1., −1500., 0, 0, 0, 0, 0, 0, 0, 0, 2000.]
    [0, 0, 0, 0, 1., 0, 0, 1., 0, 0, 0, 0, 0, 0, 0, 0, 12.]
    [0, 0, 0, 0, 0, 1., 0, 0, 1., 0, 0, 0, 0, 0, 0, 0, 3000.]
    [0, 0, 1., 0, 0, 0, 0, 1., 0, 1., 0, 0, 0, 0, −1., 0, 42.]
    [0, 0, 0, 1., 0, −.05, 0, 0, 0, 0, 1., 0, 0, 0, 0, −1., 100.]
    [0, 1., 0, 0, 0, .05, 0, −3., 0, −1., −1., 1., 0, 0, 1., 1., 414.]
    [0, 0, 0, 0, 0, .280, 0, −57.9, 0, .1, .4, −1., 1., 0, −.1, −.4, 368.2]
    [0, 0, 0, 0, 0, .265, 0, −57.0, 0, .2, .7, −1.4, 0, 1., −.2, −.7, 680.0]
```

We can extract g from Tf by deleting columns. (in two bunches).

```
g := delcols(Tf, 2..6 );
```

$$g := \begin{bmatrix} 1. & 0 & 22.50 & 0 & -4.90 & -7.40 & 21.30 & 0 & 0 & 4.90 & 7.40 & 80537.00 \\ 0 & 1. & -1500. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2000. \\ 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 12. \\ 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3000. \\ 0 & 0 & 1. & 0 & 1. & 0 & 0 & 0 & 0 & -1. & 0 & 42. \\ 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & -1. & 100. \\ 0 & 0 & -3. & 0 & -1. & -1. & 1. & 0 & 0 & 1. & 1. & 414. \\ 0 & 0 & -57.9 & 0 & .1 & .4 & -1. & 1. & 0 & -.1 & -.4 & 368.2 \\ 0 & 0 & -57.0 & 0 & .2 & .7 & -1.4 & 0 & 1. & -.2 & -.7 & 680.0 \end{bmatrix}$$

```
g := delcols(g,10..12);
```

$$g := \begin{bmatrix} 1. & 0 & 22.50 & 0 & -4.90 & -7.40 & 21.30 & 0 & 0 \\ 0 & 1. & -1500. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1. & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1. & 0 & 1. & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1. & 0 & 0 & 0 \\ 0 & 0 & -3. & 0 & -1. & -1. & 1. & 0 & 0 \\ 0 & 0 & -57.9 & 0 & .1 & .4 & -1. & 1. & 0 \\ 0 & 0 & -57.0 & 0 & .2 & .7 & -1.4 & 0 & 1. \end{bmatrix}$$

Multiply g by the original tableau to check that we get the final tableau.

```
evalm(g&*orig);
```

Good. We are ready to roll.

The method for getting the networth as a function of the value of a hen is to substitute theta in for 4.25 in the expression for networth .

```
v := theta->34*so + 27.5*co + 20.75*wh +
850*(da+30) + theta*(he+2000) +
(1600-(1 *so + .9*co + .6*wh + 60*da + .3*he))*5+ (2100-(1.4*so +
1.2*co + .7*wh +60*da + .3*he))*5.5 +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000 ;
```

$$v := \theta \to 21.30\, so + 16.40\, co + 13.90\, wh + 70.0\, da - 3.90\, he + \theta\, (he + 2000) + 60300.00$$

Also, a form of scenario modified to put any values will be useful for checking purposes.

```
scenario := proc(a,b,c,d,e,f,g)
local net,constraints,soln,wsh,sfh,so,
co,wh,he,da;
net := a*so + b*co + c*wh +
d*(da+30) + e*(he+2000) +
(4000-wsh)*f+(4500-sfh)*g +
20000 - 1500*da - 3*he +
.9*(35000 + 1500*da) + .75*(5000 + 3*he)
-40000;
constraints := [
wsh <= 4000,
sfh <= 4500,
1500*da + 3*he <= 20000,
30+da <= 42,
2000+he <= 5000,
co >= (da+30),
wh >= .05*(he+2000),
so + co + wh + 2*(da+30) + .0 * (he+2000) <= 640,
1 *so + .9*co + .6*wh + 6*10*(da+30) + 6*.05*(he+2000) = wsh,
1.4*so + 1.2*co + .7*wh + 6*10*(da+30) + 6*.05*(he+2000) = sfh];
soln:=maximize(net,constraints,NONNEGATIVE);
subs(soln,matrix(8,2,[networth,net,
corn, co, soybeans, so,
wheat,wh,cows,da,hens,he,
winterspring,wsh,summerfall,sfh ])); end:
```

We saw how to use the scenario function to track the networth as a function of theta the value of a hen on Monday. Basically, by just tabulating at increments of 1 cent we found the planting scedules changed in at 4.28, 4.32, and 4.38 roughly, and that in between the networth increased linearly, but with different slopes in the different intervals.

```
scenario(34,27.5,20.75,850,4.37,5,5.5);
```

$$\left[\begin{array}{cc} \textit{networth} & 80904.47253 \\ \textit{corn} & 39.78887536 \\ \textit{soybeans} & 332.0219245 \\ \textit{wheat} & 188.6114495 \\ \textit{cows} & 9.788875355 \\ \textit{hens} & 1772.228989 \\ \textit{winterspring} & 4000.000000 \\ \textit{summerfall} & 4163.606577 \end{array}\right]$$

The function showgt shows the final modified tableau. We have to inspect this manually to decide the appropriate gauss matrix to premultiply by.

```
showgt := proc(theta,g)
local net, eqns, T;
net := v(theta);
eqns := [z-net+68800=68800,
1500*da + 3*he + sl1 = 20000,
da + sl2 = 12,
he +sl3 = 3000,
co - da - sp1 + ar1 = 30,
wh - .05*he -sp2 + ar2 = 100,
so + co + wh + 2*da + sl4 = 580,
so + .9*co + .6*wh + 60*da + .3*he +sl5 = 1600,
1.4* so + 1.2*co + .7*wh + 60*da + .3*he +sl6 = 2100];
T := genmatrix(eqns,
[z,so,co,wh,da,he,sl1,sl2,sl3,ar1,ar2,sl4,sl5,sl6,sp1,sp2],flag);
evalm(g&*T); end:
```

Now we are ready to go.

parametric programming on the value of a hen

```
ok := showgt(theta,g);
```

```
ok :=
    [1., 0, 0, 0, 0, −1. θ + 4.2700, 0, 22.50, 0, −4.90, −7.40, 21.30, 0, 0, 4.90, 7.40,
    2000. θ + 72037.00]
    [0, 0, 0, 0, 0, 3., 1., −1500., 0, 0, 0, 0, 0, 0, 0, 0, 2000.]
    [0, 0, 0, 0, 1., 0, 0, 1., 0, 0, 0, 0, 0, 0, 0, 0, 12.]
    [0, 0, 0, 0, 0, 1., 0, 0, 1., 0, 0, 0, 0, 0, 0, 0, 3000.]
    [0, 0, 1., 0, 0, 0, 0, 1., 0, 1., 0, 0, 0, 0, −1., 0, 42.]
    [0, 0, 0, 1., 0, −.05, 0, 0, 0, 0, 1., 0, 0, 0, 0, −1., 100.]
    [0, 1., 0, 0, 0, .05, 0, −3., 0, −1., −1., 1., 0, 0, 1., 1., 414.]
    [0, 0, 0, 0, 0, .280, 0, −57.9, 0, .1, .4, −1., 1., 0, −.1, −.4, 368.2]
    [0, 0, 0, 0, 0, .265, 0, −57.0, 0, .2, .7, −1.4, 0, 1., −.2, −.7, 680.0]
```

```
ineqns := row(ok,1);
sol :=
solve({seq(ineqns[i]>=0,i=2..9),seq(ineqns[j]>=0,j=12..16)},{theta}%
> ;
```

```
ineqns := [1., 0, 0, 0, 0, −1. θ + 4.2700, 0, 22.50, 0, −4.90, −7.40, 21.30, 0, 0, 4.90, 7.40,
    2000. θ + 72037.00]
                              sol := {θ ≤ 4.270000000}
```

So for theta <= 4.27, the optimum networth is 72037+2000*theta.

```
2000/3.,414/.05,368/.28,680/.265;
```

$$666.6666667,\ 8280.000000,\ 1314.285714,\ 2566.037736$$

```
g1 :=evalm(gauss(col(ok,6),2)) ;
```

$$g1 := \begin{bmatrix} 1 & .3333333333\,\theta - 1.423333333 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .3333333333 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -.3333333333 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & .01666666667 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -.01666666667 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & -.09333333332 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -.08833333332 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

```
ok2 := evalm(g1&*ok);
```

$$\begin{array}{l} ok2 := \\ \quad [1.\,,0\,,0\,,0\,,0\,,-.1\,10^{-9}\,\theta + .1\,10^{-8}\,,\ .3333333333\,\theta - 1.423333333\,, \\ \quad 2157.500000 - 500.0000000\,\theta\,,\ 0\,,\ -4.90\,,\ -7.40\,,\ 21.30\,,\ 0\,,\ 0\,,\ 4.90\,,\ 7.40\,, \\ \quad 2666.666667\,\theta + 69190.33333] \\ \quad [0\,,0\,,0\,,0\,,0\,,\ .9999999999\,,\ .3333333333\,,\ -500.0000000\,,\ 0\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,, \\ \quad 666.6666666] \\ \quad [0\,,0\,,0\,,0\,,1.\,,0\,,0\,,1.\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,12.] \\ \quad [0\,,0\,,0\,,0\,,0\,,\ .1\,10^{-9}\,,\ -.3333333333\,,\ 500.0000000\,,\ 1.\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,, \\ \quad 2333.333333] \\ \quad [0\,,0\,,1.\,,0\,,0\,,0\,,0\,,1.\,,0\,,1.\,,0\,,0\,,0\,,0\,,-1.\,,0\,,42.] \\ \quad [0\,,0\,,0\,,1.\,,0\,,\ .1\,10^{-10}\,,\ .01666666667\,,\ -25.00000001\,,\ 0\,,0\,,1.\,,0\,,0\,,0\,,0\,,-1.\,, \\ \quad 133.3333333] \\ \quad [0\,,1.\,,0\,,0\,,0\,,\ -.1\,10^{-10}\,,\ -.01666666667\,,\ 22.00000001\,,\ 0\,,-1.\,,-1.\,,1.\,,0\,,0\,,1.\,,1.\,, \\ \quad 380.6666667] \\ \quad [0\,,0\,,0\,,0\,,0\,,0\,,\ -.09333333332\,,\ 82.1000000\,,\ 0\,,.1\,,.4\,,-1.\,,1.\,,0\,,-.1\,,-.4\,, \\ \quad 181.5333334] \\ \quad [0\,,0\,,0\,,0\,,0\,,0\,,\ -.08833333332\,,\ 75.5000000\,,\ 0\,,.2\,,.7\,,-1.4\,,0\,,1.\,,-.2\,,-.7\,, \\ \quad 503.3333334] \end{array}$$

```
evalf(evalm(ok2),3);
```

$$\begin{array}{l} [1.\,,0\,,0\,,0\,,0\,,-.1\,10^{-9}\,\theta + .1\,10^{-8}\,,\ .333\,\theta - 1.42\,,\ 2160. - 500.\,\theta\,,\ 0\,,\ -4.90\,,\ -7.40\,, \\ \quad 21.3\,,\ 0\,,\ 0\,,\ 4.90\,,\ 7.40\,,\ 2670.\,\theta + 69200.] \\ \quad [0\,,0\,,0\,,0\,,0\,,1.00\,,\ .333\,,\ -500.\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,667.] \\ \quad [0\,,0\,,0\,,0\,,1.\,,0\,,0\,,1.\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,12.] \\ \quad [0\,,0\,,0\,,0\,,0\,,\ .1\,10^{-9}\,,\ -.333\,,\ 500.\,,\ 1.\,,0\,,0\,,0\,,0\,,0\,,0\,,0\,,2330.] \\ \quad [0\,,0\,,1.\,,0\,,0\,,0\,,0\,,1.\,,0\,,1.\,,0\,,0\,,0\,,0\,,-1.\,,0\,,42.] \\ \quad [0\,,0\,,0\,,1.\,,0\,,\ .1\,10^{-10}\,,\ .0167\,,\ -25.0\,,0\,,0\,,1.\,,0\,,0\,,0\,,0\,,-1.\,,133.] \\ \quad [0\,,1.\,,0\,,0\,,0\,,\ -.1\,10^{-10}\,,\ -.0167\,,\ 22.0\,,0\,,-1.\,,-1.\,,1.\,,0\,,0\,,1.\,,1.\,,381.] \\ \quad [0\,,0\,,0\,,0\,,0\,,0\,,\ -.0933\,,\ 82.1\,,0\,,.1\,,.4\,,-1.\,,1.\,,0\,,-.1\,,-.4\,,182.] \\ \quad [0\,,0\,,0\,,0\,,0\,,0\,,\ -.0883\,,\ 75.5\,,0\,,.2\,,.7\,,-1.4\,,0\,,1.\,,-.2\,,-.7\,,503.] \end{array}$$

```
ineqns2 := row(ok2,1);
sol2 :=
solve({seq(ineqns2[i]>=0,i=7..9),seq(ineqns2[j]>=0,j=12..16)},{thet
a});
```

$$\mathit{ineqns2} := [1., 0, 0, 0, 0, -.1\,10^{-9}\,\theta + .1\,10^{-8}, .3333333333\,\theta - 1.423333333, 2157.500000 - 500.0000000\,\theta, 0, -4.90, -7.40, 21.30, 0, 0, 4.90, 7.40, 2666.666667\,\theta + 69190.33333]$$

$$\mathit{sol2} := \{4.269999999 \leq \theta, \theta \leq 4.315000000\}$$

And for theta between 4.27 and 4.315, the optimum networth is 69190.33 + 2666.67*theta

```
submatrix(ok2,[seq(i,i=1..8)],[8 ,17] );
```

$$\begin{bmatrix} 2157.500000 - 500.0000000\,\theta & 2666.666667\,\theta + 69190.33333 \\ -500.0000000 & 666.6666666 \\ 1. & 12. \\ 500.0000000 & 2333.333333 \\ 1. & 42. \\ -25.00000001 & 133.3333333 \\ 22.00000001 & 380.6666667 \\ 82.1000000 & 181.5333334 \end{bmatrix}$$

```
g2 := gauss(col(ok2,8),8);
```

$$g2 := \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -26.27892815 + 6.090133985\,\theta & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 6.090133985 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -.01218026797 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -6.090133985 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & -.01218026797 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & .3045066994 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -.2679658955 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .01218026797 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -.9196102317 & 1 \end{bmatrix}$$

```
ok3 := evalm(g2&*ok2);

evalf(evalm(submatrix(ok3, 1..8 , 7..17
)),3);

ineqns3 := row(ok3,1);
sol3 := solve({seq(ineqns3[i]>=0,i=[7,9..16]) },{theta});
```

$$\mathit{ineqns3} := [1., 0, 0, 0, 0, -.1\,10^{-9}\,\theta + .1\,10^{-8}, -.2350791719\,\theta + 1.029366627, -.1\,10^{-5} + .2\,10^{-6}\,\theta, 0, -7.527892815 + .6090133985\,\theta, -17.91157126 + 2.436053594\,\theta, 47.57892815 - 6.090133985\,\theta, -26.27892815 + 6.090133985\,\theta, 0, 7.527892815 - .6090133985\,\theta, 17.91157126 - 2.436053594\,\theta, 3772.228990\,\theta + 64419.83191]$$

$$\mathit{sol3} := \{\theta \leq 4.378808291\}$$

And for theta between 4.315 and 4.379, the optimum networth is 64419.83+3772.23*theta

```
subs(theta=4.38,evalm(ineqns3));
```

$$[1., 0, 0, 0, 0, .562\,10^{-9}, -.000280146, -.124\,10^{-6}, 0, -4.860414130, -7.24165652, 20.90414130, .39585870, 0, 4.860414130, 7.24165652, 80942.19489]$$

```
row(ok3,7);
```

$$[0,\ 1.,\ 0,\ 0,\ 0,\ -.1\,10^{-10},\ .00834348357,\ -.1\,10^{-7},\ 0,\ -1.026796590,\ -1.107186358,\ 1.267965896,\ -.2679658955,\ 0,\ 1.026796590,\ 1.107186358,\ 332.0219245]$$

```
rc
:=getpiv(subs(theta=4.38,evalm(ok3)),[7,8,9],{11,12});
```

$$rc := [4,\ 7]$$

