Chemical Engineering 4G03

Tutorial 7 Practice Problems SOLUTIONS

Discrete Programming and Formulation Practice

Here for your own benefit and practice (best to do it individually)
Recommended completion: Week 08.
Grading: 0% (Practice for assignments and tests)

Problem 1 – Bread, Math, and Beyond

You are the manager of a small bakery commonly used as a study area for first year calculus called "Bread, Math and Beyond." You are considering the upcoming week in which you want to determine how many batches of your products to make based on some prior demand data. For the sake of argument, let's say you can make six products: bread (BRD), cookies (CKS), Rolls (ROL), muffins (MUF), crescents (CRE), and scones (SCO). The demographic data that you anticipate (maximum demands and profit per batch) for the coming week are reported in the table below. Moreover, the mixing time (MIX), baking time (BAK) and packing time (PAK) required for each batch is reported in the same table.

Product	Max Demand (Batch)	Profit ($/Batch)	Mixing (min/Batch)	Baking (min/Batch)	Packing (min/Batch)
BRD	20	12	25	14	6
CKS	60	16	28	26	9
ROL	25	13	16	16	7
MUF	40	19	35	10	8
CRE	75	9	16	24	4
SCO	35	20	22	21	9

There are several industrial stand mixers ovens available. You may assume that all tasks may be done **at the same time.** Based on the capacities of these units you have:

- **4800 minutes** of mixing time available per week.
- **4800 minutes** of baking time per week.
- You are the only employee working in the store and thus you have **1200 minutes of packing time** available per week.

Questions

1. Formulate this as an LP and enter it into GAMS as a starting point. We are going to introduce some binary variables as we go that will affect our solution. Run your code and **verify** that the optimal profit is $2644 per week.

   SETS
   $i \triangleq \text{products} \quad (\text{BRD}, \text{CKS}, \text{ROL}, \text{MUF}, \text{CRE}, \text{SCO})$
   $j \triangleq \text{processes} \quad (\text{MIX}, \text{BAKE}, \text{PACK})$

   PARAMETERS
   $P_i \triangleq$ profit earned from selling product i
   $D_i \triangleq$ maximum demand for product i
   $T_{i,j} \triangleq$ time required to complete process j for product i

   VARIABLES
   $\phi \triangleq$ objective function variable: maximize profit
   $x_i \triangleq$ number of batches of product i to make

$$\max_x \phi = \sum_i P_i \cdot x_i$$

s.t.

$$
\begin{aligned}
x_i &\le D_i && (\forall i)\\
\sum_i T_{i,j} \cdot x_i &\le A_j && (\forall j)\\
x_i &\ge 0 && (\forall i)
\end{aligned}
$$

The GAMS code and solution report for this LP is given below.

```
SETS
i    products    / BRD, CKS, ROL, MUF, CRE, SCO /
j    processes   / MIX, BAKE, PACK / ;


PARAMETERS
D(i)     demand (in batches)
/ BRD    20
  CKS    60
  ROL    25
  MUF    40
  CRE    75
  SCO    35 /

P(i)     profit (in $ per batch)
/ BRD    12
  CKS    16
  ROL    13
  MUF    19
  CRE    9
  SCO    20 /

A(j)     time available for process j
/ MIX    4800
  BAKE   4800
  PACK   1200 /     ;
```

```
TABLE   T(i,j)  time requirements of each process j for product i
        MIX     BAKE     PACK
BRD     25       14       6
CKS     28       26       9
ROL     16       16       7
MUF     35       10       8
CRE     16       24       4
SCO     22       21       9   ;


VARIABLES
PHI      objective: maximize profit
x(i)     number of batches of product i to make
POSITIVE VARIABLES x(i) ;

EQUATIONS
PROFIT
DEMAND
TIME   ;

PROFIT..        PHI =E= SUM(i, P(i)*x(i))    ;
DEMAND(i)..     x(i) =L= D(i)                ;
TIME(j)..       SUM(i,T(i,j)*x(i)) =L= A(j) ;

MODEL BAKERY /ALL/;
OPTION LP = CPLEX;

SOLVE BAKERY MAXIMIZING PHI USING LP;
```

```
Optimal solution found
Objective:         2644.285714

                       LOWER          LEVEL          UPPER         MARGINAL
---- EQU PROFIT          .              .              .            1.0000

---- EQU DEMAND
           LOWER          LEVEL          UPPER         MARGINAL
BRD        -INF          20.0000        20.0000         0.8571
CKS        -INF            .            60.0000          .
ROL        -INF          20.7143        25.0000          .
MUF        -INF          40.0000        40.0000         4.1429
CRE        -INF          75.0000        75.0000         1.5714
SCO        -INF          35.0000        35.0000         3.2857

---- EQU TIME
            LOWER          LEVEL          UPPER         MARGINAL
MIX         -INF        4201.4286      4800.0000          .
BAKE        -INF        3546.4286      4800.0000          .
PACK        -INF        1200.0000      1200.0000         1.8571

                        LOWER          LEVEL          UPPER         MARGINAL
---- VAR PHI            -INF         2644.2857        +INF            .
  PHI  objective: maximize profit


---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .           20.0000        +INF            .
CKS          .             .            +INF          -0.7143
ROL          .           20.7143        +INF            .
MUF          .           40.0000        +INF            .
```

```
CRE           .          75.0000       +INF          .
SCO           .          35.0000       +INF          .
```

