Math 407 Summer 2008 / Second Half

Bradley M. Bell

August 24, 2008

Contents

Chapter 1

Software

1.1 A Simple Octave / Matlab Simplex Algorithm

Matlab Documetation:
http://www.mathworks.com/support/product/product.html?product=ML

Octave Documentation:
http://www.gnu.org/software/octave/doc/interpreter/

1.1.1 Routine That Performs One Pivot Operation

Octave/Matlab Primal Pivot Routine

```
function B = pivot(A, r, c, format, fid)
% Preforms one pivot of the primal simplex algorithm.
%
% A: m by n matrix containing the tableau before the pivot operation
% r: integer between 1 and m specifying the row index for the pivot
% c: integet between 1 and n specifying the column for the pivot
% B: m by n matrix containing the tableau after the pivot operation
% format: C style format string for outputing each value in B (optional)
% fid: file id that will record the values of r, c, and B (optional)
%
% Printing B
% If r is not zero, the values (r, c), and the matrix B are written to the
% file corresponding to fid. In this case, A(r, c) must not be zero.
% If format is not present, the default format '%8g' is used.
% If fid is not present, the output is written to the screen.
%
% Printing A
% If r is zero, only the matrix A is written to the file corresponding to fid.
% If format is not present, the default format '%8g' is used.
% If fid is not present, the output is written to the screen.
% ------------------------------------------------------------------------
```

```
[m, n]     = size(A);  % size of the input and output matrices
B          = A;        % initialize B as equal to A
% if r is not equal to zero
if r ~= 0
     % pivot row of B = pivot row of A / pivot element
     B(r, :)    = A(r, :) / A(r, c);
     % for each row index 1 though m
     for i = 1 : m
          % if this is not the pivot row
          if i ~= r
               % row operation that results in zero for B(i, c)
               B(i, :) = A(i, :) - A(i, c) * B(r, :);
          end
    end
end
format_ = '%8g';
if nargin >= 4
     % format is present in argument sequence
     format_ = format;
end
fid_ = 1;  % default output is to screen
if nargin >= 5
     % fid is present in argument sequence
     fid_ = fid;
end
if r ~= 0  % if r is not zero
     fprintf(fid_, '(r, c) = (%g,%g)\n', r, c);
end
for i = 1 : m
     fprintf(fid_, format_, B(i, :));
     fprintf(fid_, '\n');
end
```

1.1.2 Example

Equation 2.1

Program: Octave/Matlab Program for Equation 2.1

```
% Equation 2.1 in Vasek Chvatal's book: Linear Programming
% maximize    5*x1 + 4*x2 + 3*x3
% subject to  2*x1 + 3*x2 +   x3 <= 5
%             4*x1 +   x2 + 2*x3 <= 11
%             3*x1 + 4*x2 + 2*x3 <= 8
%               x1, x2, x3 >= 0
% The solution is x1 = 2, x2 = 0, x3 = 1
%
fid    = fopen('eq_2_1.out', 'wt'); % file that will contain the print results
```

```
format = '%8g';                     % format for printing each tableau element
% heading for each of the colums in the tableau
fprintf(fid, '%8s',
  'x1',  'x2',  'x3',  's1',  's2',  's3', 'b');
fprintf(fid, '\n');
A = [ ... % initial tableau
    2  ,   3  ,   1  ,   1  ,   0  ,   0  ,  5
    4  ,   1  ,   2  ,   0  ,   1  ,   0  ,  11
    3  ,   4  ,   2  ,   0  ,   0  ,   1  ,  8
    5  ,   4  ,   3  ,   0  ,   0  ,   0  ,  0 ...
];
r = 0; c = 0;                      % special row value for just printing
pivot(A, r, c, format, fid);       % print initial tableau
r = 1; c = 1;                      % row and column for first pivot
B = pivot(A, r, c, format, fid);   % print r, c, and B
r = 3; c = 3;                      % row and column for next pivot
C = pivot(B, r, c, format, fid);   % print r, c, and C
fclose(fid);                       % close the file
```

Output: Octave/Matlab Output for Equation 2.1

1.1.3 Problem

We are given the problem

$$\begin{array}{rll} \text{maximize} & -5x_1 + 6x_2 - 9x_3 - 8x_4 & \text{w.r.t} \quad x \in \mathbf{R}_+^4 \\ \text{subject to} & x_1 + 2x_2 + 3x_3 + x_4 & \leq 5 \\ & x_1 + x_2 + 2x_3 + 3x_4 & \leq 3 \end{array}$$

Fill in the missing characters below (denoted by underbars) so that the Octave/Matlab program preforms one fesible pivot for the problem above:

```
A = [ ... % initial tableau
    1  ,   2  ,   3  ,  __  ,  __  ,  __  ,  5
    1  ,   1  ,   2  ,  __  ,  __  ,  __  ,  3
   __  ,  __  ,  __  ,  __  ,   0  ,   0  ,  0 ...
];
format = '%8.3g';
fprintf('%8s', 'x1', 'x2', 'x3', 'x4', 's1', 's2', 'b', '\n');
r = __; c = __;
B = pivot(A, r, c, format);
```

1.2 Neos E-mail Interface to the Clp Linear Program Solver

Purpose: The Neos server is a free public service, see
    `http://www-neos.mcs.anl.gov/neos/faq.html#who`

This is a description of how to use the Neos E-mail interface to Clp to solve linear programming problems.

Clp Mailing List: If you have problems using Clp, you are encouraged to join the Clp Mailing List and then ask the list members what the problem is.

E-mail To: When sending a problem to Neos for solution, the `To` field of your e-mail message should contain the address `neos@mcs.anl.gov` Questions about using Neos, and user feedback, can be sent to the address `neos-comments@mcs.anl.gov`

E-mail Subject: The Subject field of the e-mail message should contain a name that describes the problem being solved.

E-mail Message: The body of the e-mail message should be the following text:

```
<document>
<priority>short</priority>
<category>lp</category>
<solver>Clp</solver>
<inputMethod>MPS</inputMethod>
<comments><![CDATA[
ProblemDescription
]]></comments>
<MPS><![CDATA[*
MpsFormat
*]]></MPS>
<param><![CDATA[
ClpCommands
]]></param>
</document>
```

where the text replacements for *ProblemDescription*, *MpsFormat*, and *ClpCommands* are discussed below:

Priority: The command

```
<priority>short</priority>
```

instructs Neos that this is a short job and should have high priority (if the job does not finish quickly, it will be terminated).

Problem Description: The text *ProblemDescription* above should be replaced by information that will remind you what problem you are solving.

MPS Format: The text *MpsFormat* above should be replaced by a valid instance of MPS format. This defines the linear equations and the constraints for the linear programming problem you are solving.

Clp Commands: The text *ClpCommands* is a sequence of Clp commands.

Example: Section 1.2.3 contains an example e-mail input and solution output for this interface to Clp.

1.2.1 MPS Input File Format

Purpose: This is a specification of sufficient conditions a file to be in MPS format; i.e., if a file meets these specifications, it will be called an MPS formatted file.

References: You can find the specifications below in Section 9-2 of B.A. Murtagh's book *Advanced Linear Programming: Computation and Practice*, McGraw-Hill International Book Co., New York, 1981. You can also find a discussion of this input format in Chapter 3 of J.L. Nazareth's book *Computer Solution of Linear Programs*, Oxford University Press, New York, 1987.

Notation: We use $x \in \mathbf{R}^n$ to denote the problem variable vector, $b \in \mathbf{R}^m$ to denote the constraint right hand side vector, and $\in \mathbf{R}^{m \times n}$ to denote the constraint matrix.

File Structure: An MPS input file has the following structure:
NameRecord
RowsRecord
ColumnsRecord
RhsRecord
BoundsRecord
Endata

Fields: We use the field names below to refer to the corresponding columns:

Field	*Keyword*	*Op*	*Name0*	*Name1*	*Value1*	*Name2*	*Value2*
Columns	1-*	2-3	5-12	15-22	25-36	40-47	50-61

Comment Lines: A comment line has a `*` in column one and any characters in the rest of the line. Any other character in column one is the start of a Keyword.

Keyword: *Keyword* is special in that it always begins in column one and extends to the first space character in the line. If the Keyword field extends into another field, the field it extends into is not present in that line.

Name Fields: It is unspecified if leading and trailing white space has any significance in *Name0*, *Name1*, and *Name2*. For example, the seven characters between the quotes "`x1     `" may or may not signify the same information as the seven characters "`     x1`".

Value Fields: Leading and trailing white space in the Value fields is ignored.

Name Record: The *NameRecord* is a single line with the following structure:
Keyword Name1
where *Keyword* is `NAME` and *Name1* is the name of the problem for this input file.