```
ok4
:=evalm(gauss(col(ok3,rc[2]),rc[1])&*ok3);
```

$$ok4 :=$$

$$[1.\,,\ 0\,,\ 0\,,\ 0\,,\ 0\,,\ -.1\,10^{-19}\,\theta + .5621191709\,10^{-9}\,,\ 0\,,\ -.1242383418\,10^{-6}\,,\ .9999999999\,\theta - 4.378808291\,,\ -4.861139896 + .1\,10^{-9}\,\theta\,,\ -7.24455959\,,\ 20.91139896 - .1\,10^{-8}\,\theta\,,\ .38860104 + .1\,10^{-8}\,\theta\,,\ 0\,,\ 4.861139896 - .1\,10^{-9}\,\theta\,,\ 7.24455959\,,\ 5000.000000\,\theta + 59043.65803]$$

$$[0\,,\ 0\,,\ 0\,,\ 0\,,\ 0\,,\ 1.000000000\,,\ 0\,,\ 0\,,\ .9999999999\,,\ .1\,10^{-9}\,,\ 0\,,\ -.1\,10^{-8}\,,\ .1\,10^{-8}\,,\ 0\,,\ -.1\,10^{-9}\,,\ 0\,,\ 3000.000000]$$

$$[0\,,\ 0\,,\ 0\,,\ 0\,,\ 1.\,,\ -.4835924002\,10^{-12}\,,\ 0\,,\ .9671848004\,10^{-9}\,,\ -.004835924002\,,\ .001727115714\,,\ .006908462862\,,\ -.01727115714\,,\ .01727115714\,,\ 0\,,\ -.001727115714\,,\ -.006908462862\,,\ 3.851468058]$$

$$[0\,,\ 0\,,\ 0\,,\ 0\,,\ 0\,,\ .4253886007\,10^{-9}\,,\ .9999999999\,,\ -.8507772014\,10^{-6}\,,\ 4.253886007\,,\ -2.590673574\,,\ -10.36269430\,,\ 25.90673574\,,\ -25.90673574\,,\ 0\,,\ 2.590673574\,,\ 10.36269430\,,\ 5222.797919]$$

$$[0\,,\ 0\,,\ 1.\,,\ 0\,,\ 0\,,\ -.4835924002\,10^{-12}\,,\ 0\,,\ .9671848004\,10^{-9}\,,\ -.004835924002\,,\ 1.001727116\,,\ .006908462862\,,\ -.01727115714\,,\ .01727115714\,,\ 0\,,\ -1.001727116\,,\ -.006908462862\,,\ 33.85146805]$$

$$[0\,,\ 0\,,\ 0\,,\ 1.\,,\ 0\,,\ .1500000000\,10^{-10}\,,\ 0\,,\ 0\,,\ .05000000002\,,\ 0\,,\ 1.000000000\,,\ 0\,,\ 0\,,\ 0\,,\ 0\,,\ -1.000000000\,,\ 250.0000000]$$

$$[0\,,\ 1.\,,\ 0\,,\ 0\,,\ 0\,,\ -.1354922280\,10^{-10}\,,\ .1\,10^{-11}\,,\ -.2901554398\,10^{-8}\,,\ -.03549222801\,,\ -1.005181348\,,\ -1.020725388\,,\ 1.051813472\,,\ -.0518134715\,,\ 0\,,\ 1.005181348\,,\ 1.020725388\,,\ 288.4455959]$$

$$[0\,,\ 0\,,\ 0\,,\ 0\,,\ 0\,,\ .4835924002\,10^{-12}\,,\ 0\,,\ .9999999990\,,\ .004835924002\,,\ -.001727115714\,,\ -.006908462862\,,\ .01727115714\,,\ -.01727115714\,,\ 0\,,\ .001727115714\,,\ .006908462862\,,\ 8.148531942]$$

$$[0\,,\ 0\,,\ 0\,,\ 0\,,\ 0\,,\ .1064766827\,10^{-11}\,,\ 0\,,\ -.2212953365\,10^{-7}\,,\ .01064766827\,,\ .1015544042\,,\ .3062176167\,,\ -.4155440419\,,\ -.9844559581\,,\ 1.\,,\ -.1015544042\,,\ -.3062176167\,,\ 349.4663210]$$

```
ineqns3 := row(ok4,1);
sol3 := solve({seq(ineqns3[i]>=0,i=[7,9..16]) },{theta});
```

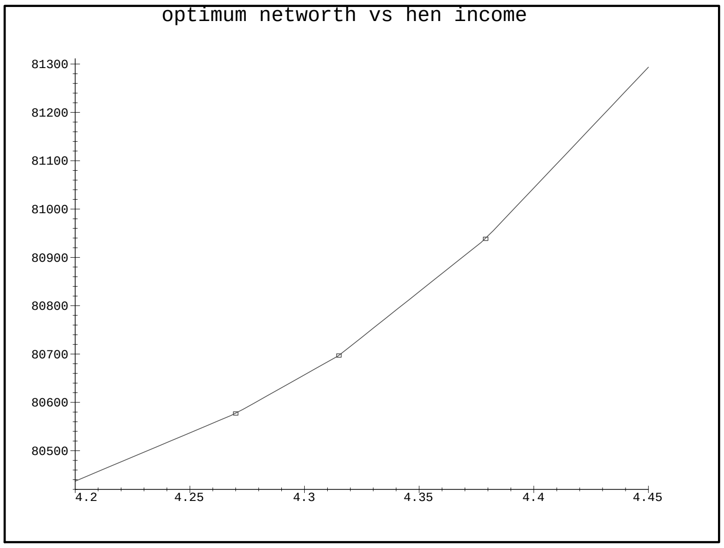
$$ineqns3 := [1.,\ 0,\ 0,\ 0,\ 0,\ -.1\,10^{-19}\,\theta + .5621191709\,10^{-9},\ 0,\ -.1242383418\,10^{-6},\ .9999999999\,\theta - 4.378808291,\ -4.861139896 + .1\,10^{-9}\,\theta,\ -7.24455959,\ 20.91139896 - .1\,10^{-8}\,\theta,\ .38860104 + .1\,10^{-8}\,\theta,\ 0,\ 4.861139896 - .1\,10^{-9}\,\theta,\ 7.24455959,\ 5000.000000\,\theta + 59043.65803]$$

$$sol3 :=$$

And beyond 4.378, the optimum networth is 5000*theta+59043.65803
We can define the networth function completely now as a function of the value of a hen.

```
net := proc(theta)
if theta <= 4.27 then 72037.+2000*theta
elif theta <= 4.315 then 69190.33 + 2666.67*theta
elif theta <= 4.379 then 64419.83+3772.23*theta
else 5000*theta+59043.65803 fi end:
```

```
plt := plot(net,4.20..4.45):
pts
:=plot({[4.27,net(4.27)],[4.315,net(4.315)],[4.379,net(4.379)]},styl
e=point,symbol=box):
plots[display]([plt,pts],title=‘optimum
networth vs hen income‘);
```



22 parametric programming example

```
T := matrix(4,7,[1,-2,-1,0,0,0,0,
0,1,1,1,0,0,10,
0,1/3,1,0,1,0,9,
0,1,1/2,0,0,1,9]);
```

$$T := \begin{bmatrix} 1 & -2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 10 \\ 0 & \frac{1}{3} & 1 & 0 & 1 & 0 & 9 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 1 & 9 \end{bmatrix}$$

```
Tf := matmax(T,[1,4,5,6]);
```

$$\begin{bmatrix} 1 & -2 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 10 \\ 0 & \frac{1}{3} & 1 & 0 & 1 & 0 & 9 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 1 & 9 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 18 \\ 0 & 0 & \frac{1}{2} & 1 & 0 & -1 & 1 \\ 0 & 0 & \frac{5}{6} & 0 & 1 & \frac{-1}{3} & 6 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 1 & 9 \end{bmatrix}$$

$$Tf := \left[\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 2 & 18 \\ 0 & 0 & \frac{1}{2} & 1 & 0 & -1 & 1 \\ 0 & 0 & \frac{5}{6} & 0 & 1 & \frac{-1}{3} & 6 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 1 & 9 \end{bmatrix}, [1, 4, 5, 2]\right]$$

```
g := submatrix(Tf[1],[1,2,3,4],[1,4,5,6]);
```

$$g := \begin{bmatrix} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & \frac{-1}{3} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

```
G := evalm(T):G[1,2]:= -theta;
```

$$G_{1,2} := -\theta$$

```
Tn := evalm(g&*G);
```

$$Tn := \begin{bmatrix} 1 & -\theta + 2 & 0 & 0 & 0 & 2 & 18 \\ 0 & 0 & \frac{1}{2} & 1 & 0 & -1 & 1 \\ 0 & 0 & \frac{5}{6} & 0 & 1 & \frac{-1}{3} & 6 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 1 & 9 \end{bmatrix}$$

```
T1 := evalm(gauss(col(Tn,2),4)&*Tn);
```

$$T1 := \begin{bmatrix} 1 & 0 & \frac{1}{2}\theta - 1 & 0 & 0 & \theta & 9\,\theta \\ 0 & 0 & \frac{1}{2} & 1 & 0 & -1 & 1 \\ 0 & 0 & \frac{5}{6} & 0 & 1 & \frac{-1}{3} & 6 \\ 0 & 1 & \frac{1}{2} & 0 & 0 & 1 & 9 \end{bmatrix}$$

As theta increases from 2 on the optimum z is 9 *theta..

As theta decreases from 2 , want to bring the third column into solution.

```
T2 :=evalm(gauss(col(T1,3),2)&*T1);
```

$$T2 := \begin{bmatrix} 1 & 0 & 0 & -\theta + 2 & 0 & 2\,\theta - 2 & 8\,\theta + 2 \\ 0 & 0 & 1 & 2 & 0 & -2 & 2 \\ 0 & 0 & 0 & \frac{-5}{3} & 1 & \frac{4}{3} & \frac{13}{3} \\ 0 & 1 & 0 & -1 & 0 & 2 & 8 \end{bmatrix}$$

Now when theta gets down to 1 the 6th column is going negative so bring that into solution

```
T3 :=evalm(gauss(col(T2,6),3)&*T2);
```

$$T3 := \begin{bmatrix} 1 & 0 & 0 & \frac{3}{2}\theta - \frac{1}{2} & -\frac{3}{2}\theta + \frac{3}{2} & 0 & \frac{3}{2}\theta + \frac{17}{2} \\ 0 & 0 & 1 & \frac{-1}{2} & \frac{3}{2} & 0 & \frac{17}{2} \\ 0 & 0 & 0 & \frac{-5}{4} & \frac{3}{4} & 1 & \frac{13}{4} \\ 0 & 1 & 0 & \frac{3}{2} & \frac{-3}{2} & 0 & \frac{3}{2} \end{bmatrix}$$

Below below 1 the columns stay positive so we are done!

23 Transportation problems.

This worsheet implements the streamlined simplex method for the transportation problem as described in Hillier and Lieberman, Chapter 8.

The general transportation problem goes like this: There are m warehouses (sources) and n outlets (sinks). There is an m-dimensional supply vector s (s[i] is the supply available at the ith warehouse), an n-dimensional demand vector d (d[j] is the demand from the jth outlet), an mxn cost matrix c (c[i,j] is the cost of shipping a unit from warehouse i to outlet j), and an mxn decision matrix x (x[i,j] is the amount you decide to ship from the ith warehouse to the jth outlet. The assumption is that the total supply is (at least) the total demand and the goal is to find the shipping schedule which minimizes the total shipping costs.

```
restart;
with(linalg):
```

```
Warning, new definition for norm

Warning, new definition for trace
```

```
with(simplex):
```

```
Warning, new definition for basis

Warning, new definition for maximize

Warning, new definition for minimize

Warning, new definition for pivot
```

First we'll make up a random cost matrix for a 4 by 5 problem say.

```
c := map(abs,randmatrix(4,5));
c[1,1] := M; c[3,4] := M; op(c);
```

and put some 'big M's' in a few entries, to make some routes prohibitively expensive. Next a procedure to generate a random supply (demand) vector. s is the desired sum and n is the number of sources (sinks).

```
randsupply := proc(s,n)
local i,j,ls,l;
l := [seq(rand(),i=1..n)];
ls := evalf(evalm( s/sum(l[j],j=1..n)*l));
convert(ls,list) end:
s := randsupply(100,4) ;
d := randsupply(100,5);
printlevel:=1;
```

We can avoid the use of artificial variables in transportation problems, because it is easy to come up with feasible solutions. For example, the northwest corner rule is a method for generating a feasible solution to a random transportation problem. The idea is to start filling in the decision matrix from the upper lefthand corner, filling the demand with the supply available as you work down the warehouse list. If total supply is sufficient to fill total demand, a feasible solution is obviously generated. A Maple word to implement the northwest corner rule has inputs the cost matrix, and supply and demand vectors. The outputs would be the decision matrix, together with a list of the basic variables.

```
nwcrule := proc(c,s,d )
```

```
   local bv,x, i,j,rw,cl,n,m,fin;
   n := rowdim(c):
   m := coldim(c):
   x := matrix(n,m,[0$n*m]);
   bv := NULL: rw := 1:
   for cl from 1 to m do
   while rw < n+1 and sum(x[i,cl],i=1..rw-1 ) < d[cl] do
   x[rw,cl] := min(d[cl]-sum(x[i,cl],i=1..rw-1),
   s[rw]-sum(x[rw,j],j=1..cl-1));
   bv := bv,[rw,cl];
   rw := rw+1;
   od;
   if rw > n or s[rw]-sum(x[rw,j],j=1..cl) > 0 then rw := rw-1 fi
   od;
   [evalm(x),{bv}];
   end:
   c := matrix(3,3,[7,6,20,10,15,5,8,10,3]);
   s := [15,5,25]:  d := [10,15,20]:
   sol := nwcrule(c,s,d ) ;
   x := sol[1];
   bv := sol[2];
```

getuv finds the shadow prices that accompany the current basis. The idea

```
   getuv := proc(c,x,bv,‘u‘,‘v‘)
   local eqs, i, j,t,t1,rw;
   eqs := {}:
   for i from 1 to rowdim(x) do
   for j from 1 to coldim(x) do
   if member([i,j],bv) then
   eqs := eqs union {c[i,j]=u[i]+v[j]} fi;
   od od;
   #t := 0;
   #for i from 1 to rowdim(x) do
   # t1 := nops({seq(x[i,j],j=1..coldim(x))});
   #if t < t1 then t := t1; rw := i fi;
   #od;
   #u[rw]:= 0;
   # assign(solve( eqs ,
   #{seq(u[i],i=1..rw-1),
   # seq(u[i],i=rw+1..rowdim(x)), #seq(v[j],j=1..coldim(x))}));
   v[1]:=0;
   assign(solve( eqs ,
{seq(u[i],i= 1..rowdim(x)), seq(v[j],j=2..coldim(x))}));
   end:
   u := ’u’:  v := ’v’:
   getuv(c,x,bv,u,v);
```

```
Error, (in rowdim) expecting a matrix
```

```
   op(v);
   op(u);
```

checkopt returns true if the current solution is optimal, else returns the entering variable.

```
shad := evalm(c); for i from 1 to 3 do for j
from 1 to 3 do
shad[i,j]:= shad[i,j]-u[i]-v[j] od od:
evalm(shad);
checkopt := proc(c,x,bv,u,v)
local i,j ,t,t1,inc,shad ;
t := 0;
shad := matrix(rowdim(c),coldim(c),[seq(0,i=1..rowdim(c)*coldim(c))]);
inc := true:  for i from 1 to rowdim(c) do
for j from 1 to coldim(c) do
t1 := subs(M=infinity,c[i,j]-u[i]-v[j]);
shad[i,j]:=t1;
if not member([i,j],bv)
then if t1 < t then
t := t1; inc := [i,j] fi fi;
od od;
print(‘reduced costs‘);
print(evalm(shad));
RETURN(inc); end:
ev := checkopt(c,x,bv,u,v);
```

getcycle takes the current basis and entering variable and returns the cycle in bv union ev containing ev.

```
getcycle := proc(bv,ev)
local paths, lev, fin, newpaths, i,j, p, v;
paths := table() ;
paths[1] := [ev];
lev := 0; fin := false;
while not fin do
newpaths := table(); i := 0;
for j from 1 to nops({indices(paths)}) do
for v in bv minus {op(paths[j])} do
if v[lev mod 2 +1] = paths[j][-1][lev mod 2 +1]
then i := i+1;
newpaths[i] := [op(paths[j]),v];
if v[lev+1 mod 2 +1] = ev[lev+1 mod 2 +1] then
RETURN(newpaths[i]) fi;
fi;
od od;
paths := op(newpaths);
lev := lev+1;
od;
end:
```

This is the iteration step.

```
cy := getcycle(bv,ev);
t := infinity; r := 0,0; for i from 1 to nops(cy)/2 do
r1 := op(cy[2*i ]);
t1 := subs(M=infinity, x[r1]);
if t1 < t then t := t1; r := r1; fi
od ; lv := [r];
for i from 1 to nops(cy) do
x[op(cy[i])] := x[op(cy[i])]+(-1)^(i+1)*t od;
```

```
bv := {ev} union (bv minus {lv});
u := 'u':  v := 'v':  getuv(c,x,bv,u,v);
ev := checkopt(c,x,bv,u,v);
```

We need a cost function.

```
cost := proc(c,x,bv)
convert([seq(c[op(bv[i])]*x[op(bv[i])],i=1..nops(bv))],`+`); end:
cost(c,x,bv);
```

Finally, putting this all together in the word fasttransport.

```
fasttransport := proc(c,s,d,prt)
local sol, x, bv, fin, u, v, ev, st,cy,cyc, t, r, i, r1, t1, lv;
sol := nwcrule(c,s,d ) ;
x := sol[1];
bv := sol[2];
fin := false;
while not fin do
if nargs()=4 then print(evalm(x)); print(cost(c,x,bv)) fi;
u := 'u':  v := 'v':
getuv(c,x,bv,u,v);
ev := checkopt(c,x,bv,u,v);
if ev=true then RETURN([bv,op(x),cost(c,x,bv)]) fi;
cy := getcycle(bv,ev);
cyc := matrix(nops(s),nops(d),[seq(0,i=1..nops(s)*nops(d))]);
for st in bv do cyc[op(st)]:=1 od:
for st in cy do cyc[op(st)]:=`y` od;
cyc[op(ev)]:=`e`;
print( evalm(cyc));
t := infinity; r := 0,0; for i from 1 to nops(cy)/2 do
r1 := op(cy[2*i ]);
t1 := subs(M=infinity, x[r1]);
if t1 < t then t := t1; r := r1; fi
od ; lv := [r];
for i from 1 to nops(cy) do
x[op(cy[i])] := x[op(cy[i])]+(-1)^(i+1)*t od;
bv := {ev} union (bv minus {lv});
od; end:
```

Check this on a problem.