2. OK, let's start to add some yes/no decisions in here. First, consider the case where, to avoid complications and to limit your orders, you are limited to producing only 4 different products per week. Add the constraint and variables that will make this happen. Add the binary variables and constraints to your GAMS code by declaring the binary as `BINARY VARIABLE`. You must change the solve statement to solve using the `MIP` solver (NOT the `LP` solver). The MIP default is CPLEX, so you will not need to change that option. Before running your code, ask yourself: is the profit *guaranteed* to be worse, better, or the same? Verify your result by running the code.

   Add the binary variable $y_i \in \{0,1\}$ to represent whether or not product i is made. The new problem formulation is therefore

$$
\begin{aligned}
\max_{x} \phi &= \sum_i P_i \cdot x_i \\
&\text{s.t.} \\
x_i &\le D_i \cdot y_i && (\forall i) \\
\sum_i T_{i,j} \cdot x_i &\le A_j && (\forall j) \\
\sum_i y_i &\le 4 \\
x_i &\ge 0 && (\forall i) \\
y_i &\in \{0,1\} && (\forall i)
\end{aligned}
$$

   Additions and changes to the GAMS code for the base-case LP in Part 1 are given in red. All previously defines sets and parameters remain the same.

```
SCALARS
MAXPROD   maximum number of products the bakery can make   / 4 / ;


VARIABLES
PHI      objective: maximize profit
x(i)     number of batches of product i to make
y(i)     whether or not product i is made ;
POSITIVE VARIABLES x(i) ;
BINARY VARIABLES   y(i) ;


EQUATIONS
PROFIT
DEMAND
TIME
PRODLIM ;

PROFIT..        PHI =E= SUM(i, P(i)*x(i))    ;
DEMAND(i)..     x(i) =L= D(i)*y(i)           ;
TIME(j)..       SUM(i,T(i,j)*x(i)) =L= A(j)  ;
PRODLIM..       SUM(i,y(i)) =L= MAXPROD      ;

MODEL BAKERY /ALL/;
OPTION MIP = CPLEX;

SOLVE BAKERY MAXIMIZING PHI USING MIP;
```

```
Proven optimal solution
MIP Solution:          2606.111111     (4 iterations, 0 nodes)
Final Solve:           2606.111111     (1 iterations)

---- EQU DEMAND
           LOWER          LEVEL          UPPER         MARGINAL
BRD        -INF            .              .              1.3333
CKS        -INF          -30.5556         .               .
ROL        -INF            .              .              0.5556
MUF        -INF            .              .              4.7778
CRE        -INF            .              .              1.8889
SCO        -INF            .              .              4.0000

---- EQU TIME
            LOWER          LEVEL          UPPER         MARGINAL
MIX         -INF         4194.4444      4800.0000         .
BAKE        -INF         3700.5556      4800.0000         .
PACK        -INF         1200.0000      1200.0000        1.7778

                            LOWER          LEVEL          UPPER         MARGINAL
---- EQU PRODLIM            -INF           4.0000         4.0000          .

                            LOWER          LEVEL          UPPER         MARGINAL
---- VAR PHI                -INF         2606.1111        +INF            .
         PHI  objective: maximize profit


---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .            +INF             .
CKS          .            29.4444        +INF             .
ROL          .              .            +INF             .
MUF          .            40.0000        +INF             .
CRE          .            75.0000        +INF             .
SCO          .            35.0000        +INF             .


---- VAR y  whether or not product i is made
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .             1.0000        26.6667
CKS          .             1.0000         1.0000          EPS
ROL          .              .             1.0000        13.8889
MUF          .             1.0000         1.0000       191.1111
CRE          .             1.0000         1.0000       141.6667
SCO          .             1.0000         1.0000       140.0000
```