Rows Record: The first line of *RowsRecord* contains the text
Keyword
where *Keyword* is `ROWS`. Each of the other lines in the *RowsRecord* has the following structure:
Op Name0

In the i-th line below the `ROWS` line, *Name0* specifies the name of the i-th row of the matrix; i.e., the name associated with the value

$$A_{i,1}x_1 + \ldots + A_{i,n}x_n$$

The Op field specifies the type of the row and is one of the following values:

Op	Equation or inequality
`E`	$A_{i,1}x_1 + \ldots + A_{i,n}x_n = b_i$
`G`	$A_{i,1}x_1 + \ldots + A_{i,n}x_n \geq b_i$
`L`	$A_{i,1}x_1 + \ldots + A_{i,n}x_n \leq b_i$
`N`	$A_{i,1}x_1 + \ldots + A_{i,n}x_n = z$

The objective function is defined as the first row in which *Op* is equal to `N`. The other rows with *Op* equal to `N` are ignored.

Columns Record: The *ColumnsRecord* is used to name the variables x_1, through x_n and to set the value of the non-zero coefficients in the matrix A (zero is the default value for $A_{i,j}$). The first line of the *ColumnsRecord* contains the text

Keyword

where *Keyword* is `COLUMNS`. Each of the other lines has the structure

Name0 Name1 Value1

or it has the structure

Name0 Name1 Value1 Name2 Value2

Name0: The string *Name0* specifies the name corresponding to the variable x_j for the information on the line. All of the lines corresponding to one variable (all the lines with the same value for *Name0*) must be grouped together. The record is called the *ColumnsRecord* because each variable corresponds to a column of the matrix A.

Name1, Value1: The string *Name1* must be one of the row names in the *RowsRecord*. The number *Value1* specifies the value of $A_{i,j}$ where i corresponds to *Name1* and j corresponds to *Name0* for this line.

Name2, Value2: If *Name2* is present, it must also be equal to one of the row names in the *RowsRecord*. In this case, the number *Value2* specifies the value of $A_{i,j}$ where i corresponds to *Name2* and j corresponds to *Name0* for this line.

Rhs Record: This record is used to specify non-zero coefficients in the right hand side for each row of the matrix A; i.e., each equation or inequality (zero is the default right hand side value). The first line of the *RhsRecord* contains the text

Keyword

where *Keyword* is `RHS`. Each of the other lines has the structure Each of the other lines has the structure

Name0 Name1 Value1

Name0: The string *Name0* is a name for the entire right hand side vector $b \in \mathbf{R}^m$. It must be the same for all of the lines in the *RhsRecord*.

Name1: The string *Name1* must be one of the names in the *RowsRecord*. It specifies the row index i

Value1: corresponding to the information on this line. The number *Value1* specifies the value for this right hand side coefficient; i.e., b_i.

Bounds Record: For each variable x_1 through x_n, the default lower bound is zero and the default upper bound is plus infinity; i.e., by default

$$0 \leq x_j < \infty$$

If these are the only bounds you want, you need not include *BoundsRecord*. The first line of *BoundsRecord* contains the text

Keyword

where *Keyword* is `BOUNDS`. Each of the other lines has the structure

Op Name0 Name1

or the structure

Op Name0 Name1 Value1

Op, Value1: The string *Op* specifies the type of each bound and is one of the following values:

Op	Description	
`LO`	lower bound	$Value1 \leq x_j < +\infty$
`UP`	upper bound	$0 \leq x_j \leq Value1$
`FX`	fixed variable	$x_j = Value1$
`FR`	free variable	$-\infty < x_j < +\infty$
`MI`	minus infinity	$-\infty < x_j \leq 0$
`PL`	plus infinity	$0 \leq x_j < +\infty$

Name0: The string *Name0* specifies a name for the bound. It must be the same for all of these lines.

Name1: The string *Name1* must be one of the variable names in the *Name0* field of the *ColumnsRecord*. The corresponding variable x_j has it's bounds set by the current line.

Endata Record: The *Endata Record* is just one line

Keyword

where *Keyword* is `ENDATA`.

Example: In Section 1.2.3 the MPS input file is the text below the line

```
<MPS><![CDATA[*
```

and above the line

```
*]]></MPS>
```

1.2.2 Some Clp Executable Commands

Purpose: The executable for the Clp linear programming solver has an extensive set of commands. This is section contains documentation for some of these commands.

Convention: The command syntax

leading(trailing)

means that the characters in *leading* are necessary and that any number of the characters in *trailing* are optional. If the *trailing* characters are present, they need not all appear but those that do must be in order. Some of the commands below are of limited use or not helpful at all when using the Neos e-mail interface to Clp. In this case, a comment to this effect is added at the end of the command description.

direction: The syntax for this command is one of the following three choices:

`direction min`*(*`imize`*)*
`direction max`*(*`imize`*)*
`direction zero`

This sets the objective direction for optimization (the default is minimize). You can also set the objective direction using the commands `minimize` and `maximize`.

directory: The syntax for this command is

`directory` *DefaultDirectory*

This sets the default directory for reading and writing files. To be specific, the commands `import`, `export`, `saveModel`, and `restoreModel`, The initial value for this default directory is the current working directory; i.e. `./`. This command is not useful with the e-mail Neos interface to Clp.

import: The syntax for this command is

`import` *MpsFileName*

This will read the Mps formatted file specified by *MpsFileName*. It will use the default directory as specified by the `directory` command. If *MpsFileName* is equal to `@`, the previous value for the file name is used. The file name is initially empty; i.e. there is no default value and it must be set. If you have `libgz`, Clp can read compressed files by ending *MpsFileName* with the three characters `.gz`. This command is not useful with the e-mail Neos interface to Clp.

maximize: The syntax for this command

`max`*(*`imize`*)*

This sets the optimization direction to maximize. (The default optimization direction is to minimize.) You can also use the command `direction` to set the optimization direction to maximize.

minimize: The syntax for this command

`min`*(*`imize`*)*

This sets the optimization direction to minimize. (The default optimization direction is to minimize.) You can also use the command `direction` to set the optimization direction to minimize.

primalPivot: The syntax for this command is

`primalP`*(*`ivot`*) selection*

This command determines the primal pivot selection method. The possible values for *selection* are:

`auto`*(*`matic`*)*
`exa`*(*`ct`*)*
`dant`*(*`zig`*)*
`part`*(*`ial`*)*
`steep`*(*`est`*)*
`change`
`sprint`

The Dantzig method is implemented to show a simple method but its use is deprecated. Exact devex is the method of choice and there are two variants which keep all weights updated but only scan a subset each iteration. Partial switches this on while change initially does dantzig until the factorization becomes denser. This is still a work in progress.

primalSimplex: The syntax for this command is

`primalS`*(*`implex`*)*

This command solves the current model using the primal simplex algorithm. The default is to use the exact devex column selection method. The time and iterations may be effected by settings such as presolve, scaling, crash and also by column selection method, infeasibility weight and dual and primal tolerances.

printingOptions: The syntax for this command is

`printi`*(*`ngOptions`*) option*

where *option* is one of the following choices:

option	Description
`normal`	print the non-zero column variable values
`rows`	print non-zero column variable and row activities
`all`	print all column variable and row activities

The default value for this option is `normal`.

solution: The syntax for this command is

`solu`*(*`tion`*) SolutionFileName*

print the current solution to the file specified by *SolutionFileName*. This will use the default directory as specified by the previous `directory` command. If *SolutionFileName* is `@`, the previous value for the solution file name is used. This solution file name is initialized to standard output. The amount of output can be varied using the `printing options` or `printMask`. We use m for the number of rows and n for the number of columns in the matrix A specified by the Mps formatted input. The solution is output in two parts: The `rows` section of the solution contains the following lines: for $i = 0, \ldots, m-1$

i NameRow_i PrimalRow_i DualRow_i

The `columns` section of the solution contains the following lines: for $j = 0, \ldots, n-1$

j NameColumn_j PrimalalueColumn_j DualColumn_j

If you are using this command with the e-mail Noes interface to Clp, you should use *SolutionFileName* equal to `-` which instructs Clp to write the solution to standard output.

1.2.3 Example: Neos E-mail Interface to Clp

Input

The e-mail message sent to Neos for this problem was: Neos Clp Input for Equation 2.1

```
<document>

<category>lp</category>
<solver>Clp</solver>
<inputMethod>MPS</inputMethod>

<comments><![CDATA[
Equation 2.1 in Vasek Chvatal's book: Linear Programming
```

```
maximize    5*x1 + 4*x2 + 3*x3              z
subject to  2*x1 + 3*x2 +   x3 <= 5         r1
            4*x1 +   x2 + 2*x3 <= 11        r2
            3*x1 + 4*x2 + 2*x3 <= 8         r3
              x1 ,   x2 ,   x3 >= 0
The solution is x1 = 2, x2 = 0, x3 = 1
The residuals are 0=s1=5-r1, 1=s2=11-r2, 0=s3=8-r3
]]></comments>

<MPS><![CDATA[*
*Op Name0---  Name1---  Value1------   Name2---  Value2------
*23 56789012  56789012  567890123456   01234567  012345678901
NAME          eq_2.1
ROWS
 N  z
 L  r1
 L  r2
 L  r3
COLUMNS
    x1        z         5              r1        2
    x1        r2        4              r3        3
*
    x2        z         4              r1        3
    x2        r2        1              r3        4
*
    x3        z         3              r1        1
    x3        r2        2              r3        2
RHS
    b         r1        5
    b         r2        11
    b         r3        8
* The next two lines are not necessary because 0 <= x1 is the default bound
BOUNDS
 LO e         x1        0
ENDATA
*]]></MPS>

<param><![CDATA[
maximize
primalSimplex
printingOptions all
solution -
]]></param>

</document>
```