```
c := matrix(3,3,[5,6,1,3,2,2,5,2,5 ]);
```

$$c := \begin{bmatrix} 5 & 6 & 1 \\ 3 & 2 & 2 \\ 5 & 2 & 5 \end{bmatrix}$$

```
d := [1,1,1]:  s := [1,1,1]:
time();
```

61.049

```
t := time():sol := fasttransport(c,s,d,p);
ftime := time()-t;
```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$12$$

reduced costs

$$\begin{bmatrix} 0 & 0 & -5 \\ 2 & 0 & 0 \\ 1 & -3 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & y & e \\ 0 & y & y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$12$$

reduced costs

$$\begin{bmatrix} 0 & 5 & 0 \\ -3 & 0 & 0 \\ -4 & -3 & 0 \end{bmatrix}$$

$$\begin{bmatrix} y & 0 & y \\ 0 & 1 & 1 \\ e & 0 & y \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$8$$

reduced costs

$$\begin{bmatrix} 0 & 5 & 0 \\ -3 & 0 & 0 \\ 0 & 1 & 4 \end{bmatrix}$$

$$\begin{bmatrix} y & 0 & y \\ e & 1 & y \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$8$$

reduced costs

$$\begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 3 \\ 0 & -2 & 4 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 1 \\ y & y & 0 \\ y & e & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$6$$

$$\mathit{reduced\ costs}$$

$$\begin{bmatrix} 0 & 2 & 0 \\ 0 & 0 & 3 \\ 2 & 0 & 6 \end{bmatrix}$$

$$sol := [\{[2, 1], [2, 2], [1, 1], [1, 3], [3, 2]\}, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, 6]$$

$$ftime := .810$$

We need to check this against the standard setup a transportation problem.

```
cost(c,sol[2],sol[1]);
```

$$6$$

```
slowtransport := proc(c,s,d)
local m, n, x, eqs, i, j, obj,soln;
m := rowdim(c):  n := coldim(c):
x := matrix(m,n);
eqs := NULL;
for i from 1 to m do
eqs := eqs,sum(x[i,j],j=1..n)<= s[i],
sum(x[i,j],j=1..n)>= s[i] od;
i := 'i':
for j from 1 to n do
eqs := eqs,sum(x[i,j],i=1..m)<=d[j],
sum(x[i,j],i=1..m)>=d[j] od;
eqs := {eqs};
j := 'j':
obj := sum(sum(c[i,j]*x[i,j],j=1..n),i=1..m);
soln:=minimize(obj,eqs,NONNEGATIVE);
assign(soln);
[subs(soln,obj),evalm(x)];
end:
```

```
Digits := 7;
```

```
t := time():sln := slowtransport(subs(M=10^3
,op(c)),s,d);
time()-t;
```

$$sln := [6, \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}]$$

$$.410$$

Well, so much for the streamlinedness of the fast transportation algorithm. Slow is faster than fast in this implementation. We could try a larger problem to see what happens.

```
c := map(abs,randmatrix(5,7));
s := randsupply(100,5);
d := randsupply(100,7);
t := time():sln := fasttransport(subs(M=10^3
,op(c)),s,d);
time()-t;
Digits:=7;define_zero(10^(-7));
t := time():sln := slowtransport(subs(M=10^3
,op(c)),s,d);
time()-t;
```

What do you make of this?

Solution: The problem here is that fasttransport is working on a problem with no solution, most of the time, because of the use of evalf.

Assignment problems.

If you have n people to do n jobs and it costs c_{ij} dollars for the ith person to do the jth job, then you have an assignment problem. It can be treated as a transportation problem where the supply and demand equations all have right hand sides 1. A basic feasible solution will have exactly n assignments of value 1, and the other n-1 basic variables will be assigned value 0. This will cause a lot of 'wasted iterations' when we apply the transportation simplex algorithm, but that's ok. They're cheap.

I have expanded the definitions of the words fastransport, checkopt, and get cycle so as to print out more intermediate information to enable you to see the work going into getting the entering variable, the cycle, and the leaving variable.

```
   restart;
   with(linalg):
   with(simplex):
```

The northwest corner rule is unchanged.

```
   nwcrule := proc(c,s,d )
   local bv,x, i,j,rw,cl,n,m,fin;
   n := rowdim(c):
   m := coldim(c):
   x := matrix(n,m,[0$n*m]);
   bv := NULL: rw := 1:
   for cl from 1 to m do
   while rw < n+1 and sum(x[i,cl],i=1..rw-1 ) < d[cl] do
   x[rw,cl] := min(d[cl]-sum(x[i,cl],i=1..rw-1),
   s[rw]-sum(x[rw,j],j=1..cl-1));
   bv := bv,[rw,cl];
   rw := rw+1;
   od;
   if rw > n or s[rw]-sum(x[rw,j],j=1..cl) > 0 then rw := rw-1 fi
   od;
   [evalm(x),{bv}];
   end:
```

getuv finds the shadow prices that accompany the current basis. These enable you to choose the entering variable. Note that as we discussed in class, the u[i] and v[j]'s are not unique and so you can set any one of them to zero and then determine the others. We can set v[1]=0.

```
   getuv := proc(c,x,bv,`u`,`v`)
   local eqs, i, j,t,t1,rw;
   eqs := {}:
   for i from 1 to rowdim(x) do
   for j from 1 to coldim(x) do
   if member([i,j],bv) then
   eqs := eqs union {c[i,j]=u[i]+v[j]} fi;
   od od;
   v[1]:=0;
   assign(solve( eqs ,
{seq(u[i],i= 1..rowdim(x)), seq(v[j],j=2..coldim(x))}));
   end:
```

Checkopt returns true if the current solution is optimal, else returns the entering variable.

```
   checkopt := proc(c,x,bv,u,v,prt)
   local i,j ,t,t1,inc,shad ;
   t := 0;
```

```
shad := matrix(rowdim(c),coldim(c),[seq(0,i=1..rowdim(c)*coldim(c))]);
inc := true:  for i from 1 to rowdim(c) do
for j from 1 to coldim(c) do
t1 := subs(M=infinity,c[i,j]-u[i]-v[j]);
if member ([i,j],bv) then shad[i,j]:=`b`
else shad[i,j]:= t1 fi;
if not member([i,j],bv)
then if t1 < t then
t := t1; inc := [i,j] fi fi;
od od;
if nargs() = 5 then print(`reduced costs`);
print(evalm(shad)); fi;
RETURN(inc); end:
```

getcycle takes the current basis and entering variable and returns the cycle in bv union ev containing ev.

```
getcycle := proc(bv,ev)
local paths, lev, fin, newpaths, i,j, p, v;
paths := table() ;
paths[1] := [ev];
lev := 0; fin := false;
while not fin do
newpaths := table(); i := 0;
for j from 1 to nops({indices(paths)}) do
for v in bv minus {op(paths[j])} do
if v[lev mod 2 +1] = paths[j][-1][lev mod 2 +1]
then i := i+1;
newpaths[i] := [op(paths[j]),v];
if v[lev+1 mod 2 +1] = ev[lev+1 mod 2 +1] then
RETURN(newpaths[i]) fi;
fi;
od od;
paths := op(newpaths);
lev := lev+1;
od;
end:
```

We need a cost function.

```
cost := proc(c,x,bv)
convert([seq(c[op(bv[i])]*x[op(bv[i])],i=1..nops(bv))],`+`); end:
```

Finally, putting this all together in the word fasttransport2.

```
fasttransport2 := proc(c,s,d,prt)
local sol, x,bv,iter, fin, u, v, ev, st,cy,cyc, t, r, i, r1, t1, lv;
sol := nwcrule(c,s,d ) ;
x := sol[1];
bv := sol[2];
fin := false;
iter := 0;
while not fin do
if nargs()=4 then print(evalm(x)); print(cost(c,x,bv)) fi;
u := 'u':  v := 'v':
getuv(c,x,bv,u,v);
ev := checkopt(c,x,bv,u,v);
```

```
if ev=true then RETURN([bv,op(x),cost(c,x,bv)]) fi;
iter := iter+1; print(`iteration `,iter);
cy := getcycle(bv,ev);
cyc := matrix(nops(s),nops(d),[seq(0,i=1..nops(s)*nops(d))]);
for st in bv do cyc[op(st)]:=1 od:
for st in cy do cyc[op(st)]:=`y` od;
cyc[op(ev)]:=`e`;
print( evalm(cyc));
t := infinity; r := 0,0; for i from 1 to nops(cy)/2 do
r1 := op(cy[2*i ]);
t1 := subs(M=infinity, x[r1]);
if t1 < t then t := t1; r := r1; fi
od ; lv := [r];
for i from 1 to nops(cy) do
x[op(cy[i])] := x[op(cy[i])]+(-1)^(i+1)*t od;
bv := {ev} union (bv minus {lv});
od; end:
```

Now let's look at the assignment problem from class.

```
c := matrix(3,3,[5,6,1,3,2,2,5,2,5 ]);
d := [1,1,1]:  s := [1,1,1]:
t := time():sol := fasttransport2(c,s,d,p);
ftime := time()-t;
```

We need to check this against the standard setup a transportation problem.

```
slowtransport := proc(c,s,d)
local m, n, x, eqs, i, j, obj,soln;
m := rowdim(c):  n := coldim(c):
x := matrix(m,n);
eqs := NULL;
for i from 1 to m do
eqs := eqs,sum(x[i,j],j=1..n)<= s[i],
sum(x[i,j],j=1..n)>= s[i] od;
i := 'i':
for j from 1 to n do
eqs := eqs,sum(x[i,j],i=1..m)<=d[j],
sum(x[i,j],i=1..m)>=d[j] od;
eqs := {eqs};
j := 'j':
obj := sum(sum(c[i,j]*x[i,j],j=1..n),i=1..m);
soln:=minimize(obj,eqs,NONNEGATIVE);
assign(soln);
[subs(soln,obj),evalm(x)];
end:
t := time():sln := slowtransport(subs(M=10^3
,op(c)),s,d);
time()-t;
```

Well, so much for the streamlinedness of the fast transportation algorithm. Slow is faster than fast in this implementation. We could try a larger problem to see what happens. Make a random 10 by 10 assignment problem.

```
c := map(abs,randmatrix(10,10));
s := [1$10];
```

```
d :=s;
t := time():sln := fasttransport2(subs(M=10^3
,op(c)),s,d);
time()-t;
t := time():sln := slowtransport(subs(M=10^3
,op(c)),s,d);
time()-t;
```

Exam topics:

1. Duality theory. Be able to write down the dual of a given lp problem. Know the statement of the duality theorem. Be able to discuss the pros and cons of working on the dual problem.

2. Sensitivity analysis and Parametric programming. Know how to use matrices to perform sensitivity analysis on the coefficients of the objective function and on the right hand sides of the constraints. Know how to use matrices to introduce new decision variables and new constraints into an lp problem. Know the dual simplex algorithm. Know what parametric programming is and how to use matrices to track the optimum as a function of theta (nuff said).

3. Transportation problems. Know how to set up a transportation problem as an lp problem. Know how to carry out the transportation simplex algorithm. Know assigment problems.

24 Ma 416 Exam 2

1. a) Write down the dual of the linear programming problem
maximize $z = 3\,x_1 + 2\,x_2$ subject to $x_1 + x_2 \le 3$, $2\,x_1 + x_2 \le 5$, $0 \le x_1$, $0 \le x_2$

b) In general, what does the duality theorem say about the optimum feasible solutions of the primal problem and its dual? In particular, what can you conclude about the optimum feasible solution of the dual in a).

2. When the maximization problem in #1 is solved with the simplex method, the initial and final tableaux look like this:

Initial: Final:

$$\begin{bmatrix} 1 & -3 & -2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 3 \\ 0 & 2 & 1 & 0 & 1 & 5 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 8 \\ 0 & 0 & 1 & 2 & -1 & 1 \\ 0 & 1 & 0 & -1 & 1 & 2 \end{bmatrix}$$

a) Explain the significance of the submatrix of the final tableau consisting of columns 1, 4 and 5.

b) Explain briefly how to use these tableau to carry out sensitivity analysis on on the problem. Illustrate by doing the sensitivity analysis of the coefficient of x_1 in the objective function.

3. Consider the linear programming problem with parameter θ:
maximize $z = 3\,x_1 + 2\,x_2$ subject to $x_1 + x_2 \le \theta$, $2\,x_1 + x_2 \le 5$, $0 \le x_1$, $0 \le x_2$
(Note that the problem has been solved when $\theta = 1$, in 2 above)
Let $z(\theta)$ be the optimum value of the problem. Make a sketch of the graph of $z(\theta)$.

4. A fleet of 15 ships is distributed in three ports: 5 ships in each of port A, B, and C. These ships need to be sent to pick up stuff at four different ports: 1 ship needed at port D, 4 at port E, 4 at port F, and 6 at port G. The table of travel costs from port X to port Y is given by $\begin{bmatrix} & D & E & F & G \\ A & 5 & 4 & 3 & 2 \\ B & 10 & 8 & 4 & 7 \\ C & 9 & 9 & 8 & 4 \end{bmatrix}$.
The problem is to decide on a shipping schedule which minimizes total cost.

a) Set this problem up as a linear programming problem: Give names to the decision variables and write down the objective function and constraints. (You don't have to write them all down, just say what they are and how many.)

b) Starting with the feasible solution you get from the northwest corner rule, $\begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}$, and the reduced cost table for the nonbasic variables,
$\begin{bmatrix} b & b & b & -4 \\ 4 & 3 & b & b \\ 6 & 7 & 7 & b \end{bmatrix}$, where 'b' stands for 'basic variable', perform one iteration of the transportation simplex algorithm. Is this a wasted iteration?

5. Take home:

a) Complete the solution of the transportation problem in 4)

b) Consider the transportation problem in 4) with travel costs

$\left[\begin{array}{ccccc} & D & E & F & G \\ A & 5 & 4 & 3 & 2 \\ B & \theta & 8 & 4 & 7 \\ C & 9 & 9 & 8 & 4 \end{array}\right]$. Let $z(\theta)$ denote the optimum travel cost. Sketch the graph of $z(\theta)$.

Network analysis using the networks package in Maple

```
restart;with(networks);
```

```
[acycpoly, addedge, addvertex, adjacency, allpairs, ancestor, arrivals, bicomponents,
    charpoly, chrompoly, complement, complete, components, connect, connectivity, contract,
    countcuts, counttrees, cube, cycle, cyclebase, daughter, degreeseq, delete, departures,
    diameter, dinic, djspantree, dodecahedron, draw, duplicate, edges, ends, eweight, flow,
    flowpoly, fundcyc, getlabel, girth, graph, graphical, gsimp, gunion, head, icosahedron,
    incidence, incident, indegree, induce, isplanar, maxdegree, mincut, mindegree, neighbors,
    new, octahedron, outdegree, path, petersen, random, rank, rankpoly, shortpathtree, show,
    shrink, span, spanpoly, spantree, tail, tetrahedron, tuttepoly, vdegree, vertices, void,
    vweight]
```

A network is first a graph, a set of vertices and edges. You can define graph in a Maple worksheet. First load in the networks package, which we have done. We need a problem to work on.

Suppose we have n cities and a table of distances between the cities.

Devise a road system of minimum total length which connects all the cities.

First create the graph. Initially it is empty.

```
new(dist):
```

Now we can add vertices (or nodes) to it.

```
addvertex([v1,v2,v3,v4],dist);
```

```
v1, v2, v3, v4
```

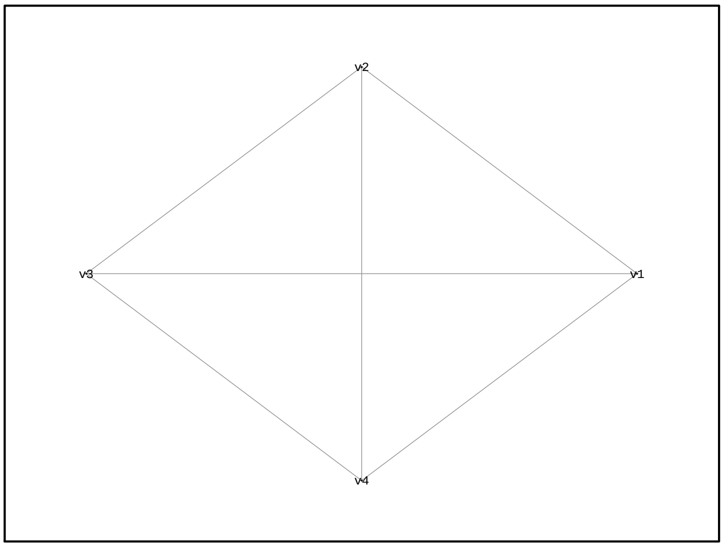
Then we can add edges.

```
addedge([{v1,v2},{v1,v3},{v1,v4},{v2,v
3},{v2,v4},{v3,v4}],dist);
```

```
e1, e2, e3, e4, e5, e6
```

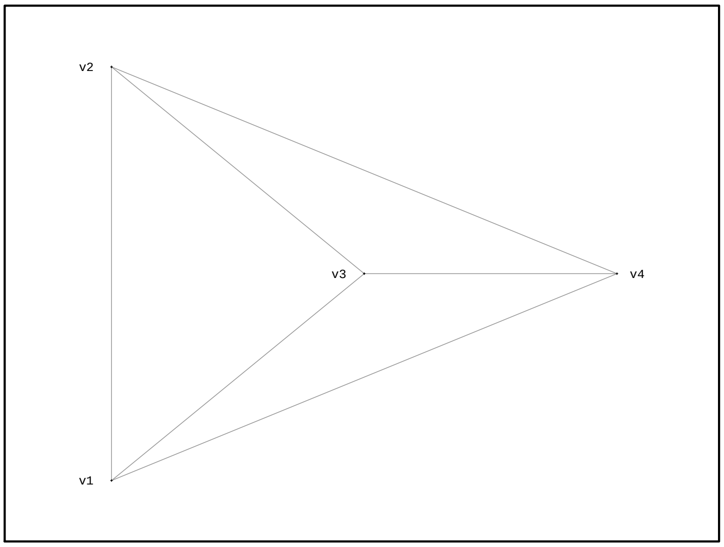
We can draw the graph constructed so far.