3. Repeat the above but do it for 3 maximum products and 2 maximum products. Do the active constraints change significantly as you reduce the number of products you are making? What about the combination of products you are choosing?
The only thing that changes in the GAMS codes is the value of parameter `MAXPROD`.

GAMS solution for **3** maximum products:

```
Proven optimal solution
MIP Solution:          2420.000000     (4 iterations, 0 nodes)
Final Solve:           2420.000000     (0 iterations)


---- EQU DEMAND
           LOWER          LEVEL          UPPER         MARGINAL
BRD        -INF            .              .             12.0000
```

```
CKS          -INF            .              .             16.0000
ROL          -INF            .              .             13.0000
MUF          -INF            .              .             19.0000
CRE          -INF            .              .              9.0000
SCO          -INF            .              .             20.0000
---- EQU TIME
               LOWER          LEVEL          UPPER         MARGINAL
MIX             -INF        3850.0000      4800.0000          .
BAKE            -INF        2695.0000      4800.0000          .
PACK            -INF        1175.0000      1200.0000          .

                              LOWER          LEVEL          UPPER         MARGINAL
---- EQU PRODLIM              -INF           3.0000         3.0000           .


                              LOWER          LEVEL          UPPER         MARGINAL
---- VAR PHI                  -INF        2420.0000         +INF            .
         PHI  objective: maximize profit


---- VAR x  number of batches of product i to make
             LOWER          LEVEL          UPPER         MARGINAL
BRD            .              .            +INF            .
CKS            .            60.0000        +INF            .
ROL            .              .            +INF            .
MUF            .            40.0000        +INF            .
CRE            .              .            +INF            .
SCO            .            35.0000        +INF            .


---- VAR y  whether or not product i is made
             LOWER          LEVEL          UPPER         MARGINAL
BRD            .              .            1.0000        240.0000
CKS            .            1.0000         1.0000        960.0000
ROL            .              .            1.0000        325.0000
MUF            .            1.0000         1.0000        760.0000
CRE            .              .            1.0000        675.0000
SCO            .            1.0000         1.0000        700.0000
```

GAMS solution for **2** maximum products:

```
Proven optimal solution
MIP Solution:          1720.000000     (3 iterations, 0 nodes)
Final Solve:           1720.000000     (0 iterations)

---- EQU DEMAND
             LOWER          LEVEL          UPPER         MARGINAL
BRD          -INF            .              .             12.0000
CKS          -INF            .              .             16.0000
ROL          -INF            .              .             13.0000
MUF          -INF            .              .             19.0000
CRE          -INF            .              .              9.0000
SCO          -INF            .              .             20.0000

---- EQU TIME
               LOWER          LEVEL          UPPER         MARGINAL
MIX             -INF        3080.0000      4800.0000          .
BAKE            -INF        1960.0000      4800.0000          .
PACK            -INF         860.0000      1200.0000          .


                              LOWER          LEVEL          UPPER         MARGINAL
```

```
---- EQU PRODLIM          -INF            2.0000         2.0000            .

                          LOWER           LEVEL          UPPER          MARGINAL
---- VAR PHI              -INF         1720.0000         +INF              .
         PHI  objective: maximize profit

---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .            +INF             .
CKS          .            60.0000        +INF             .
ROL          .              .            +INF             .
MUF          .            40.0000        +INF             .
CRE          .              .            +INF             .
SCO          .              .            +INF             .


---- VAR y  whether or not product i is made
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .             1.0000       240.0000
CKS          .             1.0000         1.0000       960.0000
ROL          .              .             1.0000       325.0000
MUF          .             1.0000         1.0000       760.0000
CRE          .              .             1.0000       675.0000
SCO          .              .             1.0000       700.0000
```