Confirmation

The following confirmation was returned when Neos queued this problem for execution: Noes Clp Confirmation for Equation 2.1

```
From neos@mcs.anl.gov Sat Jul 12 07:36:43 2008
Date: Sat, 12 Jul 2008 09:36:42 -0500
From: Neos Server <neos@mcs.anl.gov>
To: Some One <some_one@some_where.edu>
Subject: NEOS Confirmation for Job #1672727

NEOS has received and assigned your submission as job #1672727, password 'hFLyEHzq'
Current queue:
 Running:
job#1672671  go    ASA      AMPL  submitted 07/11 23:58 started 07/12 08:50 ...
job#1672701  milp  scip     AMPL  submitted 07/12 03:05 started 07/12 03:06 ...
job#1672702  milp  scip     AMPL  submitted 07/12 03:08 started 07/12 03:09 ...
job#1672722  milp  scip     AMPL  submitted 07/12 07:07 started 07/12 07:08 ...
job#1672723  milp  scip     AMPL  submitted 07/12 07:09 started 07/12 07:09 ...

Queued:
job#1672727  lp    Clp      MPS   submitted 07/12 09:36
```

Output

The following results were returned when Neos finished this problem: Noes Clp Output for Equation 2.1

```
From neos@mcs.anl.gov Sat Jul 12 07:41:43 2008
Date: Sat, 12 Jul 2008 09:41:43 -0500
From: neos@mcs.anl.gov
To: some_one@some_where.edu
Subject: NEOS Results for Job #1672727

You are using the solver clp-mps.

\%%%%%%%%%%%%%%%%%%%% CLP Results %%%%%%%%%%%%%%%%%%%

Load Avg: ( 0.0 , 0.0 , 0.0 )
Coin LP version 1.03.03, build Aug 18 2006
command line - /home/neos/neos-bin/clp clp.mps -
At line 4 NAME          eq_2.1
At line 5 ROWS
At line 10 COLUMNS
At line 19 RHS
At line 23 ENDATA
Problem CE-2.1 has 3 rows, 3 columns and 9 elements
Model was imported from ./clp.mps in 0.001 seconds
Switching to line mode
```

```
Clp:Clp:Clp:Presolve 3 (0) rows, 3 (0) columns and 9 (0) elements
0  Obj -0 Dual inf 15 (3)
2  Obj 13
Optimal - objective value 13
Optimal objective 13 - 2 iterations time 0.002
Clp:Clp:
      0 r1                 5                       1
      1 r2                10                      -0
      2 r3                 8                       1
      0 x1                 2          1.110223e-16
      1 x2                 0                      -3
      2 x3                 1         5.5511151e-17
Clp:

%%%%%%%%%%%%%%%%%%%% CLP Results %%%%%%%%%%%%%%%%%%%
```

1.2.4 Problem

$$\begin{array}{lll} \text{maximize} & 3x_1 + x_2 & \text{w.r.t} \quad x \in \mathbf{R}^2_+ \\ \text{subject to} & x_1 - x_2 & \leq -1 \\ & -x_1 - x_2 & \leq -3 \\ & 2x_1 + x_2 & \leq 4 \end{array}$$

Fill in the missing characters denoted by _ in the message below so that e-mailing the resulting message to Neos would solve the problem above. Do not fill in any text for characters that should be a space.

```
<document>
<category>lp</category>
<solver>Clp</solver>
<inputMethod>MPS</inputMethod>
<comments><![CDATA[
Problem 3.9a in Vasek Chvatal's book: Linear Programming
]]></comments>
<MPS><![CDATA[*
*Op Name0---  Name1---  Value1------   Name2---  Value2------
*23 56789012  56789012  567890123456   01234567  012345678901
NAME          pr_3.9a
ROWS
 N  z
__  r1
__  r2
__  r3
COLUMNS
    x1        z         3              r1        1
    x1        r2        -1             r3        2
*
    __        __        __             __        -1
```

```
    x2        r2        -1             r3        1
RHS
    b         r1        __
    b         r2        __
    b         r3        __
ENDATA
*]]></MPS>
<param><![CDATA[
maximize
primalSimplex
printingOptions all
solution -
]]></param>
</document>
```

Chapter 2

General Linear Programming Duality

Notation: We are given the following information:

$m_= \in \mathbf{Z}_+$	number of equality constraints
$m_\leq \in \mathbf{Z}_+$	number of inequality constraints
$n_\pm \in \mathbf{Z}_+$	number of free variables
$n_+ \in \mathbf{Z}_+$	number of non-negative variables
$c_\pm \in \mathbf{R}^{n\pm}$	objective coefficients corresponding to the free variables
$c_+ \in \mathbf{R}^{n+}$	objective coefficients corresponding to the non-negative variables
$b_= \in \mathbf{R}^{m=}$	constraint value corresponding to equality constraints
$b_\leq \in \mathbf{R}^{m\leq}$	constraint limit corresponding to inequality constraints
$A_{=,\pm} \in \mathbf{R}^{m= \times n\pm}$	matrix block, equality constraints and free variables
$A_{=,+} \in \mathbf{R}^{m= \times n+}$	matrix block, equality constraints and non-negative variables
$A_{\leq,\pm} \in \mathbf{R}^{m\leq \times n\pm}$	matrix block, inequality constraints and free variables
$A_{\leq,+} \in \mathbf{R}^{m\leq \times n+}$	matrix block, inequality constraints and non-negative variables

Note that we use $m=$ for $m_=$ when it appears as a superscript or subscript. We use a similar notation for the other indices.

General Primal Problem:

$$\begin{array}{llll}
\text{maximize} & (c_\pm^{\mathrm{T}} \;,\; c_+^{\mathrm{T}}) \begin{pmatrix} x_\pm \\ x_+ \end{pmatrix} & \text{w.r.t.} & \begin{array}{l} x_\pm \in \mathbf{R}^{n\pm} \\ x_+ \in \mathbf{R}_+^{n+} \end{array} \\
\text{subject to} & \begin{pmatrix} A_{=,\pm} & A_{=,+} \\ A_{\leq,\pm} & A_{\leq,+} \end{pmatrix} \begin{pmatrix} x_\pm \\ x_+ \end{pmatrix} \begin{array}{c} = \\ \leq \end{array} \begin{pmatrix} b_= \\ b_\leq \end{pmatrix} & &
\end{array}$$

Primal in Standard Form: Define $x_\pm = x_\pm^+ - x_\pm^-$ where $x_\pm^+ \in \mathbf{R}_+^{n\pm}$ and $x_\pm^- \in \mathbf{R}_+^{n\pm}$. The primal problem above is equivalent to

$$\begin{array}{llll} \text{maximize} & (c_\pm^{\mathrm{T}}\ ,\ -c_\pm^{\mathrm{T}}\ ,\ c_+^{\mathrm{T}}) \begin{pmatrix} x_\pm^+ \\ x_\pm^- \\ x_+ \end{pmatrix} & \text{w.r.t.} & \begin{array}{l} x_\pm^+ \in \mathbf{R}_+^{n\pm} \\ x_\pm^- \in \mathbf{R}_+^{n\pm} \\ x_+ \in \mathbf{R}_+^{n+} \end{array} \\ \text{subject to} & \begin{pmatrix} A_{=,\pm} & -A_{=,\pm} & A_{=,+} \\ -A_{=,\pm} & A_{=,\pm} & -A_{=,+} \\ A_{\leq,\pm} & -A_{\leq,\pm} & A_{\leq,+} \end{pmatrix} \begin{pmatrix} x_\pm^+ \\ x_\pm^- \\ x_+ \end{pmatrix} & \leq \begin{pmatrix} b_= \\ -b_= \\ b_\leq \end{pmatrix} & \end{array}$$

Dual in Standard Form:

$$\begin{array}{llll} \text{minimize} & (b_=^{\mathrm{T}}\ ,\ -b_=^{\mathrm{T}}\ ,\ b_\leq^{\mathrm{T}}) \begin{pmatrix} y_=^+ \\ y_=^- \\ y_\leq \end{pmatrix} & \text{w.r.t} & \begin{array}{l} y_=^+ \in \mathbf{R}_+^{m=} \\ y_=^- \in \mathbf{R}_+^{m=} \\ y_\leq \in \mathbf{R}_+^{m\leq} \end{array} \\ \text{subject to} & \begin{pmatrix} A_{=,\pm}^{\mathrm{T}} & -A_{=,\pm}^{\mathrm{T}} & A_{\leq,\pm}^{\mathrm{T}} \\ -A_{=,\pm}^{\mathrm{T}} & A_{=,\pm}^{\mathrm{T}} & -A_{\leq,\pm}^{\mathrm{T}} \\ A_{=,+}^{\mathrm{T}} & -A_{=,+}^{\mathrm{T}} & A_{\leq,+}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} y_=^+ \\ y_=^- \\ y_\leq \end{pmatrix} & \geq \begin{pmatrix} c_\pm \\ -c_\pm \\ c_+ \end{pmatrix} & \end{array}$$