```
draw(dist);
```



We can draw it so that v1 and v2 are to the left of v3 is to the left of v4.

```
draw(Linear([v1,v2],[v3],[v4]),dist);
```



We can inquire what the edges and vertices of dist are.

```
edges(dist);
```

$$\{e1,\ e2,\ e3,\ e4,\ e5,\ e6\}$$

```
vertices(dist);
```

$$\{v1,\ v2,\ v3,\ v4\}$$

Initially, the vertices of the graph are not weighted, and the edges are each given weight 1.

```
vweight(dist);
```

$$\begin{array}{l}\mathrm{table}(\mathit{sparse},\ [\\ \quad])\end{array}$$

```
eweight(dist);
```

$$\begin{array}{l}\mathrm{table}([\\ \quad e2 = 1\\ \quad e4 = 1\\ \quad e6 = 1\\ \quad e3 = 1\\ \quad e1 = 1\\ \quad e5 = 1\\ \quad])\end{array}$$

We can assign random nonnegative integers less than 100 to be weights of the edges. That way we can construct random distance tables between citiies.

```
f := rand(100):
for e in edges(dist) do eweight(dist)[e]:=
f() od;
```

$$\mathrm{eweight}(\mathit{dist})_{e1} := 81$$
$$\mathrm{eweight}(\mathit{dist})_{e2} := 70$$
$$\mathrm{eweight}(\mathit{dist})_{e3} := 97$$
$$\mathrm{eweight}(\mathit{dist})_{e4} := 63$$
$$\mathrm{eweight}(\mathit{dist})_{e5} := 76$$
$$\mathrm{eweight}(\mathit{dist})_{e6} := 38$$

Checking that the edgeweights have indeed been altered –

```
eweight(dist);
```

```
table([
    e2 = 70
    e4 = 63
    e6 = 38
    e3 = 97
    e1 = 81
    e5 = 76
    ])
```

Now the word shortpathtree enables one to solve the problem of finding a shortest path from v1 to each of the vertices v2, v3, v4.

```
T :=shortpathtree(dist,v1):
edges(T);
```

```
{e1, e2, e3}
```

If we display the drawing of the graph dist and the shortest path tree T in it with the option insequence=true, we can toggle back and forth between the graph and the tree.

```
plots[display]([draw(T),draw(dist)],insequenc
e=true);
```

```
eweight(dist), eweight(T);
```

```
table([
    e2 = 70
    e4 = 63
    e6 = 38
    e3 = 97
    e1 = 81
    e5 = 76
    ]), table([
    e2 = 70
    e3 = 97
    e1 = 81
    ])
```

Note the shortpathtree is a spanning tree but not a minimum spanning tree, since the tree consisting of the edges e1, e4, and e5 is a minimum spanning tree.

Problem. Define a word to construct a minimum spanning tree.

A solution:

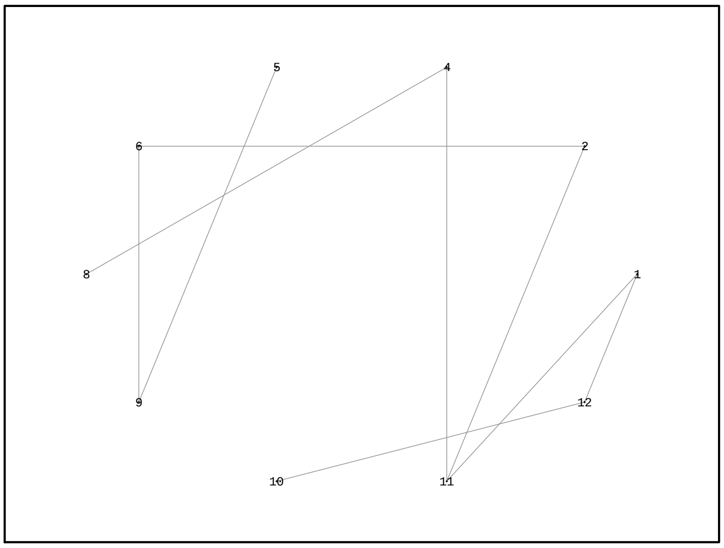
The algorithm is simple: choose a vertex, and start a tree mst with only that vertex in it. Find a shortest edge e connecting mst to a vertex v outside of mst. Add v to the vertices of mst and e to the edges. Continue this as long as you can, each time adding an edge of minimum length to the tree you have constructed so far. If you have exhausted all the vertices, fine. mst is a minimum spanning tree of the graph you started with. Otherwise, the graph you started with is not connected, and you have only constructed a minimum spanning tree of the component of the graph containing the vertex you started from.

How do you know it is a tree? Because at each step you glue on an edge exactly one of whose endpoints is in the tree mst. So there is no loop created. How do you know it is a minimum spanning tree? Because at each step you choose the shortest edge exactly one of whose ends is in the tree mst. So at each step mst is a minimum spanning tree for the set of vertices of mst.

```
minspantree := proc(dist,v1,‘mst‘)
local verts, edgs, v, tmp, e,e0;
new(mst);
verts := vertices(dist);
edgs := edges(dist):
addvertex(v1,mst);
verts := verts minus {v1};
while not verts ={} do
tmp := [infinity,e0];
for e in edgs do
if nops(ends(e,dist) intersect vertices(mst))=1
and eweight(e,dist) < tmp[1] then
tmp := [eweight(e,dist),e] fi;
od;
if tmp[1] < infinity then
v := op(ends(tmp[2],dist) minus vertices(mst));
e := tmp[2];
addvertex(v,mst);
verts := verts minus {v};
addedge(ends(e,dist),names=e,weights=eweight(dist)[e],
mst);
edgs := edgs minus {e} else ERROR(‘graph not connected‘) fi;
od;
NULL: end:
G := random(12,10):
minspantree(G,1,tr);
```

```
Error, (in minspantree) graph not connected
```

```
draw(tr);
```

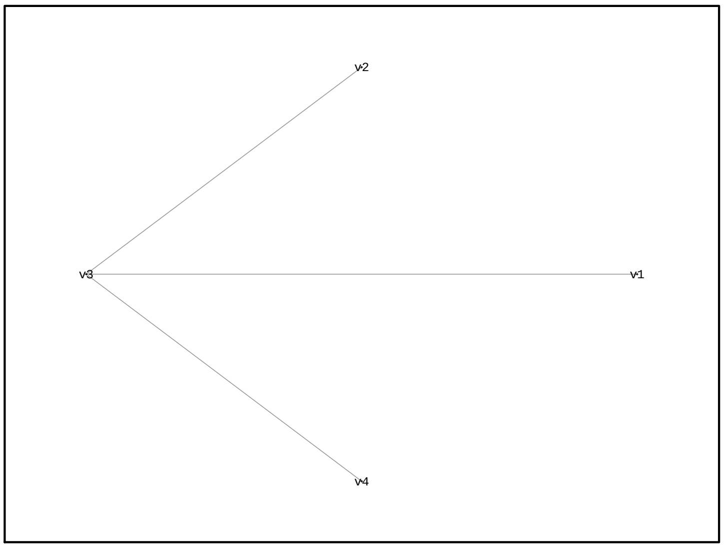


```
minspantree(dist,v1,mst):
eweight(mst);
```

```
table([
```

$e2 = 70$
$e4 = 63$
$e6 = 38$
])

```
draw(mst);
```



Let's check the sum of the edge weights in each tree T and mst.

```
convert({entries(eweight(mst))},‘+‘);
```

[171]

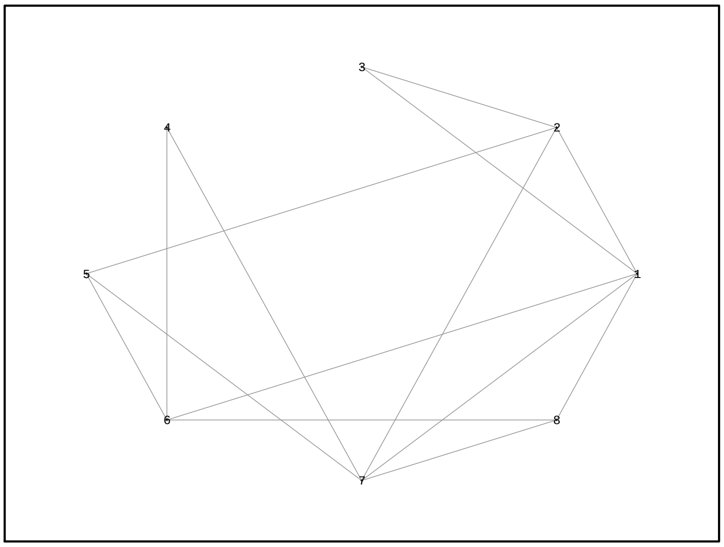
```
convert({entries(eweight(T))},‘+‘);
```

[248]

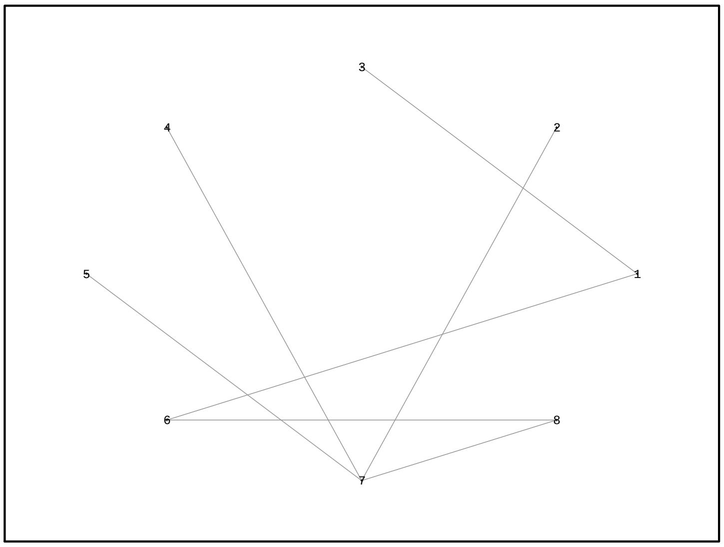
So we can see that the shortpath tree rooted at v1 is a long ways from the minimum spanning tree we grew from v1.

Exercise: Test this out on some random graphs. This is a good way to discover bugs in your code.

```
G := random(8,14):
draw(G);
```



```
minspantree(G,2,tr):
draw(tr);
```

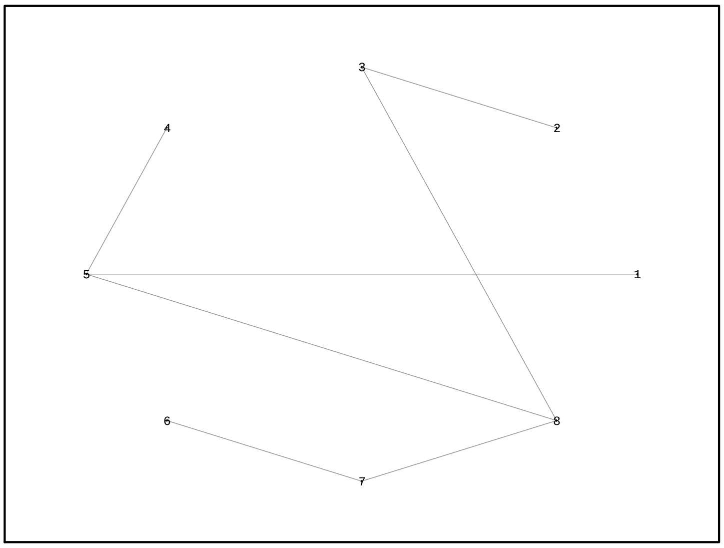


```
   new(G):
   addvertex([1,2,3,4,5,6,7,8],G):
   addedge([{2,1},{3,1},{4,1},{5,1},{6,
   1},{7,1},{8,1},
{3,2},{4,2},{5,2},{6,2},{7,2},{8,2},
{4,3},{5,3},{6,3},{7,3},{8,3},
{5,4},{6,4},{7,4},{8,4},
{6,5},{7,5},{8,5},
{7,6},{8,6},
{8,7}],G):
```

```
wts := [1.3,2.1,.9,.7,1.8,2.0,1.5,
.9,1.8,1.2,2.6,2.3,1.1,
2.6,1.7,2.5,1.9,1.,
.7,1.6,1.5,.9,
.9,1.1,.8,
.6,1.0,.5]:
edges(G);
```

```
{e7, e24, e25, e28, e8, e9, e10, e1, e2, e3, e4, e5, e6, e14, e15, e13, e18, e17, e16, e11,
   e12, e27, e19, e22, e21, e23, e20, e26}
```

```
for i from 1 to 28 do eweight(G)[e.i] :=
wts[i] od:
minspantree(G,1,mst):
draw(mst);
```



```
convert({entries(eweight(mst))},‘+‘);
```

```
                              [4.5]
```

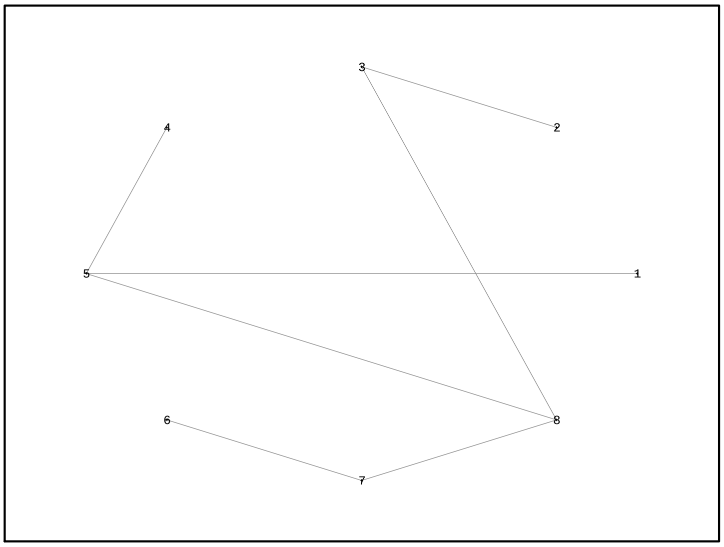
```
for i from 1 to 7 do for j from 8-i to 8 do
```

```
Warning, incomplete statement or missing semicolon
```

```
for e in edges(mst) do
print(e,eweight(mst)[e]) od;
```

```
                              e25, .8
                              e28, .5
                              e8, .9
                              e4, .7
                              e18, 1.
                              e19, .7
                              e26, .6
```

```
minspantree(G,8,mst):
draw(mst);
```



```
sort([op(edges(G))]);
```

```
[e1, e10, e11, e12, e13, e14, e15, e16, e17, e18, e19, e2, e20, e21, e22, e23, e24, e25, e26,
    e27, e28, e3, e4, e5, e6, e7, e8, e9]
```

```
for e in edges(G) do
print(ends(e,G),eweight(G)[e]) od:
```

```
{1, 8}, 1.5
{5, 7}, 1.1
{5, 8}, .8
{7, 8}, .5
{2, 3}, .9
{2, 4}, 1.8
{2, 5}, 1.2
{1, 2}, 1.3
{1, 3}, 2.1
{1, 4}, .9
{1, 5}, .7
{1, 6}, 1.8
{1, 7}, 2.0
{3, 4}, 2.6
{3, 5}, 1.7
{2, 8}, 1.1
{3, 8}, 1.
{3, 7}, 1.9
{3, 6}, 2.5
{2, 6}, 2.6
{2, 7}, 2.3
{6, 8}, 1.0
{4, 5}, .7
{4, 8}, .9
{4, 7}, 1.5
{5, 6}, .9
{4, 6}, 1.6
{6, 7}, .6
```

25 Minimum cost flow problems

Minimum cost flow problems can be set up as a linear programming problems, although the overhead gets out of hand rather quickly.

For example, the problem in our text which is used to explicate the network simplex algorithm is easy to check using simplex.

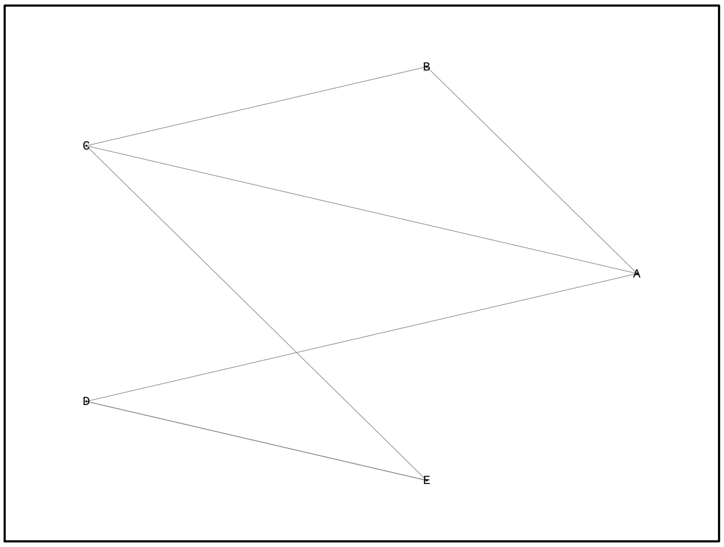
```
restart;with(linalg):with(networks):
```

```
Warning, new definition for norm

Warning, new definition for trace

Warning, new definition for charpoly

Warning, new definition for rank
```

```
new(G):
addvertex([A,B,C,D,E],weights=[40,50,0,-30,-6
0],G);
```

$$C,\, A,\, E,\, D,\, B$$

```
addedge([[A,C],[A,D],[A,B],[B,C],[C,E],[E,D],
[D,E]],names=[xAC,xAD,xAB,xBC,xCE,xED,xDE],
weights=[4,9, 2,3,1,2,3],G):
draw(G);
```



```
cost:=
sum(eweight(G)[edges(G)['i']]*edges(G)['i'],'i'=1..nops(edges(G)))
;
```

$$cost := 4\, xAC + 2\, xAB + 3\, xDE + 2\, xED + xCE + 3\, xBC + 9\, xAD$$

```
cost :=
4*xAC+3*xDE+2*xED+xCE+3*xBC+2*xAB+9*xAD;
```

$$cost := 4\, xAC + 2\, xAB + 3\, xDE + 2\, xED + xCE + 3\, xBC + 9\, xAD$$

```
cnsts := [xAB+xAC
+xAD=50,-xAB+xBC=40,xCE-xAC-xBC=0,
xCE+xDE-xED=60,xAD+xED-xDE=30,xCE<=80,xAB<=10];
```

$$\mathit{cnsts} := [\mathit{xAB} + \mathit{xAC} + \mathit{xAD} = 50,\ -\mathit{xAB} + \mathit{xBC} = 40,\ \mathit{xCE} - \mathit{xAC} - \mathit{xBC} = 0,\\ \mathit{xCE} + \mathit{xDE} - \mathit{xED} = 60,\ \mathit{xAD} + \mathit{xED} - \mathit{xDE} = 30,\ \mathit{xCE} \leq 80,\ \mathit{xAB} \leq 10]$$

```
soln :=
simplex[minimize](cost,cnsts,NONNEGATIVE);
```

$$\mathit{soln} := \{\mathit{xAB} = 0,\ \mathit{xDE} = 0,\ \mathit{xBC} = 40,\ \mathit{xED} = 20,\ \mathit{xAC} = 40,\ \mathit{xAD} = 10,\ \mathit{xCE} = 80\}$$

```
subs(soln,cost);
```

$$490$$

So simplex returns the same solution that the network simplex algorithm returns (see page 387).

Let's make up our own five node min cost flow problem. We will define it in terms of a tree, then use that to generate a cost function and set of constraints to feed to minimize in simplex.