4. In this scenario, we can make anything we want, BUT we are obliged to make 20 batches of cookies we promised to a local retailer (Café Oranje). Reformulate your program to account for this new constraint and re-solve the problem.

$$
\begin{aligned}
\max_{x} \phi \quad &= \quad \sum_i P_i \cdot x_i \\
&\text{s.t.} \\
x_i \quad &\leq \quad D_i \cdot y_i \qquad (\forall i) \\
\sum_i T_{i,j} \cdot x_i \quad &\leq \quad A_j \qquad (\forall j) \\
\sum_i y_i \quad &\leq \quad 4 \\
x_{CKS} \quad &\geq \quad 20 \\
x_i \quad &\geq \quad 0 \qquad (\forall i) \\
y_i \quad &\in \quad \{0,1\} \qquad (\forall i)
\end{aligned}
$$

Additions and changes to the GAMS code for the base-case MILP in Part 2 are given in red. All previously defines sets and parameters remain the same.

```
SCALARS
MAXPROD   maximum number of products the bakery can make   / 6 / ;


VARIABLES
PHI      objective: maximize profit
x(i)     number of batches of product i to make
y(i)     whether or not product i is made ;
POSITIVE VARIABLES x(i) ;
BINARY VARIABLES   y(i) ;
```

```
EQUATIONS
PROFIT
DEMAND
COOKIES
TIME
PRODLIM ;

PROFIT..        PHI =E= SUM(i, P(i)*x(i))    ;
DEMAND(i)..     x(i) =L= D(i)*y(i)           ;
COOKIES..       x('CKS') =G= 20              ;
TIME(j)..       SUM(i,T(i,j)*x(i)) =L= A(j) ;
PRODLIM..       SUM(i,y(i)) =L= MAXPROD      ;

MODEL BAKERY /ALL/;
OPTION MIP = CPLEX;

SOLVE BAKERY MAXIMIZING PHI USING MIP;
```

```
Proven optimal solution
MIP Solution:          2625.000000    (1 iterations, 0 nodes)
Final Solve:           2625.000000    (1 iterations)


---- EQU DEMAND
           LOWER          LEVEL          UPPER         MARGINAL
BRD        -INF          -5.8333           .              .
CKS        -INF         -40.0000           .              .
ROL        -INF         -25.0000           .              .
MUF        -INF            .               .             3.0000
CRE        -INF            .               .             1.0000
SCO        -INF            .               .             2.0000

                          LOWER          LEVEL          UPPER         MARGINAL
---- EQU COOKIES          20.0000        20.0000        +INF          -2.0000

---- EQU TIME
            LOWER          LEVEL          UPPER         MARGINAL
MIX         -INF         4284.1667      4800.0000          .
BAKE        -INF         3653.3333      4800.0000          .
PACK        -INF         1200.0000      1200.0000         2.0000

                          LOWER          LEVEL          UPPER         MARGINAL
---- EQU PRODLIM          -INF            6.0000         6.0000          .


                          LOWER          LEVEL          UPPER         MARGINAL
---- VAR PHI              -INF         2625.0000        +INF            .
         PHI  objective: maximize profit


---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .            14.1667        +INF            .
CKS          .            20.0000        +INF            .
ROL          .              .            +INF          -1.0000
MUF          .            40.0000        +INF            .
CRE          .            75.0000        +INF            .
SCO          .            35.0000        +INF            .


---- VAR y  whether or not product i is made
```

```
             LOWER          LEVEL          UPPER          MARGINAL
BRD            .            1.0000         1.0000           EPS
CKS            .            1.0000         1.0000           EPS
ROL            .            1.0000         1.0000           EPS
MUF            .            1.0000         1.0000         120.0000
CRE            .            1.0000         1.0000          75.0000
SCO            .            1.0000         1.0000          70.0000
```