General Dual Problem: Define $y_= \in \mathbf{R}^{m=}$ by $y_= = y_=^+ - y_=^-$. The general dual form is

$$\begin{array}{llll} \text{minimize} & (b_=^{\mathrm{T}}\ ,\ b_\leq^{\mathrm{T}}) \begin{pmatrix} y_= \\ y_\leq \end{pmatrix} & \text{w.r.t} & \begin{array}{l} y_= \in \mathbf{R}^{m=} \\ y_\leq \in \mathbf{R}_+^{m\leq} \end{array} \\ \text{subject to} & \begin{pmatrix} A_{=,\pm}^{\mathrm{T}} & A_{\leq,\pm}^{\mathrm{T}} \\ A_{=,+}^{\mathrm{T}} & A_{\leq,+}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} y_= \\ y_\leq \end{pmatrix} & \begin{array}{c} = \\ \geq \end{array} \begin{pmatrix} c_\pm \\ c_+ \end{pmatrix} & \end{array}$$

General Duality Theorem: The following combinations for the general primal and dual problems are possible and impossible:

	Dual Optional	Dual Infeasible	Dual Unbounded
Primal Optimal	Possible	Impossible	Impossible
Primal Infeasible	Impossible	Possible	Possible
Primal Unbounded	Impossible	Possible	Impossible

In the case where the primal and dual have optimal values, these optimal values are equal. If $(x_\pm, x_+)$ is feasible for the primal and $(y_\pm, y_+)$ is feasible for the dual,

$$(c_\pm^{\mathrm{T}}\ ,\ c_+^{\mathrm{T}}) \begin{pmatrix} x_\pm \\ x_+ \end{pmatrix} \leq (y_=^{\mathrm{T}}\ ,\ y_\leq^{\mathrm{T}}) \begin{pmatrix} A_{=,\pm} & A_{=,+} \\ A_{\leq,\pm} & A_{\leq,+} \end{pmatrix} \begin{pmatrix} x_\pm \\ x_+ \end{pmatrix} \leq (b_=^{\mathrm{T}}\ ,\ b_\leq^{\mathrm{T}}) \begin{pmatrix} y_= \\ y_\leq \end{pmatrix}$$

2.1 Matrix Games

Introduction: We are given two players X and Y and a matrix $A \in \mathbf{R}^{m\times n}$. If player X makes choice $j \in \{1, \ldots, n\}$ and player Y makes choice $i \in \{1, \ldots, m\}$, player X wins (player Y loses) $A_{i,j}$ units.

Strategy: We use the notation $\mathbf{P}(n)$ to denote the set of probability measures on the indices $\{1, \ldots, n\}$; i.e.,

$$\mathbf{P}(n) = \left\{ x \in \mathbf{R}^n_+ \text{ such that } \sum_{j=1}^{n} x_j = 1 \right\}$$

If player X adopts a strategy $x \in \mathbf{P}(n)$ and player Y adopts a strategy $y \in \mathbf{P}(m)$, the expected winnings for player X (losing for player Y) is

$$y^{\mathrm{T}} A x = \sum_{i=1}^{m} \sum_{j=1}^{n} y_i A_{i,j} x_j$$

Lemma:

$$\inf \left\{ \left[\sup\{ y^{\mathrm{T}} A x \mid x \in \mathbf{P}(n) \} \right] \,\middle|\, y \in \mathbf{P}(m) \right\} \geq \sup \left\{ \left[\inf\{ y^{\mathrm{T}} A x \mid y \in \mathbf{P}(m) \} \right] \,\middle|\, x \in \mathbf{P}(n) \right\}$$

Proof: For all $\hat{x} \in \mathbf{P}(n)$ and all $\hat{y} \in \mathbf{P}(m)$,

$$\sup\{ \hat{y}^{\mathrm{T}} A x \mid x \in \mathbf{P}(n) \} \geq \hat{y}^{\mathrm{T}} A \hat{x} \geq \inf\{ y^{\mathrm{T}} A \hat{x} \mid y \in \mathbf{P}(m) \}$$

Taking the inf of the left side with respect to $\hat{y} \in \mathbf{P}(m)$ and the sup of the right side with respect to $\hat{x} \in \mathbf{P}(n)$ we obtain the conclusion of the lemma.

Lemma: For each $\hat{x} \in \mathbf{R}^n$ and $\hat{y} \in \mathbf{R}^m$,

$$\begin{aligned} \inf \left\{ y^{\mathrm{T}} A \hat{x} \mid y \in \mathbf{P}(m) \right\} &= \min \left\{ \sum_{j=1}^{n} A_{i,j} \hat{x}_j \mid i \in \{1, \ldots, m\} \right\} \\ \sup \left\{ \hat{y}^{T} A x \mid x \in \mathbf{P}(n) \right\} &= \max \left\{ \sum_{i=1}^{m} \hat{y}_i A_{i,j} \mid j \in \{1, \ldots, n\} \right\} \end{aligned}$$

Proof: Let $k \in \{1, \ldots, m\}$ be an index such that

$$\sum_{j=1}^{n} A_{k,j} \hat{x}_j = \min \left\{ \sum_{j=1}^{n} A_{i,j} \hat{x}_j \mid i \in \{1, \ldots, m\} \right\}$$

Define $\bar{y} \in \mathbf{R}^m$ by $\bar{y}_k = 1$ and for $i \neq k$, $\bar{y}_i = 0$. Note that $\bar{y} \in \mathbf{P}(m)$ and that,

$$\begin{aligned} \sum_{j=1}^{n} A_{k,j} \hat{x}_j &= \bar{y}^{\mathrm{T}} A \hat{x} \\ \min \left\{ \sum_{j=1}^{n} A_{i,j} \hat{x}_j \mid i \in \{1, \ldots, m\} \right\} &\geq \inf \left\{ y^{\mathrm{T}} A \hat{x} \mid y \in \mathbf{P}(m) \right\} \end{aligned}$$

But, from the choice of k, we also have that for any $\tilde{y} \in \mathbf{P}(m)$,

$$
\begin{aligned}
\sum_{j=1}^{n} A_{k,j} x_j &= \left(\sum_{j=1}^{n} A_{k,j} x_j \right) \sum_{i=1}^{m} \tilde{y}_i \\
\min \left\{ \sum_{j=1}^{n} A_{i,j} \hat{x}_j \mid i \in \{1, \ldots, m\} \right\} &= \sum_{i=1}^{m} \tilde{y}_i \sum_{j=1}^{n} A_{k,j} x_j \\
&\leq \sum_{i=1}^{m} \tilde{y}_i \sum_{j=1}^{n} A_{i,j} x_j \\
&\leq \tilde{y}^{\mathrm{T}} A \hat{x} \\
&\leq \inf \left\{ y^{\mathrm{T}} A \hat{x} \mid y \in \mathbf{P}(m) \right\}
\end{aligned}
$$

This completes the proof of the first inequality in the lemma. This other inequality can be proved in a similar way.

Primal: Suppose that player X chooses her strategy first and then player Y gets to choose his strategy. It follows that player X should choose the strategy $x \in \mathbf{P}(n)$ that solves the problem

$$\text{maximize } \left[\inf\{ y^{\mathrm{T}} A x \mid y \in \mathbf{P}(m) \} \right] \text{ w.r.t } x \in \mathbf{P}(n)$$

We use the lemma above to convert the primal problem to the following equivalent linear programing problem

$$
\begin{array}{llll}
\text{maximize} & z & \text{w.r.t} & x \in \mathbf{R}_{+}^{n} \\
\text{subject to} & \sum_{j=1}^{n} A_{i,j} x_j & \geq z & (i = 1, \cdots, m) \\
 & \sum_{j=1}^{n} x_j & = 1 &
\end{array}
$$

We use the notation $0_{m \times n}$ ($1_{m \times n}$) to denote an $m \times n$ matrix of zeros (ones). The problem above can be written as

$$
\begin{array}{llll}
\text{maximize} & (1\ ,\ 0_{1 \times n}) \begin{pmatrix} z \\ x \end{pmatrix} & \text{w.r.t.} & \begin{array}{c} z \in \mathbf{R} \\ x \in \mathbf{R}_{+}^{n} \end{array} \\
\text{subject to} & \begin{pmatrix} 0 & 1_{1 \times n} \\ 1_{m \times 1} & -A \end{pmatrix} \begin{pmatrix} z \\ x \end{pmatrix} & \begin{array}{c} = \\ \leq \end{array} & \begin{array}{c} 1 \\ 0_{m \times 1} \end{array}
\end{array}
$$

Dual: The problem above has the form of a general primal problem. The corresponding dual problem is

$$
\begin{array}{llll}
\text{minimize} & (1\ ,\ 0_{1 \times m}) \begin{pmatrix} w \\ y \end{pmatrix} & \text{w.r.t.} & \begin{array}{c} w \in \mathbf{R} \\ y \in \mathbf{R}_{+}^{m} \end{array} \\
\text{subject to} & \begin{pmatrix} 0 & 1_{1 \times m} \\ 1_{n \times 1} & -A^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} w \\ y \end{pmatrix} & \begin{array}{c} = \\ \geq \end{array} & \begin{pmatrix} 1 \\ 0_{n \times 1} \end{pmatrix}
\end{array}
$$

This problem is equivalent to

$$\text{minimize } \left[\sup\{ y^{\mathrm{T}} A x \mid x \in \mathbf{P}(n) \} \right] \text{ w.r.t } y \in \mathbf{P}(m)$$

This problem corresponds to the case where player Y makes his choice first and then player X makes her choice.