```
new(G):
```

The graph is called G. It has 5 nodes called 1,2,3,4,5; 1 and 2 are source nodes of weights 10 and 20, 3 is a transhipment node (of weight 0) and 4 and 5 are destination nodes of weights 15 and 15.

```
addvertex([1,2,3,4,5],
weights=[10,20,0,-15,-15],G);
```

$$1,\ 2,\ 3,\ 4,\ 5$$

Now we'll add the roads (edges). There are 8 of them. The edge-weights are the costs.

```
addedge([[1,3],[1,4],[2,3],[2,5],[3,4],[3,5],
[4,5],[5,4]], names=[x[1,3],x[1,4],x[2,3],x[2,5],x[3,4],
x[3,5],x[4,5],x[5,4]],
weights=[2,1,1,2,2,3,2,1],G);
```

$$x_{1,3},\ x_{1,4},\ x_{2,3},\ x_{2,5},\ x_{3,4},\ x_{3,5},\ x_{4,5},\ x_{5,4}$$

The cost function is generated thusly:

```
cost:= sum(eweight
(G)[edges(G)[i]]*edges(G)[i],i=1..nops(edges(G)));
```

$$\mathit{cost} := 2\,x_{1,3} + x_{1,4} + x_{2,3} + 2\,x_{2,5} + 2\,x_{3,4} + 3\,x_{3,5} + 2\,x_{4,5} + x_{5,4}$$

The constraints are 5 in number: one for each node. In addition, we have decided there are finite capacities of 5 and 3 on the edges [1,4] and [4,5]. This gives two 'upper bound' constraints.

```
cnsts := [x[1,3]+x[1,4]=10,x[2,3]+x[2,5]=20,
x[1,3]+x[2,3]-x[3,4]-x[3,5]=0,
x[1,4]+x[3,4]+x[5,4]-x[4,5]=15,
x[2,5]+x[3,5]+x[4,5]-x[5,4]=15,x[1,4]<=5,
x[4,5]<=3];
```

$$\mathit{cnsts} := [x_{1,3} + x_{1,4} = 10,\ x_{2,3} + x_{2,5} = 20,\ x_{1,3} + x_{2,3} - x_{3,4} - x_{3,5} = 0,\\ x_{1,4} + x_{3,4} + x_{5,4} - x_{4,5} = 15,\ x_{2,5} + x_{3,5} + x_{4,5} - x_{5,4} = 15,\ x_{1,4} \leq 5,\ x_{4,5} \leq 3]$$

Now feed.

```
soln
:=simplex[minimize](cost,cnsts,NONNEGATIVE);
`min cost = `,subs(soln,cost);
```

$$\mathit{soln} := \{x_{3,5} = 0,\ x_{4,5} = 0,\ x_{5,4} = 0,\ x_{2,3} = 5,\ x_{2,5} = 15,\ x_{1,3} = 5,\ x_{3,4} = 10,\ x_{1,4} = 5\}$$

$$\mathit{min\ cost\ \ =},\ 70$$

Generating random minimum cost flow problems.

We want to define some words to work with minimum cost flow problems.

Here is a word to generate a random set of n positive integers whose sum is s.

```
randsupply := proc(s,n)
local i,j,ra,f,g,l;
f := rand(n);
l := vector([seq(1,i=1..n)]);
while sum(l[j],j=1..n)<s-n do
g := rand(s-sum(l[j],j=1..n));
ra := f()+1;
l[ra]:= l[ra]+g();
od;
while sum(l[j],j=1..n)<s do
ra := f()+1;
l[ra]:= l[ra]+1;
od;
convert(l,list) end:
randsupply(1000,4) ;
```

```
[171, 770, 8, 51]
```

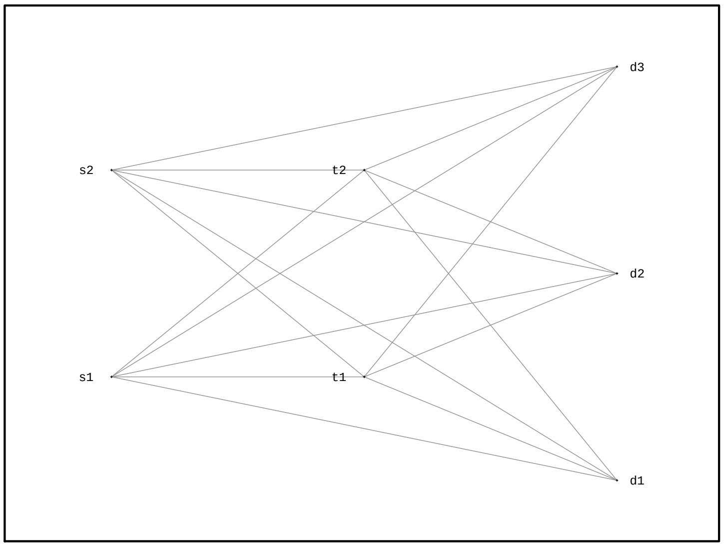
Here is a word to set up a minimum cost flow problem with m supply nodes with a total supply of su, r transhipment nodes, and n destination nodes. All capacities are infinite.

```
setupmcf := proc(m,su ,r,n,`G`)
local sup, des, g, sources, destin, trans, i, t, j,k,cost,cnsts, s,
d;
sup := randsupply(su,m);
des := randsupply(su,n);
g := rand(100);
sources := [seq(s.i,i=1..m)];
destin := [seq(d.i,i=1..n)];
trans := [seq(t.i,i=1..r)];
new(G);
for i from 1 to nops(sources) do
addvertex(sources[i],weights=sup[i],G) od;
for t in trans do
addvertex(t,weights=0 ,G) od;
for j from 1 to nops( destin) do
addvertex(destin[j],weights= des[j],G) od;
for s in sources do for d in destin do
addedge([s,d],names=x[s,d],weights=g(),G) od od;
for s in sources do for t in trans do
addedge([s,t],names=x[s,t],weights=g(),G) od od;
for t in trans do for d in destin do
addedge([t,d],names=x[t,d],weights=g(),G) od od;
print( draw(Linear([seq(s.i,i=1..m)],[seq(t.j,j=1..r)],
[seq(d.k,k=1..n)]),G)) ;
i := 'i':
cost := sum(eweight(G)[edges(G)[i]]*edges(G)[i],
i=1..nops(edges(G)));
cnsts := NULL: i := 'i':  j := 'j':  k :='k':
printlevel:=1;
```

```
for i from 1 to m do
cnsts := cnsts,
convert([seq(
op(edges([s.i,d.j],G)),j=1..n)] ,`+`)
+ convert([ seq(
op(edges([s.i,t.k],G)),k=1..r)] ,`+`) =
vweight(G)[s.i] od;
for i from 1 to r do
cnsts := cnsts,
- (convert([seq(
op(edges([t.i,d.j],G)),j=1..n)] ,`+`))
+ convert([ seq(
op(edges([s.k,t.i],G)),k=1..m)] ,`+`) =
0 od;
for i from 1 to n do
cnsts := cnsts,
convert([seq(
op(edges([s.j,d.i],G)),j=1..m)] ,`+`)
+ convert([ seq(
op(edges([t.k,d.i],G)),k=1..r)] ,`+`) =
vweight(G)[d.i] od;
[cost,[cnsts]];
end:
```

We can test this out.

```
prob:= setupmcf(2,100,2,3,G);
soln:=simplex[minimize](prob[1],prob[2],NONNEGATIVE);
subs(soln,prob[1]);
```



$$\begin{aligned}prob := [&65\, x_{s2,\, d2} + 35\, x_{t2,\, d3} + 38\, x_{s1,\, d1} + 49\, x_{t2,\, d2} + 60\, x_{s2,\, t2} + 82\, x_{t1,\, d1} + 92\, x_{t1,\, d2}\\ &+ 13\, x_{t1,\, d3} + 77\, x_{t2,\, d1} + 95\, x_{s1,\, t1} + 74\, x_{s1,\, t2} + 91\, x_{s2,\, d3} + 20\, x_{s1,\, d3} + 44\, x_{s2,\, d1} + 9\, x_{s2,\, t1}\\ &+ 13\, x_{s1,\, d2}, [x_{s1,\, d1} + x_{s1,\, d2} + x_{s1,\, d3} + x_{s1,\, t1} + x_{s1,\, t2} = 98,\\ &x_{s2,\, d1} + x_{s2,\, d2} + x_{s2,\, d3} + x_{s2,\, t1} + x_{s2,\, t2} = 2,\ -x_{t1,\, d1} - x_{t1,\, d2} - x_{t1,\, d3} + x_{s1,\, t1} + x_{s2,\, t1} = 0,\\ &-x_{t2,\, d1} - x_{t2,\, d2} - x_{t2,\, d3} + x_{s1,\, t2} + x_{s2,\, t2} = 0,\ x_{s1,\, d1} + x_{s2,\, d1} + x_{t1,\, d1} + x_{t2,\, d1} = 9,\\ &x_{s1,\, d2} + x_{s2,\, d2} + x_{t1,\, d2} + x_{t2,\, d2} = 77,\ x_{s1,\, d3} + x_{s2,\, d3} + x_{t1,\, d3} + x_{t2,\, d3} = 14]]\end{aligned}$$

$$soln := \{x_{t2,\,d2} = 0,\ x_{s1,\,t1} = 0,\ x_{t1,\,d3} = 2,\ x_{s1,\,d2} = 77,\ x_{s1,\,d3} = 12,\ x_{s2,\,t1} = 2,\ x_{s2,\,t2} = 0,\ x_{t1,\,d2} = 0,\ x_{s1,\,t2} = 0,\ x_{s2,\,d3} = 0,\ x_{s2,\,d1} = 0,\ x_{s1,\,d1} = 9,\ x_{t2,\,d1} = 0,\ x_{s2,\,d2} = 0,\ x_{t2,\,d3} = 0,\ x_{t1,\,d1} = 0\}$$

$$1627$$

26 Ma 416 Exam 2

1. a) Write down the dual of the linear programming problem

maximize $z = 3\,x_1 + 2\,x_2$ subject to $x_1 + x_2 \le 3$, $2\,x_1 + x_2 \le 5$, $0 \le x_1$, $0 \le x_2$

minimize $3\,y1 + 5\,y2$

subject to $3 \le y1 + 2\,y2$, $5 \le y1 + y2$, $0 \le y1$, $0 \le y2$

b) In general, what does the duality theorem say about the optimum feasible solutions of the primal problem and its dual? In particular, what can you conclude about the optimum feasible solution of the dual in a) using the solution

given in 2 below.

The duality theorem says that if the primal problem has an optimum feasible solution, then so does the dual problem and in fact, the shadow costs of the

constraints in the final tableau give an optimum feasible solution to the dual

problem, and the optimum value of the objectives are equal. The duality

theorem also says that if the primal problem has an unbounded objective,

then the dual has no feasible solutions.

In particular, looking at the final tableau of the primal problem given below

we see that the solution to the dual problem is y1=1, y2=1, with a minimum

objective value of by* = 8.

2. When the maximization problem in #1 is solved with the simplex method, the initial and final tableaux look like this:

Initial: Final:

$$\begin{bmatrix} 1 & -3 & -2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 3 \\ 0 & 2 & 1 & 0 & 1 & 5 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 8 \\ 0 & 0 & 1 & 2 & -1 & 1 \\ 0 & 1 & 0 & -1 & 1 & 2 \end{bmatrix}$$

a) Explain the significance of the submatrix of the final tableau consisting of columns 1, 4 and 5.

This matrix contain a complete record of the iterations of the simplex algorithm performed on the initial tableau, so that if we multiply this matrix times the initial tableau, we get the final tableau. This makes it easier to carry out sensitivity analysis on the parameters of the problem and also do parametric programming.

b) Explain briefly how to use these tableau to carry out sensitivity analysis on on the problem. Illustrate by doing the sensitivity analysis of the coefficient of x_1 in the objective function.

```
init:=matrix([[1, -3, -2, 0, 0, 0], [0, 1, 1,
1, 0, 3], [0, 2, 1, 0, 1, 5]]);
```

$$init := \begin{bmatrix} 1 & -3 & -2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 3 \\ 0 & 2 & 1 & 0 & 1 & 5 \end{bmatrix}$$

```
fin:=matrix([[1, 0, 0, 1, 1, 8], [0, 0, 1, 2,
-1, 1], [0, 1, 0, -1, 1, 2]]);
```

$$fin := \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 8 \\ 0 & 0 & 1 & 2 & -1 & 1 \\ 0 & 1 & 0 & -1 & 1 & 2 \end{bmatrix}$$

```
with(linalg):
```

```
Warning, new definition for norm

Warning, new definition for trace
```

```
ga :=submatrix(fin,[1,2,3],[1,4,5]);
```

$$ga := \begin{bmatrix} 1 & 1 & 1 \\ 0 & 2 & -1 \\ 0 & -1 & 1 \end{bmatrix}$$

```
init[1,2]:=-(3+theta):init[1,3]:=-2:evalm(ini
t);
```

$$\begin{bmatrix} 1 & -3-\theta & -2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 3 \\ 0 & 2 & 1 & 0 & 1 & 5 \end{bmatrix}$$

```
evalm(ga&*init);
```

$$\begin{bmatrix} 1 & -\theta & 0 & 1 & 1 & 8 \\ 0 & 0 & 1 & 2 & -1 & 1 \\ 0 & 1 & 0 & -1 & 1 & 2 \end{bmatrix}$$

```
m:=evalm(matrix(3,3,[1,0,theta,0,1,0,0,0,1]))
;
```

$$m := \begin{bmatrix} 1 & 0 & \theta \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

```
evalm(m&*ga&*init);
```

$$\begin{bmatrix} 1 & 0 & 0 & 1-\theta & 1+\theta & 8+2\,\theta \\ 0 & 0 & 1 & 2 & -1 & 1 \\ 0 & 1 & 0 & -1 & 1 & 2 \end{bmatrix}$$

By inspection, θ can range from -1 to 1 before the optimum corner changes, so the coefficient of x_1 ranges from 2 to 4.

3. Consider the linear programming problem with parameter θ:
maximize $z = 3\,x_1 + 2\,x_2$ subject to $x_1 + x_2 \leq \theta$, $2\,x_1 + x_2 \leq 5$, $0 \leq x_1$, $0 \leq x_2$
(Note that the problem has been solved when $\theta = 3$, in 2 above)
Let z(θ) be the optimum value of the problem. Make a sketch of the graph of z(θ).

```
init[1,2]:=-3:init[2,6]:=theta:evalm(init);
```

$$\begin{bmatrix} 1 & -3 & -2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & \theta \\ 0 & 2 & 1 & 0 & 1 & 5 \end{bmatrix}$$

```
l :=evalm(ga&*init);
```

$$l := \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 5+\theta \\ 0 & 0 & 1 & 2 & -1 & -5+2\,\theta \\ 0 & 1 & 0 & -1 & 1 & 5-\theta \end{bmatrix}$$

By inspection, we see that $z(\boldsymbol{\theta}) = \mathbf{5} + \boldsymbol{\theta}$, for theta between 5/2 and 5.
At 5/2, we bring the 5th column into solution and take out the 3rd.

```
m := gauss(col(1,5),2);
```

$$m := \begin{bmatrix} 1 & 1 & 0 \\ 0 & -1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

```
evalm(m&*l);
```

$$\begin{bmatrix} 1 & 0 & 1 & 3 & 0 & 3\,\theta \\ 0 & 0 & -1 & -2 & 1 & 5-2\,\theta \\ 0 & 1 & 1 & 1 & 0 & \theta \end{bmatrix}$$

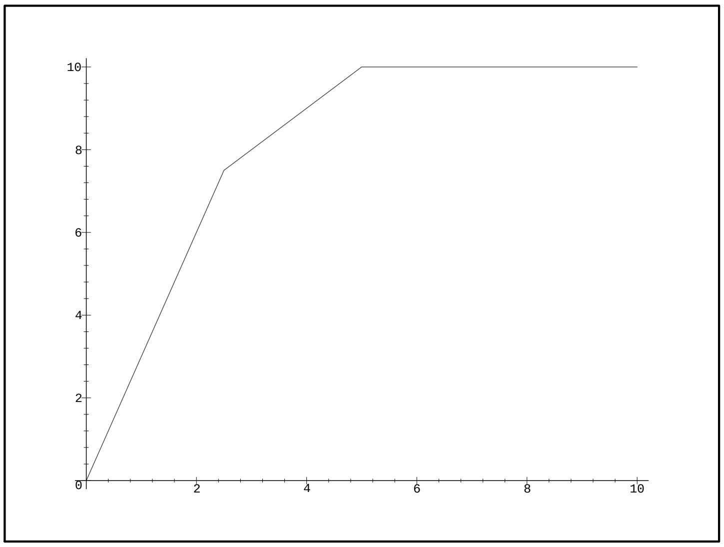
From 5/2 down to 0, $z(\boldsymbol{\theta}) = \mathbf{3}\,\boldsymbol{\theta}$.
Going up from 5, we bring the 4th column into solution and take out the 2nd.