5. Now consider the case where we can make anything we want again with no cookies required. HOWEVER, there is a fixed charge for making any one product of \$300 (so \$300 per product we choose to make in ANY quantity), which is the delivery charge for the necessary ingredients (also accounts for you investing your own time to unload and store all the ingredients). Reformulate your problem again to account for this new cost. Re-solve it using GAMS to report the new optimum.

$$
\begin{aligned}
\max_{x} \phi \quad &= \quad \sum_i (P_i \cdot x_i - 300y_i) \\
&\text{s.t.} \\
x_i \quad &\leq \quad D_i \cdot y_i \qquad (\forall i) \\
\sum_i T_{i,j} \cdot x_i \quad &\leq \quad A_j \qquad (\forall j) \\
x_i \quad &\geq \quad 0 \qquad (\forall i) \\
y_i \quad &\in \quad \{0,1\} \qquad (\forall i)
\end{aligned}
$$

Additions and changes to the GAMS code for the base-case MILP in Part 2 are given in red. All previously defines sets and parameters remain the same.

```
SCALARS
MAXPROD    maximum number of products the bakery can make    / 6   /
F          fixed cost of making any one product              / 300 / ;

VARIABLES
PHI       objective: maximize profit
x(i)      number of batches of product i to make
y(i)      whether or not product i is made ;
POSITIVE VARIABLES x(i) ;
BINARY VARIABLES    y(i) ;


EQUATIONS
PROFIT
DEMAND
TIME
PRODLIM ;

PROFIT..         PHI =E= SUM(i, P(i)*x(i) - F*y(i))    ;
DEMAND(i)..      x(i) =L= D(i)*y(i)                    ;
TIME(j)..        SUM(i,T(i,j)*x(i)) =L= A(j)           ;
PRODLIM..        SUM(i,y(i)) =L= MAXPROD               ;

MODEL BAKERY /ALL/;
OPTION MIP = CPLEX;
```

```
SOLVE BAKERY MAXIMIZING PHI USING MIP;
```

```
Proven optimal solution
MIP Solution:          1520.000000    (3 iterations, 0 nodes)
Final Solve:           1520.000000    (0 iterations)

Best possible:         1520.000000
Absolute gap:             0.000000
Relative gap:             0.000000

---- EQU DEMAND
           LOWER          LEVEL          UPPER         MARGINAL
BRD        -INF             .              .            12.0000
CKS        -INF             .              .            16.0000
ROL        -INF             .              .            13.0000
MUF        -INF             .              .            19.0000
CRE        -INF             .              .             9.0000
SCO        -INF             .              .            20.0000

---- EQU TIME
            LOWER          LEVEL          UPPER         MARGINAL
MIX         -INF         3850.0000      4800.0000          .
BAKE        -INF         2695.0000      4800.0000          .
PACK        -INF         1175.0000      1200.0000          .

                           LOWER          LEVEL          UPPER         MARGINAL
---- EQU PRODLIM           -INF            3.0000         6.0000          .


                           LOWER          LEVEL          UPPER         MARGINAL
---- VAR PHI               -INF         1520.0000        +INF            .
         PHI  objective: maximize profit

---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .            +INF             .
CKS          .            60.0000        +INF             .
ROL          .              .            +INF             .
MUF          .            40.0000        +INF             .
CRE          .              .            +INF             .
SCO          .            35.0000        +INF             .


---- VAR y  whether or not product i is made
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .             1.0000        -60.0000
CKS          .             1.0000         1.0000        660.0000
ROL          .              .             1.0000         25.0000
MUF          .             1.0000         1.0000        460.0000
CRE          .              .             1.0000        375.0000
SCO          .             1.0000         1.0000        400.0000
```

6. Return to the base case again. Argue convincingly that if I were to impose on you the requirement "if you make any of one product, you must make at least 10 batches of it", the optimum will either go down or stay the same in the best possible case. This would be true if, for instance, ingredients come in minimum shipments of 10 batches-worth.

If we are required to make at least 10 batches of any product we choose to make, imposing this constraint essentially reduces the size of the feasible region (tightening constraints) and thus will either make the optimal objective function value worse, or unchanged at best. In this case, the optimum will be unchanged because we can see in the base case solution that we are already making more than 10 batches for all products that are being made.