Theorem: Both the primal and dual games above have an optimal solutions and the optimal values are equal; i.e.,

$$\inf \left\{ \left[\sup\{y^{\mathrm{T}} A x \mid x \in \mathbf{P}(n)\}\right] \,\middle|\, y \in \mathbf{P}(m) \right\} = \sup \left\{ \left[\inf\{y^{\mathrm{T}} A x \mid y \in \mathbf{P}(m)\}\right] \,\middle|\, x \in \mathbf{P}(n) \right\}$$

2.1.1 Example

In problem 15.1 of the text player X and player Y hide either a nickel or a dime. If the two coins match, X gets both. Otherwise, Y gets both. We define

x_1	probability that X chooses for playing a nickel
x_2	probability that X chooses for playing a dime
y_1	probability that Y chooses for playing a nickel
y_2	probability that Y chooses for playing a dime

Payoff Matrix: The payoff matrix for this game is

$$A = \begin{pmatrix} 5 & -10 \\ -5 & 10 \end{pmatrix}$$

Primal Problem: The corresponding primal problem is

$$\begin{array}{rccl} \text{maximize} & z & \text{w.r.t} & z \in \mathbf{R}\ ,\ x_1 \in \mathbf{R}_+\ ,\ x_2 \in \mathbf{R}_+ \\ \text{subject to} & x_1 + x_2 & = 1 & \\ & z - 5x_1 + 10x_2 & \leq 0 & \\ & z + 5x_1 - 10x_2 & \leq 0 & \end{array}$$

Standard Form: Replacing $z \in \mathbf{R}$ by $z = z^+ - z^-$ where $z^+ \in \mathbf{R}_+$ and $z^- \in \mathbf{R}_+$ and using $x_2 = 1 - x_1$ to replace x_2 we obtain the following standard form representation of the problem

$$\begin{array}{rccl} \text{minimize} & z^+ - z^- & \text{w.r.t} & z^+ \in \mathbf{R}_+\ ,\ z^- \in \mathbf{R}_+\ ,\ x_1 \in \mathbf{R}_+ \\ \text{subject to} & x_1 & \leq 1 & \\ & z^+ - z^- - 5x_1 + 10(1 - x_1) & \leq 0 & \\ & z^+ - z^- + 5x_1 - 10(1 - x_1) & \leq 0 & \end{array}$$

Solution Using Simplex Method

We can use x_2 as as the slack variable and after one pivot obtain the tableau corresponding to the problem above. Then, after a pivot with z^- as the entering variable and the smallest right hand side as the leaving variable, we obtain a feasible basis. From there the simplex method proceeds as normal. Here is the corresponding output printed by the program on 5: Pivot Routine Output for Problem 15.1

So the solution to the standard form version of the problem is $z^+ = 0$, $z^- = 0$, $x_1 = 2/3$, $x_2 = 1/3$. Also note the solution to the dual variables $y_1 = 1/2$ and $y_2 = 1/2$ can be found along the bottom row (because the slack variable s_1 corresponds to y_1 and s_2 corresponds to y_2).

Check Answer: The dual problem is

$$\begin{array}{rccl} \text{minimize} & w & \text{w.r.t} & w \in \mathbf{R}\ ,\ y_1 \in \mathbf{R}_+\ ,\ y_2 \in \mathbf{R}_+ \\ \text{subject to} & y_1 + y_2 & = 1 & \\ & w - 5y_1 + 5y_2 & \geq 0 & \\ & w + 10y_1 - 10y_2 & \geq 0 & \end{array}$$

It suffices to check that $z = 0$, $x_1 = 2/3$, and $x_2 = 1/3$ is feasible for the primal and that $w = 0$, $y_1 = 1/2$, and $y_2 = 1/2$ is feasible for the dual.

Solution Using Neos and Clp

Input: The e-mail message sent to Neos (see page 8) for this problem was: Neos Clp Input for Problem 15.1

```
<document>

<category>lp</category>
<solver>Clp</solver>
<inputMethod>MPS</inputMethod>

<comments><![CDATA[
Problem 15.1 in Vasek Chvatal's book: Linear Programming
maximize    x0                              z
subject to        x1 + x2      = 1          r1
            x0 - 5*x1 + 10*x2 <= 0          r2
            x0 + 5*x1 - 10*x2 <= 0          r3
              x1 , x2 >= 0
The primal solution is x0 = 0, x1 = 2/3, x2 = 1/3
]]></comments>

<MPS><![CDATA[*
*Op Name0---  Name1---  Value1------   Name2---  Value2------
*23 56789012  56789012  567890123456   01234567  012345678901
NAME          pr_15.1
ROWS
 N  z
 E  r1
 L  r2
 L  r3
COLUMNS
    x0        z         1              r1        0
    x0        r2        1              r3        1
*
    x1        r1        1              r2        -5
    x1        r3        5
*
    x2        r1        1              r2        10
    x2        r3        -10
RHS
    b         r1        1
    b         r2        0
    b         r3        0
BOUNDS
 FR e         x0
```

```
ENDATA
*]]></MPS>

<param><![CDATA[
maximize
primalSimplex
printingOptions all
solution -
]]></param>

</document>
```

Output: The following results were returned when Neos finished this problem: Neos Clp Output for Problem 15.1

```
From neos@mcs.anl.gov Tue Jul 22 08:23:30 2008
Date: Tue, 22 Jul 2008 10:23:29 -0500
From: neos@mcs.anl.gov
To: some_one@some_where.edu
Subject: NEOS Results for Job #1677349

You are using the solver clp-mps.

\%%%%%%%%%%%%%%%%%%% CLP Results %%%%%%%%%%%%%%%%%%%

Load Avg: ( 0.03 , 0.05 , 0.01 )
Coin LP version 1.03.03, build Aug 18 2006
command line - /home/neos/neos-bin/clp clp.mps -
At line 4 NAME          pr_15.1
At line 5 ROWS
At line 10 COLUMNS
At line 19 RHS
At line 23 BOUNDS
At line 25 ENDATA
Problem pr_15.1 has 3 rows, 3 columns and 8 elements
Model was imported from ./clp.mps in 0 seconds
Switching to line mode
Clp:Clp:Clp:Presolve 2 (-1) rows, 2 (-1) columns and 4 (-4) elements
0  Obj -0 Primal inf 0.666667 (1) Dual inf 1.01e+10 (2)
2  Obj -0
Optimal - objective value -0
After Postsolve, objective 0, infeasibilities - dual 0 (0), primal 0 (0)
Optimal objective 0 - 2 iterations time 0.002, Presolve 0.00
Clp:Clp:
      0 r1                   1                     -0
      1 r2       6.6613381e-16                    0.5
      2 r3      -6.6613381e-16                    0.5
      0 x0                   0                      0
```

```
      1 x1          0.66666667                    0
      2 x2          0.33333333                    0
Clp:

%%%%%%%%%%%%%%%%%%%% CLP Results %%%%%%%%%%%%%%%%%%%%
```

2.1.2 Problem

We consider a game where players X and Y simultaneously (and repeatedly) choose one of the following options: Rock, Paper, or Scissor. The following table gives the payoff from player Y to player X for each of the possible cases

		X		
		Rock	Paper	Scissor
	Rock	0	1	-1
Y	Paper	-1	0	1
	Scissor	1	-1	0

Write down a linear programming problem that corresponds to X choosing the optimal strategy under the assumption that Y discovers the strategy. Show that $\hat{y} = \hat{x} = (1/3, 1/3, 1/3)^{\mathrm{T}}$ is the optimal strategy for both X and Y.

2.2 Lagrangians

Notation: We use $\bar{\mathbf{R}}$ to denote the set of real number (non-negative real numbers) together with plus and minus infinity; i.e. $\bar{\mathbf{R}} = \mathbf{R} \cup \{+\infty, -\infty\}$.