```
evalm(gauss(col(1,4),3)&*l);
```

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 2 & 10 \\ 0 & 2 & 1 & 0 & 1 & 5 \\ 0 & -1 & 0 & 1 & -1 & -5+\theta \end{bmatrix}$$

So , for theta from 5 on up, $z(\boldsymbol{\theta}) = \mathbf{10}$ is constant.

```
plot([[0,0],[5/2,15/2],[5,10],[10,10]]);
```



4. A fleet of 15 ships is distributed in three ports: 5 ships in each of port A, B, and C. These ships need to be sent to pick up stuff at four different ports: 1 ship needed at port D, 4 at port E, 4 at port F, and 6 at port G. The table of travel costs from port X to port Y is given by

$$\left[\begin{array}{ccccc} & \boldsymbol{D} & \boldsymbol{E} & \boldsymbol{F} & \boldsymbol{G} \\ \boldsymbol{A} & \mathbf{5} & \mathbf{4} & \mathbf{3} & \mathbf{2} \\ \boldsymbol{B} & \mathbf{10} & \mathbf{8} & \mathbf{4} & \mathbf{7} \\ \boldsymbol{C} & \mathbf{9} & \mathbf{9} & \mathbf{8} & \mathbf{4} \end{array}\right].$$

The problem is to decide on a shipping schedule which minimizes total cost.

a) Set this problem up as a linear programming problem: Give names to the decision variables and write down the objective function and constraints. (You don't have to write them all down, just say what they are and how many.)

Let x_{XY} denote the number of ships we decide to move from X to Y.
Then we want to mimimize the cost
$5\,x_{AD} + 4\,x_{AE} + \ldots + x_{CF} + x_{CG}$ subject to the three supply constraints
$x_{AB} + x_{AE} + x_{AF} + x_{AG} = 5$, .., and 4 demand constraints $x_{AD} + x_{BD} + x_{CD} = 1$, ...

b) Starting with the feasible solution you get from the northwest corner rule, $\begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}$, and the reduced cost table for the nonbasic variables,
$\begin{bmatrix} b & b & b & -4 \\ 4 & 3 & b & b \\ 6 & 7 & 7 & b \end{bmatrix}$, where 'b' stands for 'basic variable', perform one iteration of the transportation simplex algorithm. Is this a wasted iteration?

The reduced cost table tells us to bring the upper right hand corner into solution.
Doing this creates a 2 by 2 cycle in the solution which we break by taking out the top of the 3rd column. This is a wasted iteration, since no allocation is changed.

5. Take home:

a) Complete the solution of the transportation problem in 4)

```
cost:=matrix( [[ 5, 4, 3, 2], [ 10, 8, 4,
7], [ 9, 9, 8, 4]]);
```

$$cost := \begin{bmatrix} 5 & 4 & 3 & 2 \\ 10 & 8 & 4 & 7 \\ 9 & 9 & 8 & 4 \end{bmatrix}$$

```
fasttransport2(cost,[5,5,5],[1,4,4,6],prt);
```

$$\begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

$$reduced\ costs$$

$$\begin{bmatrix} b & b & b & -4 \\ 4 & 3 & b & b \\ 6 & 7 & 7 & b \end{bmatrix}$$

$$iteration\ ,\ 1$$

$$\begin{bmatrix} 1 & 1 & y & e \\ 0 & 0 & y & y \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 4 & 0 & 0 \\ 0 & 0 & 4 & 1 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

$$reduced\ costs$$

$$\begin{bmatrix} b & b & 4 & b \\ 0 & -1 & b & b \\ 2 & 3 & 7 & b \end{bmatrix}$$

$$iteration\ ,\ 2$$

$$\begin{bmatrix} 1 & y & 0 & y \\ 0 & e & 1 & y \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}$$

$$\mathit{reduced\ costs}$$

$$\begin{bmatrix} b & b & 3 & b \\ 1 & b & b & 1 \\ 2 & 3 & 6 & b \end{bmatrix}$$

The minimal cost is 63 with the shipping schedule send 1 ship from A to D 3 from A to E and 1 from A to G, 1 from B to E and 4 from B to F, and 5 from C to G.

b) Consider the transportation problem in 4) with travel costs

$$\left[\begin{array}{ccccc} & D & E & F & G \\ A & 5 & 4 & 3 & 2 \\ B & \theta & 8 & 4 & 7 \\ C & 9 & 9 & 8 & 4 \end{array}\right]$$

. Let $z(\theta)$ denote the optimum travel cost. Sketch the graph of $z(\theta)$.

```
cost[2,1]:=theta:
seq(slowtransport(subs(theta=i,op(cost)),[5,5
,5],[1,4,4,6]),i=0..10);
```

$$[54, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [55, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [56, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [57, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}],$$

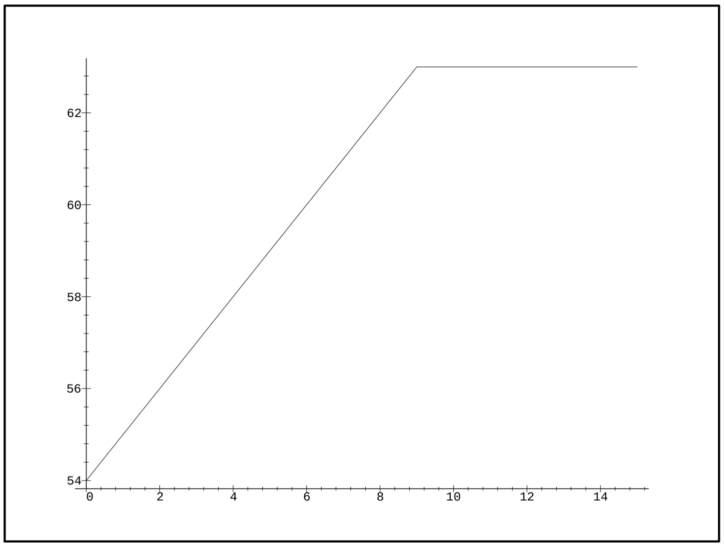
$$[58, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [59, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [60, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [61, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}],$$

$$[62, \begin{bmatrix} 0 & 4 & 0 & 1 \\ 1 & 0 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [63, \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}], [63, \begin{bmatrix} 1 & 3 & 0 & 1 \\ 0 & 1 & 4 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}]$$

By inspection, we see that $z(\theta) = 54 + \theta$, *as theta goes from 0 to 9.*

From there on $z(\theta) = 63$, since that shipment will never come back into solution.

```
plot([[0,54],[9,63],[15,63]]);
```


62
60
58
56
54
0
2
4
6
8
10
12
14

27 Tiling problems: solved with netsworks

The general tiling problem could be stated as follows: Given a bounded region R and a finite set of tiles, is it possible to fill the region R exactly with a set of nonoverlapping translates of the tiles? The region and the tiles are made of small squares of the same size. For relatively small regions R and a small number of tiles, we can investigate this question with Maple.

Since the region R is bounded we can position it in the first quadrant. The tiles and the region will be represented as polynomials in x and y with coefficients 0 or 1, with each monomial representing a quadrille. So we can make use of polynomial addition mod 2 to calculate overlap of tiles, that is, if t is a tile and R is a region, then t lies in R if (R+t) mod 2 has nops(t) fewer terms than R.

We can define a word to generate a rectangular region. Note: These words were developed jointly by myself and Paul Eakin.

```
rect := proc(n,m)
local i,j;
expand(
sum(x^'i','i' = 0 ..  m-1)
* sum(y^'j','j' = 0 ..  n-1))
end:
```

Here is a word that generates a triangular region.

```
tri := proc(n)
local i,j;
simplify( sum(x^i *sum(y^j,j=0..  n-i ), i
=0..n)) end:
```

The word degs returns a list of ordered pairs of integers which give the x-degree and y-degree of each monomial in f. These pairs are arranged in lexicographic order, that is, they are arranged first by their x-degree and within that by their y-degree.

```
degs := proc(f)
local tmp,tf,i,j;
tmp :=NULL;
tf := collect(simplify(f),[x,y]);
for i from 0 to degree(tf,x) do
for j from 0 to degree(tf,y) do
if 0 < coeff(coeff(tf,x,i),y,j)
then
tmp := tmp , [i,j]
fi
od
od;
[tmp]
end:
```

Squares is a word which we will use to draw a picture of the tiling. It works the same as degs except it returns a list of square polygons centered on integer points.

```
squares := proc(f)
local tmp,tf,i,j;
```

```
tmp :=NULL;
tf := collect(simplify(f),[x,y]);
for i from 0 to degree(tf,x) do
for j from 0 to degree(tf,y) do
if 0 < coeff(coeff(tf,x,i),y,j)
then
tmp := tmp,
[[i+.1,j+.1],[i+1-.1,j+.1],[i+1-.1,j+1-.1],[i+.1,j+1-.1]];
fi
od
od;
[tmp]
end:
```

The word nextone finds the "next" monomial (in the lexicographic order) in untiled which hasn't been covered yet. By the lexicographic order on the monomials , we mean $x^i\, y^j < x^k\, y^l$ if and only if $i < k$ or $i = k$ and $j < l$, that is, thinking of monomials as points [i,j] in the first quadrant, one is 'less' than the other if it lies to the left of the other or is in the same column as but below the other.

```
nextone := proc(untiled)
local n,x,y;
n := degs(untiled);
if not n = [] then
x^n[1][1]*y^n[1][2] else 0 fi
end:
```

Terms is defined to get around a small problem with using nops to count the terms in a polynomial. It works fine as long as the polynomial has more than one term. If you compute nops(x^2), or nops(x*y) however, you get 2 for an answer, because x^2 is of type ‘^‘ and so has 2 ops. Also nops(0) turns out to be 1, but the 0 polynomial has 0 terms.

```
terms := proc(f)
if type(f,`+`) = true then nops(f)
elif type(f,`*`) = true then 1
else 0
fi
end:
with(networks):
```

The word tiletree takes a list of tiles, a region R, an upper bound m (on the length of a tiling sequence), and proposed names for the graph of tilings, table Plots of plots, table P of tiled and untiled portions of R, and table Wt of weights or distances from the start

of a given tiling.

The algorithm builds a tree of tilings starting from the lower left hand corner of the region R and placing tiles lexicographically. It is unsuitable for even medium large regions.

```
tiletree :=
proc(tiles,R,m,`G`,`Plots`,`P`,`Wt`)
local c,k,w,v,i,d,tile,tiling,untiled;
new(G);
```

```
addvertex(v0,G);
Plots := table([]);
Plots[v0] := plot({degs(R)},style =
point);
P := table([]);
Wt := table([]);
c :=
[red,blue,yellow,purple,orange,green,black];
P[v0] := [0,R];
Wt[0] := v0;
k := 0;
for w from 0 to m do
if Wt[w] = NULL then break
else Wt[w+1] := NULL
fi;
for v in Wt[w] do
for i to nops(tiles) do
if not P[v][2] = 0 then
d := degs(P[v][2]);
tile :=
expand(x^d[1][1]*y^d[1][2]*tiles[i]);
tiling := simplify(P[v][1]+tile) mod 2;
untiled := simplify(P[v][2]+tile) mod 2;
if terms(tiling) =
terms(tile)+terms(P[v][1]) and
terms(P[v][2]) =
terms(tile)+terms(untiled) then
k := k+1;
addvertex(v.k,G);
P[v.k] :=
[tiling,untiled];
Plots[v.k] :=
Plots[v],
plots[polygonplot](
[squares(tile)],color = c[i]);
addedge([v,v.k],G);
Wt[w+1] :=
Wt[w+1],v.k
fi
fi
od
od
od;
op(Wt)
end:
```

Test tiletree out.

```
tiletree( [rect(2,1),rect(1,3)],rect(5,7)-x^3*y^2-y^4-x^6*y,20,g1,pl1,p1,wt1);
vertices(g1);
draw(Linear(seq([wt1[i]],i=0..11)),g1);
termpos := proc(h)
local tmp,i;
tmp := NULL;
for i to nops(vertices(h))-1 do
if nops(departures(v.i,h)) = 0 then
tmp := tmp,v.i
fi
od;
tmp
end:
termpos(g);
```

The word showterm shows the tilings that can't be extended to a tiling that hasn't already been defined. This includes those tilings which can't be extended at all.

```
showterm := proc(g,pl,exclude)
local v,ex;
ex := plots[polygonplot](squares(exclude),color=grey);
plots[display](
[seq(plots[display]([ex,pl[v]]),v
= [termpos(g)])],
insequence = true)
end:
showterm(g1 ,pl1 ,x^3*y^2+y^4+x^6*y);
```

Godeep simply looks for a tiling, without trying to build all of them at once. It works better than tiletree on larger tiling problems.

```
godeep :=
proc(tiles,R,m,‘G‘,‘Plots‘,‘P‘,‘Wt‘)
local c, k, w, v, i, d, tile, tiling,
untiled, Left, u, cand;
new(G);
addvertex(v0,G);
Plots := table([]);
Plots[v0] := plot({degs(R)},style =
point);
P := table([]);
Wt := table([]);
Left := table([]);
c :=
[red,blue,yellow,brown,orange,green,black];
P[v0] := [0,R];
Wt[0] := v0;
for k from 1 to m do Wt[k] := NULL od:
```

```
   Left[v0] := {op(tiles)};
   k := 0;
   w := 0:
   u := v0:
   while not Left[u] = {} and k < m
   do
   cand := Left[u][1]:
   print(cand);
   Left[u]:= Left[u] minus
{Left[u][1]};
   d := degs(P[u][2]);
   tile :=
   expand(x^d[1][1]*y^d[1][2]*cand)
   ;
   tiling :=
   simplify(P[u][1]+tile) mod 2;
   untiled :=
   simplify(P[u][2]+tile) mod 2;
   if terms(tiling) =
   terms(tile)+terms(P[u][1]) and
   terms(P[u][2]) =
   terms(tile)+terms(untiled) then
   k := k+1;
   addvertex(v.k,G);
   Left[v.k] :=
{op(tiles)};
   P[v.k] :=
   [tiling,untiled];
   Plots[v.k] :=
   Plots[u],
   plots[polygonplot](
   [squares(tile)],color = c[1+( k mod 7)]);
   addedge([u,v.k],G);
   w := w+1:
   Wt[w ] := Wt[w
   ],v.k;
   u := v.k:
   if P[u][2] = 0 then
   RETURN(‘finished¡) fi;
   fi;
   print(u,Left[u]);
   while Left[u] = {} do
   if u = v0 then RETURN(‘no
   tiling‘) fi;
   u := arrivals(u,G)[1];
```

```
print(‘going down‘,w-1);
w := w-1;
od;
od ;

op(Wt)
end:

godeep( [rect(2,1),rect(1,3)] ,
rect(7,7)-x^3*y^2-y^4-x^6*y,
50,g,pl,p,wt);
indices(pl);
plots[display](pl[v34]);

spt := shortpathtree( g,v0):
entries(wt);
pt :=path([v34,v0],spt);

plots[display]([seq(plots[display](pl[p]),p=p
t)],insequence=true);
```

28 Tiling with the simplex algorithm

Suppose we have two tiles, f and g, and wish to tile a region F. This is equivalent to finding two polynomials A and B such that $F = Af + Bg$ where the coefficients of A and B are all 1.

```
restart;
rect := proc(n,m)
local i,j;
expand(
sum(x^'i','i' = 0 ..  m-1)
* sum(y^'j','j' = 0 ..  n-1))
end:
```

Now we need some tiles

```
tile1:=1+x+x^2+y:
tile2:=y+x*y+x^2*y+x^2 :
```

Now choose a polynomial region f.

```
f:= rect(4,4);
```

We need a word to convert f to a linear form.

```
cnv := proc(f)
local tmp,tf,i,j;
tmp :=NULL;
tf := collect(simplify(f),[x,y]);
for i from 0 to degree(tf,x) do
for j from 0 to degree(tf,y) do
if not coeff(coeff(tf,x,i),y,j)=0
then
tmp :=
tmp,coeff(coeff(tf,x,i),y,j)*z[i,j];
fi
od
od;
convert([tmp],`+`);
end:
cnv(f);
```

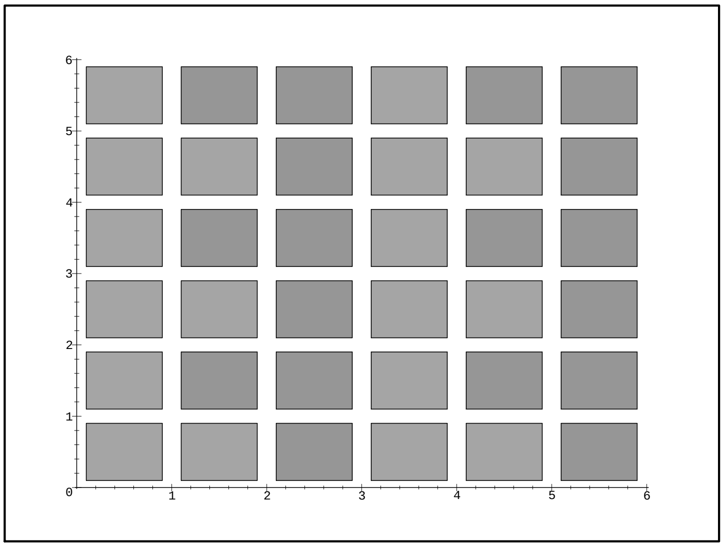
This word will be used to draw the tiling.

