CSCI 628
Homework 1

Megan Rose Bryant
Department of Mathematics
William and Mary

September 10, 2014

Problem 1

A steel company must decide how to allocate next weeks time on a rolling mill, which is a machine that takes unfinished slabs of steel as input and can produce either of two semi-finished products: bands and coils.

The mills two products come off the rolling line at different rates:

Bands 200 tons/hr
Coils 150 tons/hr
They also produce different profits:
Bands $75/ton
Coils $100/ton

Based on currently booked orders, the following upper bounds are placed on the amount of each product to produce:

Bands 6000 tons
Coils 4000 tons

Finally, the company has 9000 tons of unfinished steel to produce the bands and coils, so the total amount produced is at most 9000 tons.

Given that there are 40 hours of production time available this week, the problem is to decide how many tons of bands and how many tons of coils should be produced to yield the highest profit. Formulate this problem as a linear programming problem and solve it using the graphical method.

$$
\begin{array}{ll}
\text{maximize} & 75x_1 + 100x_2 \\
\text{subject to} & \frac{1}{200}x_1 + \frac{1}{150}x_2 \leq 40 \\
& x_1 \leq 6000 \\
& x_2 \leq 4000 \\
& x_1 + x_2 \leq 9000 \\
& x_1 \geq 0 \\
& x_2 \geq 0
\end{array}
$$

Problem 2

For each of the following linear programs, decide graphically whether the problem is infeasible, unbounded, or has an optimal solution. Then explain algebraically why your answer is correct.

(i)

$$\begin{array}{ll} \text{maximize} & -x_2 \\ \text{subject to} & x_1 + x_2 \geq 1 \\ & x_1 - x_2 \leq -3 \\ & x_2 \leq 1 \end{array}$$

(ii)

$$\begin{array}{ll} \text{maximize} & -x_2 \\ \text{subject to} & x_1 + x_2 \geq 1 \\ & x_1 - x_2 \leq -3 \\ & x_2 \geq 1 \end{array}$$

(i)

$$\begin{array}{ll} \text{maximize} & x_2 \\ \text{subject to} & x_1 + x_2 \geq 1 \\ & x_1 - x_2 \leq -3 \\ & x_2 \geq 1 \end{array}$$

Problem 3

Convert the following LP into standard inequality form:

$$\begin{array}{ll} \text{minimize} & 3x_1 + 2x_2 \\ \text{subject to} & 4x_1 - 2x_2 = 4 \\ & x_1 + x_2 \geq 3 \\ & 2x_1 - x_2 \leq -1 \\ & x_1 \geq 0 \end{array}$$

(i) Standard form:

$$\begin{array}{ll} \text{maximize} & -3x_1 - 2x_2^+ + 2x_2^- \\ \text{subject to} & -4x_1 + 2x_2^+ - 2x_2^- \leq 4 \\ & 4x_1 - 2x_2^+ + 2x_2^- \leq 4 \\ & -x_1 - x_2^+ + 2x_2^- \leq -3 \\ & 2x_1 - x_2^+ + 2x_2^- \leq -1 \\ & x_1 \geq 0 \\ & x_2^+ \geq 0 \\ & x_2^- \geq 0 \end{array}$$

Problem 4

Base Scenario

Union rules state that each full-time employee must work 5 consecutive days and then receive two days off.

Day	# of full-time employees required
1 = Monday	17
2 = Tuesday	13
3 = Wednesday	15
4 = Thursday	19
5 = Friday	14
6 = Saturday	16
7 = Sunday	11

Formulate an IP to minimize the number of fulltime employees who must be hired, implement the IP in AMPL, and solve it to find the solution.

Some pointers:
1.) The right decision variables are key for this problem. They should allow you to figure out who is working on a given day, and how many people have their day off...
2.)Start by completely writing out all constraints of your LP on paper (writing out your summations).

Minimize $\sum_{j=1}^{7} x_j$

S.T. $\sum_{k=1}^{7} x_k - x_{j-1,\ \mathrm{mod}\ 7} - x_{j-2,\ \mathrm{mod}\ 7} >= d_j$, for $j = 1 \cdots 7$

$x_j \geq 0$

AMPL Model File:

```
set DAYS = {"MON", "TUES", "WED", "THURS", "FRI", "SAT", "SUN"} circular;

param d { DAYS };

var x { DAYS } >=0, integer;
#number of people starting work on a certain day.

minimize NUMBER_OF_EMPLOYEES: sum {j in DAYS} x[j];

subject to DAYS_OF_WEEK{ j in DAYS}:
```

```
sum {k in DAYS} x[k] - x [next(j,DAYS,1)] - x [next(j,DAYS,2)] >= d[j];
#meeting the minimum number of employees for each day
```