7. Re-solve the problem with this added constraint to prove your point.

$$\max_{x} \phi = \sum_i P_i \cdot x_i$$

s.t.

$$
\begin{aligned}
x_i &\leq D_i \cdot y_i && (\forall i)\\
x_i &\geq 10 \cdot y_i && (\forall i)\\
\sum_i T_{i,j} \cdot x_i &\leq A_j && (\forall j)\\
x_i &\geq 0 && (\forall i)\\
y_i &\in \{0,1\} && (\forall i)
\end{aligned}
$$

Additions and changes to the GAMS code for the base-case MILP in Part 2 are given in red. All previously defines sets and parameters remain the same.

```
SCALARS
MAXPROD   maximum number of products the bakery can make   / 6  /
MINBATCH  minimum number of batches that must be made      / 10 / ;

VARIABLES
PHI      objective: maximize profit
x(i)     number of batches of product i to make
y(i)     whether or not product i is made ;
POSITIVE VARIABLES x(i) ;
BINARY VARIABLES   y(i) ;


EQUATIONS
PROFIT
DEMAND
TIME
PRODLIM
BATCHES ;

PROFIT..        PHI =E= SUM(i, P(i)*x(i))    ;
DEMAND(i)..     x(i) =L= D(i)*y(i)           ;
TIME(j)..       SUM(i,T(i,j)*x(i)) =L= A(j) ;
PRODLIM..       SUM(i,y(i)) =L= MAXPROD      ;
BATCHES(i)..    x(i) =G= MINBATCH*y(i)       ;


MODEL BAKERY /ALL/;
OPTION MIP = CPLEX;

SOLVE BAKERY MAXIMIZING PHI USING MIP;
```

```
Proven optimal solution
```

```
MIP Solution:         2644.285714    (1 iterations, 0 nodes)
Final Solve:          2644.285714    (1 iterations)

---- EQU DEMAND
           LOWER          LEVEL          UPPER          MARGINAL
BRD        -INF             .              .              0.8571
CKS        -INF             .              .               .
ROL        -INF           -4.2857          .               .
MUF        -INF             .              .              4.1429
CRE        -INF             .              .              1.5714
SCO        -INF             .              .              3.2857


---- EQU TIME
            LOWER          LEVEL          UPPER          MARGINAL
MIX         -INF         4201.4286      4800.0000          .
BAKE        -INF         3546.4286      4800.0000          .
PACK        -INF         1200.0000      1200.0000         1.8571

                            LOWER          LEVEL          UPPER          MARGINAL
---- EQU PRODLIM            -INF            5.0000         6.0000          .


---- EQU BATCHES
           LOWER          LEVEL          UPPER          MARGINAL
BRD          .           10.0000        +INF             .
CKS          .             .            +INF             .
ROL          .           10.7143        +INF             .
MUF          .           30.0000        +INF             .
CRE          .           65.0000        +INF             .
SCO          .           25.0000        +INF             .


                            LOWER          LEVEL          UPPER          MARGINAL
---- VAR PHI                -INF         2644.2857       +INF             .
         PHI  objective: maximize profit


---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER          MARGINAL
BRD          .           20.0000        +INF             .
CKS          .             .            +INF           -0.7143
ROL          .           20.7143        +INF             .
MUF          .           40.0000        +INF             .
CRE          .           75.0000        +INF             .
SCO          .           35.0000        +INF             .


---- VAR y  whether or not product i is made
           LOWER          LEVEL          UPPER          MARGINAL
BRD          .            1.0000         1.0000        17.1429
CKS          .             .             1.0000          EPS
ROL          .            1.0000         1.0000          EPS
MUF          .            1.0000         1.0000       165.7143
CRE          .            1.0000         1.0000       117.8571
SCO          .            1.0000         1.0000       115.0000
```

8. Re-formulate and re-solve the optimization so that if I want to make any one product, I am obliged to make 30 batches. You may adjust the demand data such that any product with a demand < 30 can be equal to 30 instead. Comment on your findings.

