

CS635 – Problem Set #7

Due Date: March 27, 2015

Instructions for Handing In Homework

Formulate the following problems in GAMS and solve them. Please follow the instructions given in the problems closely. Submit this assignment electronically to the drop box. You should hand in a single zip file containing files with the following names: hw7-1.gms, hw7-2.gms, hw7-3.gms, hw7-3.log, hw7-4.gms This assignment is worth 200 points.

1 Filtch's Paint Company

As part of his weekly production, the caretaker of Hogwarts, Augustus Filtch must produce n batches of magical paints, always the same. Every paint batch is produced in a single production process, all in the same blender that needs to be cleaned between two batches.

The cleaning times depend of the colors and the paint types. For example, a long cleaning period is required if an oil-based paint is produced after a water-based paint, or to produce white paint after a dark color. The times are given in minutes in the following table `CLEAN` where `CLEAN(i, j)` denotes the cleaning time between batch i and batch j . The durations of blending paint batches are given in the parameter `DUR(i)` below:

```
set i /1*10/;
alias(i, j);
```

```
table clean(i, j)
  1  2  3  4  5  6  7  8  9 10
1      11 7 13 11 12 4  9  7 11
2  5      13 15 15 6  8 10  9  8
3 13 15      23 11 11 16 18  5  7
4  9 13 5      3  8 10 12 14  5
5  3  7  7  7      9 10 11 12 13
6 10 6  3  4 14      8  5 11 12
7  4  6  7  3 13 7      10  4  6
8  7  8  9  9 12 11 10      10  9
9  9 14 8  4  9  6 10 8      12
10 11 17 11 6 10 4  7  9 11
;
```

```
parameter dur(i) /1 40, 2 35, 3 45, 4 32, 5 50, 6 42, 7 44, 8 30, 9 33, 10 55 /;
```

Since Filch has other activities, he wishes to deal with this weekly production in the shortest possible time (blending and cleaning). What is the corresponding order of paint batches? The order will be applied every week, so the cleaning time between the last batch of one week and the first of the following week needs to be accounted for in the total duration of cleaning.

1.1 Problem

Write a GAMS model that will determine the order of the paint batches. We would like to print out the total time to process all batches (both blending and cleaning), as well as the order in which the batches are processed.

If `obj` is the variable representing the length of the blending and cleaning process, and $y(I)$ is a variable representing the position of batch I , then the following code should do the trick.¹

```
parameter batchlength;
batchlength = obj.L;

parameter order(i) ;
loop(j,
  order(i+1)$(abs(y.L(j) - ord(i)) < 0.5) = ord(j);
);

display batchlength;
display order;
```

2 Ahhhhhhhh, Daniel-son. You SuDoku Master.

Sudoku is a puzzle game craze that swept the world. According to a popular sudoku web site (<http://www.sudoku.com/>) “There’s no math involved.” But we will show them that we can needlessly and endlessly complicate *all* things with math, even a fun puzzle like sudoku.

In sudoku, you are given an $n \times n$ grid with some of the entries of the grid being filled in with numbers from $\{1, 2, \dots, n\}$. Typically $n = 9$, and in every sudoku puzzle \sqrt{n} is an integer. The object of the game is to fill in the grid so that every row, every column, and every $\sqrt{n} \times \sqrt{n}$ “box” contains the digits 1 through n .

I’ll give you some code to get you started. We have a set $N = \{1, \dots, 9\}$, and sets $R, C, V = N$ to stand for the Rows, Columns, and Values in the instance, respectively. There is also a set $B \subset N \times R \times V$ of Blocks, where $(n, r, c) \in B$ if row index r , columns index c is in block index n . A puzzle is specified by giving the values in each row and column (as a table).

```
set N /1*9/;
alias (N,R,C,V);

* Block number 'n' has (row,column) in it.
set B(N,R,C) /
1.(1*3).(1*3),
2.(1*3).(4*6),
3.(1*3).(7*9),
4.(4*6).(1*3),
5.(4*6).(4*6),
6.(4*6).(7*9),
7.(7*9).(1*3),
8.(7*9).(4*6),
9.(7*9).(7*9)
/;
```

table	P1(R,C)	given values								
	1	2	3	4	5	6	7	8	9	
1		6	7	8	5			3	4	
2			4							
3	3	1		2			9			
4		9			3		8			
5										
6			6		1			7		

¹We assume without loss of generality that we do the first batch (1) first in the order.