2.2.1 Limits

Upper Bound: Given a set of numbers $U \subset \bar{\mathbf{R}}$, we say that b is an *upper bound* for U if $u \leq b$ for all $u \in U$. If s is an upper bound for U and $s \leq b$ for any other upper bound b, we say that s is the least upper bound for U which is also referred to as the *supremum* of U and denoted by

$$s = \sup\{u \mid u \in U\} = \sup_{u \in U} u$$

Lower Bound: Given a set of numbers $U \subset \bar{\mathbf{R}}$, we say that b is a *lower bound* for U if $u \geq b$ for all $u \in U$. If r is a lower bound for U and $r \geq b$ for any other lower bound b, we say that r is the greatest lower bound for U which is also referred to as the *infimum* of U and denoted by

$$r = \inf\{u \mid u \in U\} = \inf_{u \in U} u$$

2.2.2 Example

Primal Problem: The primal problem for the example in Section 2.1.1 is

$$\begin{array}{lcll} \text{maximize} & z & \text{w.r.t} & z \in \mathbf{R}\,,\ x \in \mathbf{R}^2_+ \\ \text{subject to} & x_1 + x_2 & = 1 & \\ & z - 5x_1 + 10x_2 & \leq 0 & \\ & z + 5x_1 - 10x_2 & \leq 0 & \end{array}$$

We use U to denote the feasible set for this problem; i.e.,

$$U = \left\{ (z,x) \in \mathbf{R} \times \mathbf{R}^2_+ \;\middle|\; \begin{array}{ccccccc} & & x_1 & + & x_2 & = & 1 \\ z & - & 5x_1 & + & 10x_2 & \leq & 0 \\ z & + & 5x_1 & - & 10x_2 & \leq & 0 \end{array} \right\}$$

The extended primal objective function $f : \mathbf{R} \times \mathbf{R}^2_+ \to \bar{\mathbf{R}}$ is defined by

$$f(z,x) = \begin{cases} z & \text{if } (z,x) \in U \\ -\infty & \text{otherwise} \end{cases}$$

Solving the primal problem is equivalent to

$$\text{maximize} \quad f(z,x) \quad \text{w.r.t} \quad z \in \mathbf{R}\,,\ x \in \mathbf{R}^2_+$$

Lagrangian: The Lagrangian $L : \mathbf{R} \times \mathbf{R}^2_+ \times \mathbf{R} \times \mathbf{R}^2_+ \to \mathbf{R}$ corresponding to the primal problem is

$$L(z,x,w,y) = z + w(1 - x_1 - x_2) + y_1(0 - z + 5x_1 - 10x_2) + y_2(0 - z - 5x_1 + 10x_2)$$

Note that the free dual variable w corresponds to the equality constraint and the non-negative dual variables y_1 , y_2 correspond to the inequality constraints.

Claim:

$$f(z,x) = \inf\{L(z,x,w,y) \mid (w,y) \in \mathbf{R} \times \mathbf{R}^2_+\}$$

Proof:

Case 1: Suppose that $(z,x) \in U$ and $(w,y) \in \mathbf{R} \times \mathbf{R}^2_+$. It follows that

$$\begin{array}{rcl} w(1 - x_1 - x_2) & = & 0 \\ y_1(0 - z + 5x_1 - 10x_2) & \geq & 0 \\ y_2(0 - z - 5x_1 + 10x_2) & \geq & 0 \\ L(z,x,w,y) & \geq & z \end{array}$$

We therefore know that

$$\inf\{L(z,x,w,y) \mid (w,y) \in \mathbf{R} \times \mathbf{R}^2_+\} \geq z$$

But substituting in zero for w and y we conclude that

$$\inf\{L(z,x,w,y) \mid (w,y) \in \mathbf{R} \times \mathbf{R}^2_+\} \leq z$$

Thus the claim holds for this case.

Case 2: Suppose that $1 - x_1 - x_2 \neq 0$. It follows that

$$\begin{aligned} L(z,x,w,0) &= z + w(1 - x_1 - x_2) \\ \inf\{L(z,x,w,0) \mid w \in \mathbf{R}\} &= -\infty \\ \inf\{L(z,x,w,y) \mid (w,y) \in \mathbf{R} \times \mathbf{R}^2_+\} &= -\infty \end{aligned}$$

Thus the claim holds for this case.

Case 3: Suppose that $0 - z + 5x_1 - 10x_2 < 0$. It follows that

$$\begin{aligned} L[z,x,0,(y_1,0)] &= z + y_1(0 - z + 5x_1 - 10x_2) \\ \inf\{L[z,x,0,(y_1,0)] \mid y_1 \in \mathbf{R}_+\} &= -\infty \\ \inf\{L(z,x,w,y) \mid (w,y) \in \mathbf{R} \times \mathbf{R}^2_+\} &= -\infty \end{aligned}$$

Thus the claim holds for this case

Conclusion: The proof of case $0 - z - 5x_1 + 10x_2 < 0$ is very similar to Case 3. Since that is the final case, this completes the proof of this claim.

Dual Problem: The extended dual objective $g : \mathbf{R} \times \mathbf{R}^2_+ \to \bar{\mathbf{R}}$ is defined by

$$g(w,y) = \sup\{L(z,x,w,y) \mid (z,x) \in \mathbf{R} \times \mathbf{R}^2_+\}$$

Regrouping the terms defining $L(z,x,w,y)$ we have

$$L(z,x,w,y) = w + z(1 - y_1 - y_2) + x_1(0 - w + 5y_1 - 5y_2) + x_2(0 - w - 10y_1 + 10y_2)$$

We define the set V by

$$V = \left\{ (w,y) \in \mathbf{R} \times \mathbf{R}^2_+ \;\middle|\; \begin{array}{ccccccc} & & y_1 & + & y_2 & = & 1 \\ w & - & 5y_1 & + & 5y_2 & \geq & 0 \\ w & + & 10y_1 & - & 10y_2 & \geq & 0 \end{array} \right\}$$

Using reasoning similar to that in the proof of the claim above, we obtain

$$\begin{aligned} g(w,y) &= \sup\{L(z,x,w,y) \mid (z,x) \in \mathbf{R} \times \mathbf{R}^2_+\} \\ &= \begin{cases} w & \text{if } (w,y) \in V \\ +\infty & \text{otherwise} \end{cases} \end{aligned}$$

The dual problem is defined as

$$\text{minimize} \quad g(w,y) \quad \text{w.r.t} \quad w \in \mathbf{R}\,,\ y \in \mathbf{R}^2_+$$

This problem is equivalent to

$$\begin{array}{rcl} \text{minimize} & w & \text{w.r.t} \quad w \in \mathbf{R}\,,\ y \in \mathbf{R}^2_+ \\ \text{subject to} & y_1 + y_2 & = 1 \\ & w - 5y_1 + 5y_2 & \geq 0 \\ & w + 10y_1 - 10y_2 & \geq 0 \end{array}$$

2.2.3 General Duality

Primal Problem: Using the notation on page 19, the primal problem is

$$\text{maximize} \quad f(x_\pm, x_+) \quad \text{w.r.t} \quad x_\pm \in \mathbf{R}^{n\pm} \ , \ x_+ \in \mathbf{R}_+^{n+}$$

where the extended primal objective $f : \mathbf{R}^{n\pm} \times \mathbf{R}_+^{n+} \to \bar{\mathbf{R}}$ is defined by

$$f(x_\pm, x_+) = \begin{cases} c_\pm^{\mathrm{T}} x_\pm \ + \ c_+^{\mathrm{T}} x_+ & \text{if} \left[\begin{array}{lll} A_{=,\pm} x_\pm \ + \ A_{=,+} x_+ & = b_= \\ A_{\le,\pm} x_\pm \ + \ A_{\le,+} x_+ & \le b_\le \end{array} \right] \\ -\infty & \text{otherwise} \end{cases}$$

Lagrangian: The Lagrangian $L : \mathbf{R}^{n\pm} \times \mathbf{R}_+^{n+} \times \mathbf{R}^{m=} \times \mathbf{R}_+^{m\le} \to \mathbf{R}$ corresponding to the primal and dual problems is

$$\begin{array}{rcl} L(x_\pm, x_+, y_=, y_\le) & = & c_\pm^{\mathrm{T}} x_\pm \ + \ c_+^{\mathrm{T}} x_+ \\ & + & y_=^{\mathrm{T}}(b_= - A_{=,\pm} x_\pm \ - \ A_{=,+} x_+) \\ & + & y_\le^{\mathrm{T}}(b_\le - A_{\le,\pm} x_\pm \ - \ A_{\le,+} x_+) \\ & = & b_=^{\mathrm{T}} y_= \ + \ b_\le^{\mathrm{T}} y_\le \\ & + & x_\pm^{\mathrm{T}}(c_\pm \ - \ A_{=,\pm}^{\mathrm{T}} y_= \ - \ A_{\le,\pm}^{\mathrm{T}} y_\le) \\ & + & x_+^{\mathrm{T}}(c_+ \ - \ A_{=,+}^{\mathrm{T}} y_= \ - \ A_{\le,+}^{\mathrm{T}} y_\le) \end{array}$$

Dual Problem: The dual problem is

$$\text{minimize} \quad g(y_=, y_\le) \quad \text{w.r.t} \quad y_= \in \mathbf{R}^{m=} \ , \ y_\le \in \mathbf{R}_+^{m\le}$$

where the extended dual objective $g : \mathbf{R}^{m=} \times \mathbf{R}^{m\le} \to \bar{\mathbf{R}}$ is defined by

$$g(y_=, y_\le) = \begin{cases} b_=^{\mathrm{T}} y_= \ + \ b_\le^{\mathrm{T}} y_\le & \text{if} \left[\begin{array}{lll} A_{=,\pm}^{\mathrm{T}} y_= \ + \ A_{\le,\pm}^{\mathrm{T}} y_\le & = c_\pm \\ A_{=,+}^{\mathrm{T}} y_= \ + \ A_{\le,+}^{\mathrm{T}} y_\le & \ge c_+ \end{array} \right] \\ +\infty & \text{otherwise} \end{cases}$$