$$
\begin{aligned}
\max_{x} \phi &= \sum_i P_i \cdot x_i \\
&\text{s.t.} \\
x_i &\le D_i \cdot y_i && (\forall i) \\
x_i &\ge 30 \cdot y_i && (\forall i) \\
\sum_i T_{i,j} \cdot x_i &\le A_j && (\forall j) \\
x_i &\ge 0 && (\forall i) \\
y_i &\in \{0,1\} && (\forall i)
\end{aligned}
$$

Additions and changes to the GAMS code for the MILP in the previous question are given in red. All previously defines sets and parameters remain the same, except the values for the maximum demand parameter D_i were increased to 30 for products $i = \mathrm{BRD}$ and $i = \mathrm{ROL}$.

```
SCALARS
MAXPROD   maximum number of products the bakery can make    / 6  /
MINBATCH  minimum number of batches that must be made       / 30 / ;

VARIABLES
PHI      objective: maximize profit
x(i)     number of batches of product i to make
y(i)     whether or not product i is made ;
POSITIVE VARIABLES x(i) ;
BINARY VARIABLES   y(i) ;


EQUATIONS
PROFIT
DEMAND
TIME
PRODLIM
BATCHES ;

PROFIT..         PHI =E= SUM(i, P(i)*x(i))    ;
DEMAND(i)..      x(i) =L= D(i)*y(i)           ;
TIME(j)..        SUM(i,T(i,j)*x(i)) =L= A(j) ;
PRODLIM..        SUM(i,y(i)) =L= MAXPROD      ;
BATCHES(i)..     x(i) =G= MINBATCH*y(i)       ;


MODEL BAKERY /ALL/;
OPTION MIP = CPLEX;

SOLVE BAKERY MAXIMIZING PHI USING MIP;
```

```
Proven optimal solution
MIP Solution:         2605.000000    (20 iterations, 0 nodes)
Final Solve:          2605.000000    (0 iterations)

---- EQU DEMAND
           LOWER          LEVEL          UPPER         MARGINAL
BRD        -INF             .              .               .
CKS        -INF             .              .               .
ROL        -INF             .              .               .
MUF        -INF             .              .              1.0000
CRE        -INF          -20.0000          .               .
SCO        -INF           -5.0000          .               .
```

```
---- EQU TIME
            LOWER          LEVEL          UPPER         MARGINAL
MIX          -INF        4170.0000      4800.0000          .
BAKE         -INF        3250.0000      4800.0000          .
PACK         -INF        1200.0000      1200.0000        2.2500

---- EQU BATCHES
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .              .            +INF          -1.5000
CKS          .              .            +INF            .
ROL          .              .            +INF          -2.7500
MUF          .            10.0000        +INF            .
CRE          .            25.0000        +INF            .
SCO          .              .            +INF          -0.2500

                           LOWER          LEVEL          UPPER         MARGINAL
---- VAR PHI               -INF         2605.0000        +INF             .
         PHI  objective: maximize profit


---- VAR x  number of batches of product i to make
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .            30.0000        +INF            .
CKS          .              .            +INF          -4.2500
ROL          .            30.0000        +INF            .
MUF          .            40.0000        +INF            .
CRE          .            55.0000        +INF            .
SCO          .            30.0000        +INF            .


---- VAR y  whether or not product i is made
           LOWER          LEVEL          UPPER         MARGINAL
BRD          .             1.0000         1.0000       -45.0000
CKS          .              .             1.0000         EPS
ROL          .             1.0000         1.0000       -82.5000
MUF          .             1.0000         1.0000        40.0000
CRE          .             1.0000         1.0000         EPS
SCO          .             1.0000         1.0000        -7.5000
```

Problem 2 – Optimization… IN SPAAAAAAACE!

The US space agency known as the National Aeronautics and Space Adventuring Helionauts (NASAH) typically has to face decisions around budgeting long-term missions and projects. For example, consider the case where they have funding available for a series of projects available over a five-stage period of time. Each project requires a certain amount of money in each stage, and each stage has a well-defined budget from the government. In the **table below** is a list of prospective projects and their anticipated cost at each stage of the budgeting plan. Furthermore, in the table is the expected "benefit" (an aggregate score based on likelihood of success and information gathered as a result of the mission) of each project and a list of projects that depend on or are mutually exclusive to it. Note that if you choose a project in the table below you **must pay for all costs in each stage for that project**.