```
squares := proc(f)
local tmp,tf,i,j;
tmp :=NULL;
tf := collect(simplify(f),[x,y]);
for i from 0 to degree(tf,x) do
for j from 0 to degree(tf,y) do
if 0 < coeff(coeff(tf,x,i),y,j)
then
tmp := tmp,
[[i+.1,j+.1],[i+1-.1,j+.1],[i+1-.1,j+1-.1],[i+.1,j+1-.1]];
fi
```

```
      od
      od;
      [tmp]end:
```

Here is the word to search for a solution using the simplex algorithm. The idea is simple:

```
      tileit := proc(f,tile1,tile2)
      local A, B,i,a,b, gf, p,eqns, sol, sln1, sln2, drawing ;
      A := convert([seq(a[i-1]*op(i,f),i=1..nops(f))],‘+‘);
      B := convert([seq(b[i-1]*op(i,f),i=1..nops(f))],‘+‘);
      gf := cnv(expand(A*tile1+tile2*B -f))
      ;
      eqns:=seq(op(1,op(i,gf))=0 ,i=1..nops(gf));
      sol:=simplex[maximize](1,[eqns],NONNEGATIVE);
   if not sol={} then sln1 := expand(subs(sol,A)) sln2 := expand(subs(sol,B)) drawing
:= NULL if not sln1 = 0 then
      drawing :=
      drawing,seq(plots[polygonplot](squares(expand(
      tile1*p)),color=tan) ,p=[op(sln1 )]) fi;
      if not sln2 = 0 then
      drawing:= drawing,
      seq(plots[polygonplot](squares(expand(
      tile2*p)),color=green) ,p=[op(sln2)]) fi;
      print(plots[display]([drawing]));
      else RETURN(‘no tiling‘) fi;
      [sln1,sln2];
      end:
      f :=rect(7,7)-x^3*y^2-y^4-x^6*y;
      tile1 := 1+x+x^2:  tile2 := 1+y+y^2:
      tileit(rect(6,6),1+x+y,y+x*y+x);
```



$$[y^4 + x^3 + x^3\,y^2 + 1 + x^3\,y^4 + y^2,\; x\,y^4 + x^4 + x + x^4\,y^2 + x^4\,y^4 + x\,y^2]$$

```
tileit(rect(8,8)-rect(4,4)+rect(2,3)+y^2+x*y^2,rect(1,2) , rect(1,3)
);
tileit(rect(6,6)-x*y-x^3*y^3-x^5*y^5,els(2)[1
],1+x);
undebug(cnv);
els(3);
```

29 Soma Cube solutions using the simplex algorithm.

The soma cube puzzle is well known to have lots of different solutions. Can we use the simplex algorithm to find one. Yes!

```
reduce := proc(tile,‘w‘)
expand(quo(tile,w^ldegree(tile,w),w)) ;
end:

redall := proc(tile)
reduce(reduce(reduce(tile,x),y),z) end:

reflect := proc(tile, w )
local tmp, reftile, mon, n, refmon, refile;
tmp := reduce(tile,w);
reftile := 0;
for mon in tmp do
n := degree(mon,w);
refmon := mon*w^(2-2*n);
reftile := reftile+refmon;
od; reftile end:

rot := proc(tile,‘s‘,‘t‘)
local tmp;
tmp := reduce(tile,s);
tmp := reduce(tmp,t);
tmp := subs({s=S,t=T},tmp);
tmp := subs({S=t,T=s},tmp);
tmp; end:

rrot := proc(tile)
local tmp;
tmp := reduce(tile,x);
tmp := reduce(tmp,y);
tmp := reduce(tmp,z);
tmp := subs({x=S,y=T,z=W},tmp);
tmp := subs({S=z,T=x,W=y},tmp);
tmp; end:

getorbit := proc(intile)
local tile, ntile;
tile := intile;
ntile :=map(redall,tile union map(rrot,tile) union map(rot,tile,x,y)
union map(reflect,tile,x) union map(rot,tile,y,z)
union map(reflect,tile,z) union map(rot,tile,x,z))
;
while nops(ntile)>nops(tile) do
tile := ntile;
ntile :=map(redall,tile union map(rrot,tile) union map(rot,tile,x,y)
union map(reflect,tile,x) union map(rot,tile,y,z)
union map(reflect,tile,z) union map(rot,tile,x,z))
;
od;
tile end:

getorbit(tee);
```

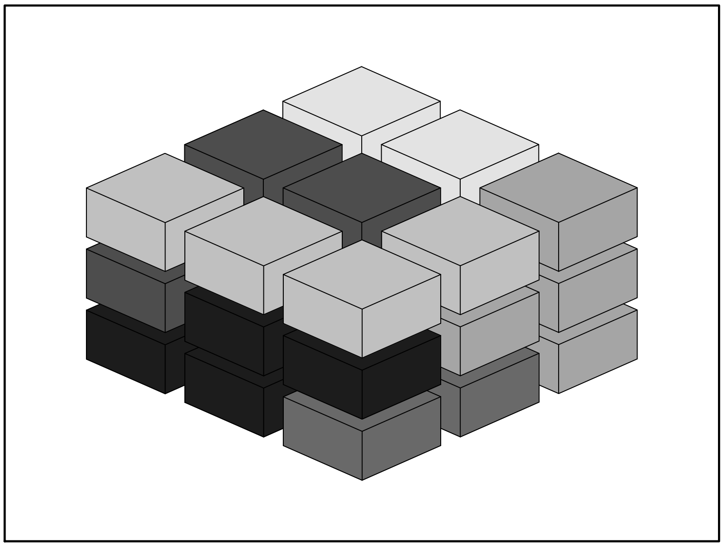
$$\{1+y+y^2+y\,z,\ x\,z^2+x\,z+x+z,\ 1+z+z^2+x\,z,\ y\,z^2+y\,z+y+z,\ 1+z+z^2+y\,z,\ 1+y+y^2+x\,y,\ 1+x+x^2+x\,z,\ x^2\,z+x\,z+z+x,\ x+x\,y+x\,y^2+y,\ x^2\,y+x\,y+y+x,\ 1+x+x^2+x\,y,\ y^2\,z+y\,z+z+y\}$$

$$\mathbf{24}$$

```
sixes := [seq( getorbit( {op(tiles[i])})
,i=1..6)];
map(nops,sixes);
```

$$[12,\ 24,\ 12,\ 12,\ 24,\ 24]$$

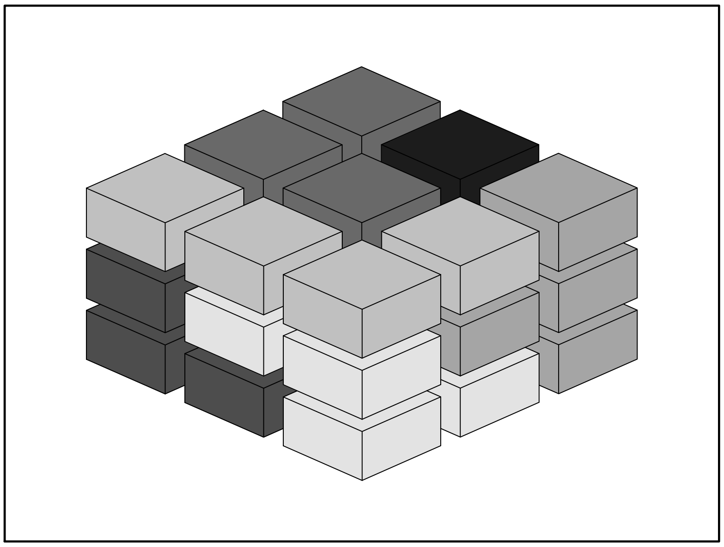
```
sol1 := [1+x+y+z, y^2+y^2*z+y^2*z^2+y^2*x*z,
x^2*z^2+x*y^2*z^2+x^2*y*z^2+x^2*y^2*z^2, x*y+x*y^2+x^2*y^2,
x^2*y+x^2*y*z+x^2*y^2*z+x^2, x*z+x^2*z+x*z^2+x*y*z^2,
y*z+y*z^2+x*y*z+z^2]:
i:='1':plots[display]([seq(drawtile(sol1[i],c
lrs[i]),i=1..7)] );
```



```
sol2 := [1+x+y+z,y^2+y^2*z+y^2*z^2+y^2*x*z,
x^2*z^2+x*y^2*z^2+x^2*y*z^2+x^2*y^2*z^2, x*z^2+x*y*z^2+z^2,
x*y+y*z+y*z^2+x*y*z, x*z+x^2*z+x^2*y+x^2,
x*y^2+x^2*y^2+x^2*y*z+x^2*y^2*z];
```

$$sol3 := [1+x+y+z,\ y^2+y^2\,z+y^2\,z^2+y^2\,x\,z,\ x^2\,z^2+x\,y^2\,z^2+x^2\,y\,z^2+x^2\,y^2\,z^2,\ x\,z^2+x\,y\,z^2+z^2,\ x\,y+y\,z+y\,z^2+x\,y\,z,\ x\,z+x^2\,z+x^2\,y+x^2,\ x\,y^2+x^2\,y^2+x^2\,y\,z+x^2\,y^2\,z]$$

```
i:='1':plots[display]([seq(drawtile(sol2[i],c
lrs[i]),i=1..7)] );
```



```
sol3 :=
[corner,expand(y^2*tee[1]),seq(sln.i,i=2..6)];
```

$$sol3 := [1 + x + y + z,\ y^2 + y^2 z + y^2 z^2 + y^2 x z,\ x z^2 + x^2 z^2 + x^2 y z^2 + z^2,\\ x y z^2 + x y^2 z^2 + x^2 y^2 z^2,\ x y + y z + y z^2 + x y z,\ x z + x^2 z + x^2 y + x^2,\\ x y^2 + x^2 y^2 + x^2 y z + x^2 y^2 z]$$

```
sol3 := [1+x+y+z,y^2+y^2*z+y^2*z^2+y^2*x*z,
x*z^2+x^2*z^2+x^2*y*z^2+z^2, x*y*z^2+x*y^2*z^2+x^2*y^2*z^2,
x*y+y*z+y*z^2+x*y*z, x*z+x^2*z+x^2*y+x^2,
x*y^2+x^2*y^2+x^2*y*z+x^2*y^2*z];
```

$$sol3 := [1 + x + y + z,\ y^2 + y^2 z + y^2 z^2 + y^2 x z,\ x z^2 + x^2 z^2 + x^2 y z^2 + z^2,\\ x y z^2 + x y^2 z^2 + x^2 y^2 z^2,\ x y + y z + y z^2 + x y z,\ x z + x^2 z + x^2 y + x^2,\\ x y^2 + x^2 y^2 + x^2 y z + x^2 y^2 z]$$

```
i:='1':plots[display]([seq(drawtile(sol3[i],c
lrs[i]),i=1..7)],insequence=true );
```

30

30.1 Some words

```
with(simplex):
box := proc(m,n,t)
local i,j,k;
expand(
sum(x^'i','i' = 0 ..  m-1)
* sum(y^'j','j' = 0 ..  n-1)
* sum(z^'k','k' = 0 ..  t-1))
end:
cnv := proc(f)
```

```
local tmp,tf,i,j,k,w;
tmp :=NULL;
tf := collect(simplify(f),[x,y,z]);
for i from 0 to degree(tf,x) do
for j from 0 to degree(tf,y) do
for k from 0 to degree(tf,z) do
if not coeff(coeff(coeff(tf,x,i),y,j),z,k)=0
then
tmp := tmp,coeff(coeff(coeff(tf,x,i),y,j),z,k)*w[i,j,k];
fi
od
od
od;
convert([tmp],`+`);
end:
drawtile := proc(f,clr)
local tmp,tf,i,j,k;
tmp :=NULL;
tf := collect(simplify(f),[x,y,z]);
for i from 0 to degree(tf,x) do
for j from 0 to degree(tf,y) do
for k from 0 to degree(tf,z) do
if 0 < coeff(coeff(coeff(tf,x,i),y,j),z,k)
then
tmp :=
tmp,plottools[hexahedron]([i,j,k],.4,color=clr,style=patch)
fi
od
od
od;
plots[display]([tmp]) end:
```

```
Warning, new definition for basis

Warning, new definition for convexhull

Warning, new definition for cterm

Warning, new definition for display

Warning, new definition for dual

Warning, new definition for feasible

Warning, new definition for maximize

Warning, new definition for pivotvar

Warning, new definition for ratio

Warning, new definition for setup
```

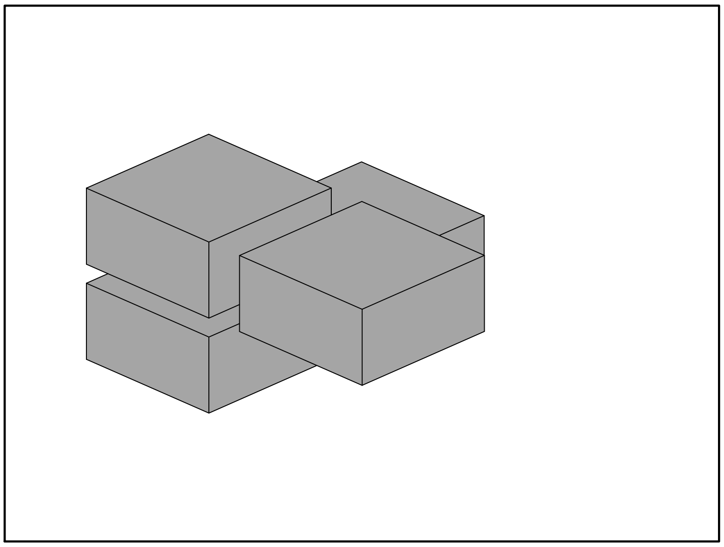
```
geteqn := proc(f,tile,`a`)
local eqn,fix,cof,j, i,t,mon;
eqn := 0:  i := 1:
```

```
fix := box(3,3,3);
for j from 1 to nops(tile) do
cof:=0:
for mon in [op(fix)] do
if nops(simplify(f+mon*tile[j] )) = nops(f) then
cof := cof+a[j,i-1]*mon; i := i+1 fi od;
eqn := eqn+expand(cof*tile[j])
od;
eqn end:
geteqns := proc(f,tile,‘a‘)
local eqn,neqn,cfs,fix,cof,j, i,t,mon;
eqn := 0:neqn:= NULL: i := 1:
fix := box(3,3,3);
for j from 1 to nops(tile) do
cof:=0:  cfs:=0:
for mon in [op(fix)] do
if nops(simplify(f+mon*tile[j] )) = nops(f) then
cof := cof+a[j,i-1]*mon;
cfs := a[j,i-1]:  i := i+1 fi od;
eqn := eqn+expand(cof*tile[j])
od;
eqn end:
drawtile(tee[1],red):
```

31

31.1 Data

```
corner := 1+x+y+z:
tee := [1+z+z^2+x*z,1+x+x^2+x*y]:
lle := [1+z+z^2+x,1+y+y^2+z,1+y +y^2+x*y^2, 1+x+x^2+y]:
le := [1+y+x*y,1+x+y,1+x+x*y,1+z+x,1+z+y,1+x+x*z,1+z+y*z]:
ze := [1+z+x*z+x*z^2,z+y*z+y+y^2,1+x+x*z+x^2*z]:
lt := [ z+x*z+x+x*y,1+z+x+x*y]:
rt := [1+z+y*z+y+x*y,1+z+y+x*y]:
corner := 1+x+y+z:
tee := {1+z+z^2+x*z }:
lle := {1+y +y^2+x*y^2 }:
le := {1+y+x*y}:
ze := {z+y*z+y+y^2 }:
lt := {z+x*z+x+x*y }:
rt := {z+y*z+y+x*y }:
drawtile(reflect(lt[1],z,x),tan);
```



```
tiles[1];
```

$$[1 + z + z^2 + x\,z]$$

```
tiles := [tee ,lle ,le ,ze ,lt ,rt ];
```

$$\mathit{tiles} := [[1 + z + z^2 + x\,z],\ [1 + y + y^2 + x\,y^2],\ [1 + y + x\,y],\ [z + y\,z + y + y^2],\ [z + x\,z + x + x\,y],\ [z + y\,z + y + x\,y]]$$

```
sixes[1][2];
```

$$x\,z^2 + x\,z + x + z$$

```
chars := [‘a‘,‘b‘,‘c‘,‘d‘,‘e‘,‘g‘]:
clrs :=
green,tan,gray,magenta,blue,red,yellow:
board := expand(
box(3,3,3)-corner-y^2*tee[1]-z^2*(y^2*x+y^2*x^2+y*x^2+x^2) );
```

$$\mathit{board} := x\,z + x\,y + x\,y^2 + y\,z + x^2\,z + x^2\,y + x^2\,y^2 + y\,z^2 + x\,z^2 + x\,y\,z^2 + x\,y\,z + x^2\,y\,z + x^2\,y^2\,z + z^2 + x^2$$

```
sixes[2];
```

$$\{1 + y + y^2 + z,\ 1 + z + z^2 + y\,z^2,\ 1 + z + z^2 + x\,z^2,\ y + y\,z + y\,z^2 + z^2,\ z + y\,z + y^2\,z + y^2,\ y + x\,y + x^2\,y + 1,\ z + x\,z + x^2\,z + 1,\ 1 + x + x^2 + z,\ y + y\,z + y\,z^2 + 1,\ 1 + y + y^2 + x\,y^2,\ 1 + y + y^2 + y^2\,z,\ 1 + x + x^2 + y,\ 1 + z + z^2 + y,\ z + y\,z + y^2\,z + 1,\ 1 + y + y^2 + x,\ y + x\,y + x^2\,y + x^2,\ x + x\,y + x\,y^2 + y^2,\ x + x\,y + x\,y^2 + 1,\ x\,z^2 + x\,z + x + z^2,\ 1 + x + x^2 + x^2\,y,\ 1 + z + z^2 + x,\ 1 + x + x^2 + x^2\,z,\ x + x\,z + x\,z^2 + 1,\ z + x\,z + x^2\,z + x^2\}$$