7		3		7		4	8
8					6		
9	4	7		8	5	3	2

;

2.1 Problem

Solve the sudoku instance given as table P1 above. Display your solution in a parameter `solution` as follows:

```
parameter solution(R,C);
loop(V,
solution(R,C)$(x.L(R,C,V) > 0.5) = ord(V);
);
display solution;
```

In real sudoku games, there is only *one* feasible solution to the IP formulation you built in Problem 2.1. This is, of course, a function of the “original” squares that are filled in. In this problem, we make the game more challenging. We will call my new game *Jeff-Doku*. In Jeff-Doku, we the set of main “diagonal” grid locations as

$$D = \{(1,1), (2,2), \dots, (n,n)\}$$

and we maximize the sum of the elements placed on the diagonal.

2.2 Problem

Suppose now, that you need not fix *all* of the values given in the instance, but rather you need fix only at least $K = 24$ of them. Modify your instance from Problem 2.1 to solve this more challenging version of the game that maximizes the sum of the elements on the main diagonal. Use `diagonal` as the GAMS variable for the value of the objective function, and include the following code after your call to solve the model:

```
parameter solution(R,C);
loop(V,
solution(R,C)$(x.L(R,C,V) > 0.5) = ord(V);
);
display solution;
display diagonal.L;
```

2.3 Problem

Even that problem was “too easy”. Now add the additional constraints and variables necessary to enforce the constraint that if you put 2 or more “9”s on the main diagonal in your Jeff-doku solution, then you must also put at least 3 “5”s on the diagonal. Include the same code after your call that you did after your solution to problem 2.2.

3 Nucular

In this problem, we are scheduling a set of power plants P over a set of time periods $T = \{1, 2, \dots, |T|\}$ to meet power demand d_t in each period. If a power plant $p \in P$ is turned on, it is able to produce any amount of power in the numerical range given by values $[\ell_p, u_p] \forall p \in P$. The cost (per MWh) of producing energy at plant $p \in P$ is c_p . The plants P are of two types—nuclear plants N and coal-fired plants C . ($P = N \cup C$).

- The objective is to minimize the total cost of delivering energy over the time horizon
- Power demand *must be* met in each period
- To meet environmental regulations, if 2 or more Coal generation plants are operating in period t , then at least 3 Nuclear generating plants must also be operating in period t

- In order to not cause a meltdown, if a nuclear generator $i \in N$ is initially turned on (started up) in period $t \in T$, then it must also be turned on for the subsequent 3 time periods.

3.1 Problem

Write a mixed integer linear program that will meet demand and minimum cost that achieves the objective and obeys all the problem restrictions. Please use the following code to make your instance.

```
sets
  T /t1*t24/
  P /p1*p10/
  C(P)
  N(P)
;

C(P) = yes$(mod(ord(P),2) = 0) ;
N(P) = yes$(mod(ord(P),2) = 1) ;

alias (S,T) ;

parameters
  d(T)
  cost(P)
  ell(P)
  u(P)
;

option seed = 666 ;
d(T) = uniform(100,1000) ;
cost(P) = uniform(6,8) ;
ell(P) = uniform(100,150) ;
u(P) = 2*ell(P) ;
```

4 Piecewise Linear Networks

Consider a transportation network with 3 commodities. Demand exists between certain origin-destination (OD) pairs for each commodity.

```
option seed=0; set nodes /1*500/;
parameter offset(nodes); offset(nodes) = round(uniform(2,5));

alias (i,j,nodes); set arcs(nodes,nodes);
arcs(i,j) = no; arcs(i,i+1) = yes; arcs(i,i+offset(i)) = yes;

set k /1*3/;
parameter demand(nodes,k) /
  1.1 -70, 6.1 70, 3.2 -25, 500.2 25, 4.3 -20, 8.3 20, 54.1 -70, 55.1 70
  23.2 -25, 89.2 25, 10.3 -20, 89.3 20, 20.3 -10, 450.3 10 /;
```

The cost of sending commodities over each arc can be modeled using $\log(x+1)$ where x represents the sum of the flows of each commodity over that arc. Total flow on each arc must be between 0 and a given capacity.

```
parameter capacity(i,j); capacity(i,j) = uniform(75,85);
```

These data statements can be downloaded from the class web page.

4.1 Problem

By approximating the concave function $\log(x+1)$ using a piecewise linear function, formulate and solve this problem for the data given below. Note that the breakpoints for each log function should be 0,5,10,100. Use appropriate solver options if necessary to ensure the optimal solution is found to within a 2% relative tolerance, and give lower and upper bounds on the optimal solution value. Hand in the log file for this problem.