AMPL Data File:

```
param d:=
"MON" 17
"TUES" 13
"WED" 15
"THURS" 19
"FRI" 14
"SAT" 16
"SUN" 11;
```

AMPL Output

```
ampl: option solver gurobi;
ampl: model lab1mod.mod;
ampl: data lab1dat.dat;
ampl: solve;
Gurobi 5.6.3: optimal solution; objective 23
8 simplex iterations

ampl: display x;
x [*] :=
  MON  2
 TUES  4
  WED  2
THURS  7
  FRI  0
  SAT  3
  SUN  5
;
```

Variation 1: Forced Overtime

The post office can now force employees to work a 6th day each week at a higher pay rate. The pay is \$100 *for a regular day and* \$130 *for the overtime day. Adapt your IP so that it minimizes the cost of meeting weekly labor requirements.*

Minimize $\$500 \sum_{j=1}^{7} x_j + \$630 \sum_{j=1}^{7} y_j$
S.T.
$\sum_{k=1}^{7} x_k - x_{(j-1) \mod 7} - x_{(j-2) \mod 7} + y_{(j+1) \mod 7} + y_{(j+2) \mod 7} >= d_j, j = 1 \cdots 7$
$x_j, y_j \geq 0$

AMPL Model File:

```
set DAYS = {"MON", "TUES", "WED", "THURS", "FRI", "SAT", "SUN"} circular;

param d { DAYS };

var x { DAYS } >=0, integer;
#number of people starting work on a certain day, 5 day work week.
var y { DAYS } >=0, integer;
#number of people starting work on a certain day, 6 day work week.

minimize PAYROLL: sum {j in DAYS} x[j] + sum {j in DAYS} y[j];

subject to DAYS_OF_WEEK{j in DAYS}:
sum {k in DAYS} x[k]-x[next(j,DAYS,1)]-x[next(j,DAYS,2)]+y[next(j,DAYS,1)]+y[next
#meeting the minimum number of employees for each day
```

AMPL Data File:

```
param d:=
"MON" 17
"TUES" 13
"WED" 15
```

```
"THURS" 19
"FRI" 14
"SAT" 16
"SUN" 11;
```

AMPL Output

```
ampl: reset;
ampl: model lab1var1mod.mod;
ampl: data lab1vardat.dat;
ampl: data lab1var1dat.dat;
ampl: option solver gurobi;
ampl: solve;
Gurobi 5.6.3: optimal solution; objective 23
7 simplex iterations
ampl: display x;
x [*] :=
  MON  2
 TUES  4
  WED  2
THURS  7
  FRI  0
  SAT  3
  SUN  5
;

ampl: display y;
y [*] :=
  MON  0
 TUES  0
  WED  0
THURS  0
  FRI  0
  SAT  0
  SUN  0
;
```

Variation 2: Maximizing Weekends

The post office now has 25 full-time employees and cannot hire or fire. Adapt your original IP to schedule the employees in order to maximize the number of weekend days off received by the employees.

Minimize $\sum_{j=6}^{7} x_j$

S.T.

$\sum_{k=1}^{7} x_k - x_{j-1,\ \mathrm{mod}\ 7} - x_{j-2,\ \mathrm{mod}\ 7} >= d_j$, for $j = 1 \cdots 7$

$\sum_{j=1}^{7} x_j = 25$

$x_j \geq 0$

AMPL Model File:

```
set DAYS = {"MON", "TUES", "WED", "THURS", "FRI", "SAT", "SUN"} circular;

param d { DAYS };

var x { DAYS } >=0, integer; #number of people starting work on a certain day.

minimize NUMBER_OF_EMPLOYEES_WORKING_WEEKENDS: x["SAT"] + x ["SUN"];

subject to DAYS_OF_WEEK{j in DAYS}:
sum {k in DAYS} x[k] - x [ next (j, DAYS, 1) ] - x [ next (j, DAYS, 2) ]  >= d[j];
#meeting the minimum number of employees for each day

subject to NUMBER_OF_EMPLOYEES:
sum {j in DAYS} x[j] = 25;
#total number of employees is 25
```

AMPL Data File:

```
param d:=
"MON" 17
"TUES" 13
```

```
"WED" 15
"THURS" 19
"FRI" 14
"SAT" 16
"SUN" 11;
```

AMPL Output

```
ampl: display x;
ampl: model lab1var2mod.mod;
ampl: data lab1var2dat.dat;
ampl: option solver gurobi;
ampl: solve;
Gurobi 5.6.3: optimal solution; objective 0
5 simplex iterations
ampl: display x;
x [*] :=
  MON   7
 TUES   6
  WED   2
THURS  10
  FRI   0
  SAT   0
  SUN   0
;
```