Weak Duality Theorem: For all $\hat{x}_\pm \in \mathbf{R}^{n\pm}$, $\hat{x}_+ \in \mathbf{R}^{n+}$, $\hat{y}_= \in \mathbf{R}^{m=}$, $\hat{y}_\le \in \mathbf{R}^{m\le}$,

$$\begin{array}{rcl} g(\hat{y}_=, \hat{y}_\le) & = & \sup\{L(x_\pm, x_+, \hat{y}_=, \hat{y}_\le) \mid (x_\pm, x_+) \in \mathbf{R}^{n\pm} \times \mathbf{R}^{n+}\} \\ f(\hat{x}_\pm, \hat{x}_+) & = & \inf\{L(\hat{x}_\pm, \hat{x}_+, y_=, y_\le) \mid (y_=, y_\le) \in \mathbf{R}^{m=} \times \mathbf{R}^{m\le}\} \\ g(\hat{y}_=, \hat{y}_\le) & \ge & f(\hat{x}_\pm, \hat{x}_+) \end{array}$$

Proof: The proof of the first two equalities is very similar to the proof of the claim on page 27. The inequality follows from the first two equalities because

$$\begin{array}{rcl} g(\hat{y}_=, \hat{y}_\le) & = & \sup\{L(x_\pm, x_+, \hat{y}_=, \hat{y}_\le) \mid (x_\pm, x_+) \in \mathbf{R}^{n\pm} \times \mathbf{R}^{n+}\} \\ & \ge & L(\hat{x}_\pm, \hat{x}_+, \hat{y}_=, \hat{y}_\le) \\ & \ge & \inf\{L(\hat{x}_\pm, \hat{x}_+, y_=, y_\le) \mid (y_=, y_\le) \in \mathbf{R}^{m=} \times \mathbf{R}^{m\le}\} \\ & = & f(\hat{x}_\pm, \hat{x}_+) \end{array}$$

Remark: It follows from the theorem above that if $g(\hat{y}_=, \hat{y}_\le) = f(\hat{x}_\pm, \hat{x}_+)$ then $(\hat{x}_\pm, \hat{x}_+)$ solves the primal problem and $(\hat{y}_=, \hat{y}_\le)$ solves the dual problem.

2.2.4 Problem

Define $X = \mathbf{R} \times \mathbf{R}_+$ and $Y = \mathbf{R}_+ \times \mathbf{R}_+ \times \mathbf{R}$ We are given the following Lagrangian $L \,:\, X \times Y \to \mathbf{R}$:

$$L(x_1, x_2, y_1, y_2, y_3) = 3x_1 + 2x_2 \;+\; y_1(4 - x_1) \;+\; y_2(-8 - 5x_1 - 3x_2) \;+\; y_3(23 - 4x_1 - 2x_2)$$

We define the extended primal objective $f \,:\, X \to \bar{\mathbf{R}}$ and dual objective $g \,:\, Y \to \bar{\mathbf{R}}$ by

$$f(x) = \inf\{L(x, y) \mid \text{w.r.t } y \in Y\} \qquad , \qquad g(y) = \sup\{L(x, y) \mid \text{w.r.t } x \in X\}$$

(1) Write down a description of the set (call it U) where $f(x)$ is equal to $-\infty$. (2) Write down the value of $f(x)$ for $x \in X$ and $x \notin U$ (in terms of x_1 and x_2 and with any inf). (3) Write down a description of the set (call it V) where $g(y)$ is equal to $+\infty$. (4) Write down the value of $g(y)$ for $y \in Y$ and $y \notin V$ (in terms of y_1, y_2 and y_3 and with any sup).

Chapter 3

Geometry

3.1 Geometry Example

Problem: Consider the plastic cup factory optimization problem from the Section 1 of Professor Burke's notes . Note that the constraint $x_1/15 + x_2/15 \leq 8$ has been converted to the equivalent form $x_1 + x_2 \leq 120$.

$$\begin{array}{rlll} \text{maximize} & 25x_1 + 20x_2 & \text{w.r.t} & x \in \mathbf{R}^2_+ \\ \text{subject to} & 20x_1 + 12x_2 & \leq 1800 & \\ & x_1 + x_2 & \leq 120 & \end{array}$$

Output: The following pivots correspond to the problem above Pivots for Plastic Cup Problem

```
        x1         x2          s1          s2          b
        20         12           1           0       1800
         1          1           0           1        120
        25         20           0           0          0
(r, c) = (1,1)
         1        0.6        0.05           0         90
         0        0.4       -0.05           1         30
         0          5       -1.25           0      -2250
(r, c) = (2,2)
         1          0       0.125        -1.5         45
         0          1      -0.125         2.5         75
         0          0      -0.625       -12.5      -2625
```

Simplex Method:

1. The initial tableau in the output above corresponds to the basic solution $x_1 = 0$, $x_2 = 0$. This corresponds to the intersection of the line labeled A with the line labeled B in the figure on page 32. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

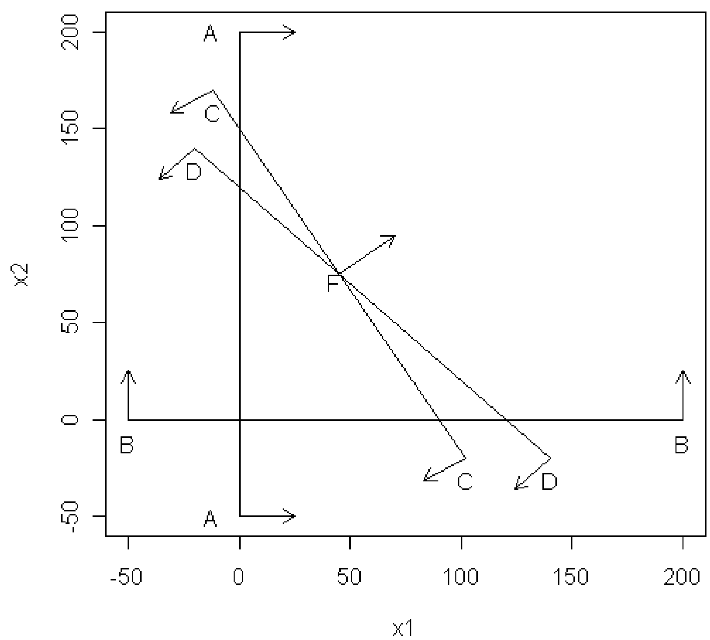

Figure 3.1: A labels $x_2 \geq 0$, B labels $x_1 \geq 0$, C labels the constraint $20x_1 + 12x_2 \leq 1800$, D labels $x_1 + x_2 \leq 120$, F labels the objective direction $(25, 20)$.

2. The second tableau in the output above corresponds to the basic solution $x_1 = 90$, $x_2 = 0$. This corresponds to the intersection of the line labeled B with the line labeled C in the figure. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 20 & 12 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1800 \\ 0 \end{pmatrix}$$

3. The third tableau in the output above corresponds to the basic solution $x_1 = 45$, $x_2 = 75$. This corresponds to the intersection of the line labeled C with the line labeled D in the figure. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 20 & 12 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1800 \\ 120 \end{pmatrix}$$

3.2 The Simplex Vertex Theorem

Vertex: We are given $A \in \mathbf{R}^{m \times n}$, $b \in \mathbf{R}^m$, $c \in \mathbf{R}^n$ and our problem is

$$\begin{array}{rll} \text{minimize} & c^{\mathrm{T}} x & \text{w.r.t.} \quad x \in \mathbf{R}^n_+ \\ \text{subject to} & Ax & \leq b \end{array}$$

A point $\hat{x}$ is called a vertex of $\{x \in \mathbf{R}^n_+ \mid Ax \leq b\}$ if it is in $\{x \in \mathbf{R}^n_+ \mid Ax \leq b\}$ and there are sets of indices $I \subset \{1, \ldots, m\}$ and $J \subset \{1, \ldots, n\}$ such that $|I| + |J| = n$ and $\hat{x}$ is the unique solution of

$$\begin{aligned} \sum_j A_{i,j} \hat{x}_j &= b_i \text{ for all } i \in I \\ \hat{x}_j &= 0 \text{ for all } j \in J \end{aligned}$$

where $|I|$ and $|J|$ are the number of indices in the sets I and J respectively.

Theorem: The simplex method proceeds from one vertex to another until it determines that the problem is unbounded or it finds an optimal solution that is also a vertex.