```
board := expand( box(3,3,3)-corner-y^2*tee[1]
);
```

$$\mathit{board} := x\,z + x\,y + x\,y^2 + y\,z + x^2\,z + x^2\,y + x^2\,y^2 + y\,z^2 + x\,z^2 + x^2\,z^2 + x\,y\,z^2 + x\,y^2\,z^2 + x^2\,y\,z^2 + x^2\,y^2\,z^2 + x\,y\,z + x^2\,y\,z + x^2\,y^2\,z + z^2 + x^2$$

```
i :='i':plots[display]([seq(drawtile(
sixes[4][i], tan),i=1..nops(sixes[4]))],insequence=true);
```

```
i := 'i':for i from 2 to 6 do
A[i] := geteqn( board, sixes[i] ,chars[i]) od:
i :='i':
gf :=cnv(sum(A[i],i=2..6)):
eqns:=seq(op(1,op(i,gf))=1 ,i=1..nops(gf)):
i:= 'i':
neqs :=
seq(convert(indets(subs({x=1,y=1,z=1},A[i])),`+`)=1,i=2..6):
nops({eqns,neqs}), nops(indets({eqns}));
```

$$24,\ 295$$

```
neqs;
```

$$\begin{aligned}
&b_{19,36}+b_{19,35}+b_{19,34}+b_{8,13}+b_{10,16}+b_{18,33}+b_{20,39}+b_{11,18}+b_{20,40}+b_{21,41}+b_{18,32}\\
&+b_{22,42}+b_{13,20}+b_{23,43}+b_{11,17}+b_{18,31}+b_{1,1}+b_{2,2}+b_{13,22}+b_{1,0}+b_{3,5}+b_{14,24}\\
&+b_{2,4}+b_{12,19}+b_{2,3}+b_{13,21}+b_{4,6}+b_{14,25}+b_{4,7}+b_{14,23}+b_{4,8}+b_{15,26}+b_{5,9}+b_{5,10}\\
&+b_{17,30}+b_{6,11}+b_{9,14}+b_{7,12}+b_{20,38}+b_{19,37}+b_{23,44}+b_{24,47}+b_{24,46}+b_{24,45}+b_{16,28}\\
&+b_{16,27}+b_{17,29}+b_{9,15}=1,c_{12,67}+c_{1,1}+c_{2,7}+c_{12,66}+c_{1,2}+c_{2,5}+c_{2,6}+c_{11,64}+c_{1,3}\\
&+c_{6,37}+c_{1,4}+c_{3,12}+c_{2,11}+c_{4,17}+c_{3,16}+c_{3,15}+c_{3,14}+c_{11,62}+c_{4,23}+c_{4,21}+c_{5,28}\\
&+c_{11,61}+c_{5,27}+c_{1,0}+c_{5,25}+c_{5,24}+c_{5,29}+c_{6,32}+c_{9,49}+c_{9,50}+c_{9,48}+c_{5,31}+c_{5,30}\\
&+c_{10,53}+c_{6,34}+c_{6,33}+c_{6,35}+c_{10,57}+c_{11,60}+c_{11,58}+c_{11,59}+c_{6,36}+c_{7,38}+c_{12,68}\\
&+c_{11,63}+c_{7,40}+c_{10,54}+c_{12,65}+c_{10,55}+c_{7,39}+c_{10,56}+c_{7,41}+c_{9,51}+c_{2,10}+c_{9,52}\\
&+c_{2,9}+c_{2,8}+c_{3,13}+c_{8,42}+c_{8,43}+c_{4,20}+c_{4,19}+c_{4,18}+c_{4,22}+c_{5,26}+c_{8,44}+c_{8,45}\\
&+c_{8,46}+c_{9,47}=1,d_{5,10}+d_{6,12}+d_{6,11}+d_{6,13}+d_{3,6}+d_{1,0}+d_{4,7}+d_{8,16}+d_{7,14}+d_{2,1}\\
&+d_{2,3}+d_{8,17}+d_{2,2}+d_{3,4}+d_{7,15}+d_{9,18}+d_{9,19}+d_{9,20}+d_{10,23}+d_{10,22}+d_{10,21}+d_{11,25}\\
&+d_{12,27}+d_{11,24}+d_{5,8}+d_{3,5}+d_{12,26}+d_{5,9}=1,e_{2,8}+e_{24,74}+e_{1,5}+e_{3,11}+e_{3,10}+e_{2,9}\\
&+e_{4,14}+e_{24,73}+e_{3,12}+e_{23,67}+e_{23,68}+e_{4,13}+e_{23,69}+e_{4,16}+e_{4,15}+e_{5,19}+e_{5,18}\\
&+e_{19,56}+e_{6,22}+e_{6,21}+e_{5,20}+e_{1,3}+e_{19,58}+e_{7,25}+e_{7,24}+e_{7,23}+e_{8,28}+e_{8,27}+e_{20,60}\\
&+e_{7,26}+e_{23,70}+e_{10,32}+e_{10,33}+e_{21,64}+e_{9,31}+e_{2,6}+e_{11,35}+e_{1,1}+e_{1,0}+e_{11,36}+e_{1,4}\\
&+e_{17,52}+e_{16,51}+e_{22,65}+e_{22,66}+e_{18,54}+e_{12,38}+e_{17,53}+e_{2,7}+e_{11,37}+e_{12,40}+e_{15,50}\\
&+e_{19,59}+e_{15,49}+e_{19,57}+e_{5,17}+e_{24,71}+e_{13,41}+e_{20,62}+e_{20,61}+e_{1,2}+e_{21,63}+e_{9,29}\\
&+e_{9,30}+e_{10,34}+e_{15,48}+e_{19,55}+e_{13,44}+e_{12,39}+e_{13,43}+e_{15,47}+e_{24,72}+e_{13,42}+e_{14,45}\\
&+e_{14,46}=1,g_{24,72}+g_{24,71}+g_{24,73}+g_{24,74}+g_{9,30}+g_{9,29}+g_{23,70}+g_{10,32}+g_{1,4}+g_{23,68}\\
&+g_{23,69}+g_{2,8}+g_{1,2}+g_{2,7}+g_{23,67}+g_{1,5}+g_{1,1}+g_{11,36}+g_{2,9}+g_{3,11}+g_{3,10}+g_{11,37}\\
&+g_{12,39}+g_{12,38}+g_{3,12}+g_{13,41}+g_{12,40}+g_{4,13}+g_{4,14}+g_{13,44}+g_{2,6}+g_{13,43}+g_{4,15}\\
&+g_{5,17}+g_{5,19}+g_{4,16}+g_{15,48}+g_{15,47}+g_{16,51}+g_{15,50}+g_{5,20}+g_{5,18}+g_{7,26}+g_{17,52}\\
&+g_{17,53}+g_{8,28}+g_{8,27}+g_{18,54}+g_{9,31}+g_{10,33}+g_{6,21}+g_{11,35}+g_{10,34}+g_{13,42}+g_{6,22}\\
&+g_{1,3}+g_{21,63}+g_{14,45}+g_{21,64}+g_{19,56}+g_{14,46}+g_{19,55}+g_{15,49}+g_{19,57}+g_{19,58}+g_{19,59}\\
&+g_{20,62}+g_{20,61}+g_{20,60}+g_{1,0}+g_{7,23}+g_{7,24}+g_{7,25}+g_{22,65}+g_{22,66}=1
\end{aligned}$$

```
infolevel[simplex]:=1:
```

```
sol:=simplex[minimize](
c[3,9]+d[10,11],[e[1,0]=1,d[10,11]=0,neqs,eqns],NONNEGATIVE);
```

$$\begin{aligned}
&\mathit{sol} := \{g_{2,7}=0,\ e_{24,73}=0,\ g_{23,69}=0,\ e_{4,14}=0,\ g_{23,68}=0,\ g_{1,4}=0,\ g_{10,32}=0,\ d_{1,0}=0,\\
&b_{11,18}=0,\ e_{1,5}=0,\ b_{8,13}=0,\ e_{3,10}=0,\ g_{23,70}=0,\ e_{3,11}=0,\ b_{20,39}=0,\ g_{9,29}=0,\ d_{6,13}=0,\\
&d_{6,12}=0,\ g_{24,74}=0,\ g_{9,30}=0,\ g_{24,73}=0,\ g_{24,72}=0,\ g_{24,71}=0,\ b_{10,16}=0,\ e_{24,74}=0,\\
&b_{19,34}=0,\ b_{19,35}=0,\ b_{19,36}=0,\ d_{5,10}=0,\ b_{21,41}=0,\ e_{2,9}=0,\ e_{3,12}=0,\ d_{4,7}=0,\\
&b_{13,20}=0,\ g_{3,10}=0,\ e_{19,56}=0,\ d_{2,2}=0,\ e_{23,67}=0,\ g_{23,67}=0,\ e_{5,18}=0,\ e_{5,19}=0,\ g_{3,11}=0,\\
&g_{2,9}=0,\ b_{22,42}=0,\ e_{4,15}=0,\ e_{4,16}=0,\ e_{23,69}=0,\ d_{2,1}=0,\ g_{11,36}=0,\ e_{23,68}=0,\ e_{4,13}=0,
\end{aligned}$$

$d_{7,14} = 0,\ d_{8,16} = 0,\ g_{1,5} = 0,\ d_{10,21} = 0,\ e_{7,26} = 0,\ g_{13,43} = 0,\ d_{10,23} = 0,\ b_{4,6} = 0,\ g_{13,44} = 0,$
$g_{4,14} = 0,\ e_{8,27} = 0,\ d_{9,20} = 0,\ e_{20,60} = 0,\ e_{7,25} = 0,\ d_{9,19} = 0,\ b_{13,21} = 0,\ e_{7,23} = 0,\ g_{4,13} = 0,$
$b_{2,3} = 0,\ e_{7,24} = 0,\ b_{14,24} = 0,\ b_{2,4} = 0,\ b_{3,5} = 0,\ b_{13,22} = 0,\ b_{1,0} = 0,\ d_{9,18} = 0,\ e_{19,58} = 0,$
$b_{2,2} = 0,\ g_{13,41} = 0,\ d_{7,15} = 0,\ g_{12,40} = 0,\ d_{3,4} = 0,\ g_{12,38} = 0,\ g_{12,39} = 0,\ e_{5,20} = 0,\ b_{1,1} = 0,$
$e_{6,21} = 0,\ b_{18,31} = 0,\ e_{6,22} = 0,\ b_{11,17} = 0,\ g_{11,37} = 0,\ b_{23,43} = 0,\ g_{15,48} = 0,\ g_{15,47} = 0,$
$b_{14,23} = 0,\ g_{5,19} = 0,\ g_{4,16} = 0,\ g_{5,17} = 0,\ e_{23,70} = 0,\ c_{3,12} = 0,\ c_{1,4} = 0,\ g_{4,15} = 0,\ e_{1,0} = 1,$
$d_{10,11} = 1,\ b_{14,25} = 0,\ e_{12,40} = 0,\ c_{10,57} = 0,\ g_{11,35} = 0,\ g_{10,34} = 0,\ c_{6,35} = 0,\ b_{20,38} = 0,$
$g_{6,21} = 0,\ g_{10,33} = 0,\ c_{6,34} = 0,\ c_{6,33} = 0,\ d_{11,24} = 0,\ b_{7,12} = 0,\ c_{10,53} = 0,\ e_{11,37} = 0,$
$c_{5,31} = 0,\ c_{5,30} = 0,\ e_{2,7} = 0,\ c_{9,48} = 0,\ c_{9,50} = 0,\ e_{17,53} = 0,\ c_{9,49} = 0,\ c_{6,32} = 0,\ g_{9,31} = 0,$
$e_{12,38} = 0,\ e_{22,66} = 0,\ g_{18,54} = 0,\ e_{18,54} = 0,\ g_{8,27} = 0,\ e_{22,65} = 0,\ e_{16,51} = 0,\ b_{6,11} = 0,$
$b_{9,14} = 0,\ g_{17,53} = 0,\ c_{11,61} = 0,\ g_{17,52} = 0,\ e_{17,52} = 0,\ b_{17,30} = 0,\ e_{1,4} = 0,\ c_{5,24} = 0,$
$d_{12,27} = 0,\ c_{5,25} = 0,\ e_{11,36} = 0,\ b_{5,10} = 0,\ c_{5,27} = 0,\ c_{11,62} = 0,\ c_{4,23} = 0,\ c_{5,28} = 0,\ g_{7,26} = 0,$
$c_{4,21} = 0,\ b_{5,9} = 0,\ e_{11,35} = 0,\ e_{9,31} = 0,\ g_{5,18} = 0,\ d_{11,25} = 0,\ e_{21,64} = 0,\ c_{3,14} = 0,\ b_{15,26} = 0,$
$g_{5,20} = 0,\ c_{3,16} = 0,\ e_{10,33} = 0,\ g_{15,50} = 0,\ e_{10,32} = 0,\ c_{4,17} = 0,\ g_{16,51} = 0,\ b_{4,8} = 0,$
$e_{14,45} = 0,\ g_{20,62} = 0,\ g_{20,61} = 0,\ c_{4,19} = 0,\ c_{4,18} = 0,\ c_{4,20} = 0,\ c_{8,43} = 0,\ e_{24,72} = 0,$
$e_{13,42} = 0,\ b_{16,28} = 0,\ b_{16,27} = 0,\ c_{8,42} = 0,\ g_{19,59} = 0,\ e_{15,47} = 0,\ e_{13,44} = 0,\ c_{3,13} = 0,$
$e_{12,39} = 0,\ e_{13,43} = 0,\ e_{15,48} = 0,\ e_{19,55} = 0,\ c_{9,52} = 0,\ c_{2,10} = 0,\ c_{2,9} = 0,\ c_{2,8} = 0,\ e_{10,34} = 0,$
$g_{19,58} = 0,\ c_{7,41} = 0,\ c_{9,51} = 0,\ d_{3,5} = 0,\ g_{15,49} = 0,\ g_{19,57} = 0,\ e_{9,30} = 0,\ b_{24,46} = 0,$
$b_{24,45} = 0,\ d_{5,8} = 0,\ e_{9,29} = 0,\ b_{24,47} = 0,\ c_{10,56} = 0,\ e_{21,63} = 0,\ g_{14,46} = 0,\ g_{19,55} = 0,$
$c_{10,55} = 0,\ c_{12,65} = 0,\ c_{7,39} = 0,\ c_{7,40} = 0,\ c_{10,54} = 0,\ g_{19,56} = 0,\ b_{23,44} = 0,\ c_{11,63} = 0,$
$g_{21,64} = 0,\ e_{20,61} = 0,\ g_{14,45} = 0,\ e_{20,62} = 0,\ c_{12,68} = 0,\ g_{21,63} = 0,\ c_{7,38} = 0,\ e_{24,71} = 0,$
$e_{13,41} = 0,\ e_{5,17} = 0,\ c_{6,36} = 0,\ e_{15,49} = 0,\ e_{19,57} = 0,\ g_{6,22} = 0,\ c_{11,59} = 0,\ c_{11,58} = 0,$
$g_{13,42} = 0,\ c_{11,60} = 0,\ b_{19,37} = 0,\ e_{19,59} = 0,\ e_{15,50} = 0,\ c_{5,29} = 0,\ g_{8,28} = 0,\ g_{2,8} = 1,$
$d_{8,17} = 0,\ d_{10,22} = 1,\ d_{6,11} = 0,\ d_{2,3} = 0,\ c_{2,11} = 0,\ c_{1,3} = 0,\ c_{3,15} = 1,\ g_{3,12} = 0,\ e_{2,8} = 0,$
$c_{6,37} = 0,\ b_{4,7} = 0,\ b_{18,33} = 0,\ c_{12,66} = 0,\ c_{12,67} = 0,\ b_{12,19} = 0,\ c_{11,64} = 0,\ d_{3,6} = 0,\ e_{8,28} = 0,$
$c_{9,47} = 0,\ b_{20,40} = 1,\ b_{18,32} = 0,\ g_{7,24} = 0,\ g_{22,66} = 0,\ c_{8,46} = 0,\ e_{14,46} = 0,\ g_{7,25} = 0,$
$g_{22,65} = 0,\ c_{8,45} = 0,\ b_{17,29} = 0,\ g_{7,23} = 0,\ b_{9,15} = 0,\ c_{8,44} = 0,\ d_{5,9} = 0,\ g_{20,60} = 0,$
$d_{12,26} = 0,\ c_{5,26} = 0,\ c_{4,22} = 0,\ g_{1,2} = 0,\ c_{1,1} = 0,\ c_{2,7} = 0,\ c_{1,2} = 0,\ c_{2,6} = 0,\ e_{1,3} = 0,$
$c_{3,9} = 0,\ g_{2,6} = 0,\ e_{2,6} = 0,\ e_{1,1} = 0,\ c_{1,0} = 0,\ g_{1,3} = 0,\ e_{1,2} = 0,\ g_{1,0} = 0,\ c_{2,5} = 0,\ g_{1,1} = 0\}$

32

32.1 More words

```
i :='i':for i from 2 to 6 do sln.i :=
expand(subs(sol,A[i])) od;
```

$$\mathit{sln2} := x\,z^2 + x^2\,z^2 + x^2\,y\,z^2 + z^2$$
$$\mathit{sln3} := x\,y\,z^2 + x\,y^2\,z^2 + x^2\,y^2\,z^2$$
$$\mathit{sln4} := x\,y + y\,z + y\,z^2 + x\,y\,z$$
$$\mathit{sln5} := x\,z + x^2\,z + x^2\,y + x^2$$
$$\mathit{sln6} := x\,y^2 + x^2\,y^2 + x^2\,y\,z + x^2\,y^2\,z$$