Proof: As a consequence of the results for Bland's selection method, we only need show that each basic feasible solution is a vertex for the feasible region. The simplex method works with the following reformulated version of the problem above:

$$\begin{array}{rll} \text{minimize} & c^{\mathrm{T}} x & \text{w.r.t.} \quad x \in \mathbf{R}^n_+ \ , \ s \in \mathbf{R}^m_+ \\ \text{subject to} & Ax + s & = b \end{array}$$

Each pivot of the simplex algorithm corresponds to multiplication of the tableau on the left by a non-singular $(m+1) \times (m+1)$ matrix (see Simplex Algorithm on a Tableau in Matrix Notation in Professor Burke's notes on the Simplex Algorithm). We use the notation $T \in \mathbf{R}^{(m+1)\times(n+m+1)}$ the initial tableau

$$T = \begin{pmatrix} A & E & b \\ c^{\mathrm{T}} & 0_{1 \times m} & 0 \end{pmatrix}$$

where E is the $m \times m$ identity matrix. Using a similar notation for the current tableau $\hat{T}$

$$\hat{T} = \begin{pmatrix} \hat{A} & \hat{R}^{-1} & \hat{b} \\ \hat{c}^{\mathrm{T}} & -\hat{y}^{T} & -\hat{z} \end{pmatrix}$$

As noted in Professor Burke's notes (where $\hat{R}^{-1}$ is denoted by $\hat{R}$), the matrix $\hat{R}^{-1} \in \mathbf{R}^{m \times m}$ is non-singular and

$$\hat{T} = \begin{pmatrix} \hat{R}^{-1} & 0_{m \times 1} \\ -y^{\mathrm{T}} & \beta \end{pmatrix} T \tag{3.1}$$

where $\beta \in \mathbf{R}$. Multiplying both sides by the inverse of the matrix factor between tableaus we have

$$\begin{pmatrix} \hat{R} & 0_{m \times 1} \\ \beta^{-1} y^{\mathrm{T}} \hat{R} & \beta^{-1} \end{pmatrix} \hat{T} = T \tag{3.2}$$

In the tableau $\hat{T}$, let J be the indices of the x components that are nonbasic, let I be the indices of s that are nonbasic, and let $(\hat{x}, \hat{s})$ denote the basic solution. This solution also satisfies the equation in the initial tableau T. For each $i \in I$ we have $\hat{s}_i = 0$ and hence

$$\begin{aligned} \sum_{j=1}^{n} A_{i,j} \hat{x}_j &= b_i \text{ for } i \in I \\ \hat{x}_j &= 0 \text{ for } j \in J \end{aligned} \tag{3.3}$$

So it suffices to show that there is only one solution to these equations to complete the proof.

We use the one to one mapping

$$B : \{1, \cdots, m\} \rightarrow \{1, \cdots, n+m\}$$

to represent the current basis; i.e. each row index in the tableau $\hat{T}$ is mapped to the index of the corresponding variable where $\{1, \ldots, n\}$ correspond to x and $\{n+1, \ldots, n+m\}$ correspond to s. We use $K(1), \ldots, K(|K|)$ to denote the set of $i \in \{1, \ldots, m\}$ such that $B(i) \leq n$. Note that each component of x that is basic can be written as $x_{B(i)}$ where $i \in K$. It follows from equation (3.2) that

$$\begin{aligned} i \in K &\quad\Rightarrow\quad \hat{R}_{k,i} = A_{k,B(i)} \text{ for } k = 1, \ldots, m \\ i \notin K &\quad\Rightarrow\quad \hat{R}_{k,i} = E_{k,B(i)-n} \text{ for } k = 1, \ldots, m \end{aligned}$$

(remember that E is the $m \times m$ identity matrix). Note that each component of s that is basic can be written as $s_{B(i)-n}$ where $i \notin K$. The matrix $\hat{R}$ is nonsingular and hence its determinant is nonzero. We use $I(1), \ldots, I(|I|)$ to index the elements I; i.e., the nonbasic components of s. Note that $|I| = |K|$. The matrix R has non-zero determinant. Using expansion by minors for the columns of $\hat{R}$ corresponding to $E_{k,B(i)-n}$ above, we conclude that the following matrix has nonzero determinant

$$A_{I,B} = \begin{pmatrix} A_{I(1),B(K(1))}, \ldots, A_{I(1),B(K(|K|)} \\ A_{I(|K|),B(K(1))}, \ldots, A_{I(|K|),B(K(|K|)} \end{pmatrix}$$

But this is just a reordering of the columns in the matrix corresponding to equation (3.3) with the nonbasic columns of x removed; i.e., the columns in J. Hence it is not singular and this completes the proof.

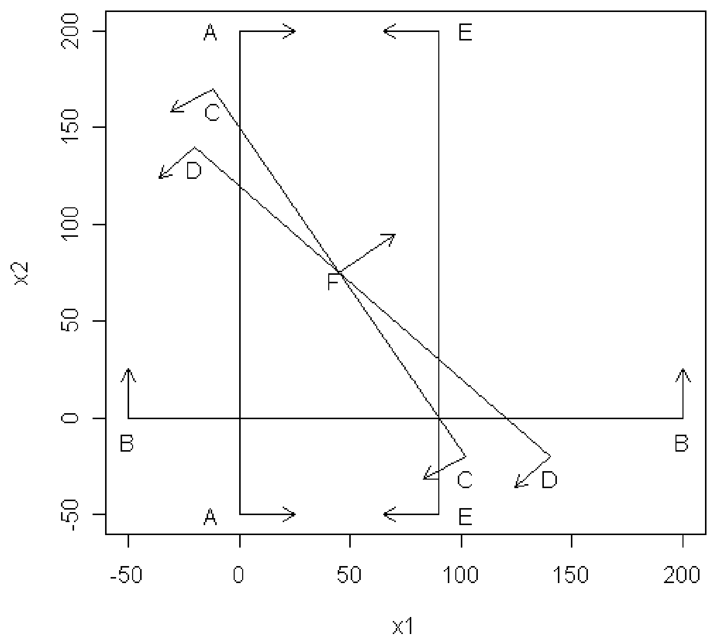

Figure 3.2: A labels $x_2 \geq 0$, B labels $x_1 \geq 0$, C labels the constraint $20x_1 + 12x_2 \leq 1800$, D labels $x_1 + x_2 \leq 120$, E labels $x_1 \leq 90$, F labels the objective direction $(25, 20)$.

3.3 Degeneracy Geometry

$$\begin{array}{rlll} \text{maximize} & 25x_1 + 20x_2 & \text{w.r.t} & x \in \mathbf{R}^2_+ \\ \text{subject to} & 20x_1 + 12x_2 & \leq 1800 & \\ & x_1 + x_2 & \leq 120 & \\ & x_1 & \leq 90 & \end{array}$$

Pivots: The following pivots correspond to the problem above Pivots for Degenerate Problem

```
        x1        x2        s1        s2        s3         b
        20        12         1         0         0      1800
         1         1         0         1         0       120
         1         0         0         0         1        90
        25        20         0         0         0         0
(r, c) = (3,1)
         0        12         1         0       -20         0
         0         1         0         1        -1        30
         1         0         0         0         1        90
         0        20         0         0       -25     -2250
```

```
(r, c) = (1,2)
         0         1   0.08333         0    -1.667         0
         0         0  -0.08333         1    0.6667        30
         1         0         0         0         1        90
         0         0    -1.667         0     8.333     -2250
(r, c) = (2,5)
         0         1    -0.125       2.5         0        75
         0         0    -0.125       1.5         1        45
         1         0     0.125      -1.5         0        45
         0         0    -0.625     -12.5         0     -2625
```

Simplex Method:

1. The initial tableau in the output above corresponds to the basic solution $x_1 = 0$, $x_2 = 0$. This corresponds to the intersection of the line labeled A with the line labeled B in the figure on page 35. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

2. The second tableau in the output above corresponds to the basic solution $x_1 = 90$, $x_2 = 0$. This corresponds to the intersection of the line labeled B with the line labeled F in the figure. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 90 \\ 0 \end{pmatrix}$$

3. The third tableau in the output above corresponds to the basic solution $x_1 = 90$, $x_2 = 0$. This time it corresponds to the intersection of the line labeled F with the line labeled C in the figure. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 1 & 0 \\ 20 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 90 \\ 180 \end{pmatrix}$$

4. The fourth tableau in the output above corresponds to the basic solution $x_1 = 45$, $x_2 = 75$. This corresponds to the intersection of the line labeled C with the line labeled D in the figure. These two lines are represented by the rows in the equation

$$\begin{pmatrix} 1 & 1 \\ 20 & 12 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 120 \\ 1800 \end{pmatrix}$$

3.4 Cycling Example

The following pivots demonstrate cycling:
Output for Cycling Demonstration

3.5 Problems

1. We are given the optimization problem

$$\begin{array}{rll} \text{maximize} & x_1 + x_2 & \text{w.r.t} \quad x \in \mathbf{R}_+^2 \\ \text{subject to} & x_1 + 2x_2 & \leq 2 \end{array}$$

   (a) Write down the two linear equations corresponding to each vertex of the feasible region corresponding to the problem above.

   (b) Write down the objective value corresponding to each of the vertices and which of the vertices correspond to the optimal value for the objective function.

2. We are given the optimization problem

$$\begin{array}{rll} \text{maximize} & x_1 + x_2/2 + x_3 & \text{w.r.t} \quad x \in \mathbf{R}_+^3 \\ \text{subject to} & 2x_1 + x_2 + 2x_3 & \leq 2 \end{array}$$

   (a) Write down the three linear equations corresponding to each vertex of the feasible region corresponding to the problem above.

   (b) Write down the objective value corresponding to each of the vertices and which of the vertices correspond to the optimal value for the objective function.

Index