This type of problem can be defined as a "multi-dimensional knapsack" problem in which we are choosing which projects to take depending on capacity (*i.e.* budget) constraints for multiple stages. It is our objective to allocate research funds that will maximize the benefit of NASAH's funding over the next five stages.

		Budget Requirement ($billions)							
i	Mission	Stage 1	Stage 2	Stage 3	Stage 4	Stage 5	Value	Not With	Depends On
1	Comms Satellite	6	-	-	-	-	200	-	-
2	Orbital Microwave	2	3	-	-	-	3	-	-
3	IO Lander	3	5	-	-	-	20	-	-
4	Uranus Orbiter 2030	-	-	-	-	10	50	5	3
5	Uranus Orbiter 2040	-	5	8	-	-	70	4	3
6	Jupiter Probe	-	-	1	8	4	20	-	3
7	Saturn Probe	1	8	-	-	-	5	-	3
8	Infrared Imaging	-	-	-	5	-	10	11	-
9	Ground-Based SETI	4	5	-	-	-	200	13	-
10	Orbital Structures	-	8	4	-	-	150	-	-
11	Color Imaging	-	-	2	7	-	18	8	2
12	Medical Technology	5	7	-	-	-	8	-	-
13	Geosynchronous SETI	-	4	5	3	3	185	9	-
14	Jetpod Shielding	-	-	3	3	1	50	-	12
Budget		10	12	14	14	14			

Questions

1. Define the variables for this problem, clearly indicating if they are continuous or binary.
   We can define the binary variable $y_i \in \{0,1\}$ to represent the choice of selecting whether or not mission i will be funded.

2. Define the objective for this problem. You may write this out in terms of hard-coded numbers or define new parameters as you see fit.
   The objective for this problem is to select the projects to fund which maximize the overall value (benefit) while respecting the budget. The objective function can be defined mathematically as:

$$\phi = \sum_{i=1}^{14} V_i \cdot y_i$$

Where $i = 1 \ldots 14$ is the set of available missions and V_i represents the expected value (benefit) associated with funding project i.

3. Define the constraint(s) that correspond(s) to the available budget for each stage in each for this problem. Try to be as concise as possible.
We can define the set $j = 1 \ldots 5$ to be the project stages. With this, a constraint that ensures we do not exceed the maximum available budget for each project stage can be written as:

$$\sum_{i=1}^{14} C_{i,j} \cdot y_i \leq B_j \qquad (\forall j)$$

Where $C_{i,j}$ represents the cost (in billions of dollars) of carrying out mission i in project stage j and B_j represents the total budget that is allocated for each stage j.

4. Code your formulation *so far* (that is, DO NOT include the mutual exclusivity or dependence constraints) in GAMS and find the solution.
Overall formulation:

$$\begin{aligned} \max_{y} \phi \quad &= \sum_i V_i \cdot y_i \\ &s.t. \\ \sum_i C_{i,j} y_i \quad &\leq B_j \qquad (\forall i) \\ y_i \quad &\in \{0,1\} \qquad (\forall i) \end{aligned}$$

The GAMS code and solution are given below.

```
SETS
i   Missions to choose from   / M1*M14 /
j   Stage of time period      / S1*S5  / ;


PARAMETERS
V(i)     Expected benefit of a mission
/ M1     200
  M2     3
  M3     20
  M4     50
  M5     70
  M6     20
  M7     5
  M8     10
  M9     200
  M10    150
  M11    18
  M12    8
  M13    185
  M14    50 /

B(j)    Total budget allocated for a stage($billions)
/ S1     10
  S2     12
  S3     14
  S4     14
  S5     14 /  ;
```

```
TABLE C(i,j) Budget requirement for the missions($billions)
          S1      S2      S3      S4      S5
M1        6       0       0       0       0
M2        2       3       0       0       0
M3        3       5       0       0       0
M4        0       0       0       0       10
M5        0       5       8       0       0
M6        0       0       1       8       4
M7        1       8       0       0       0
M8        0       0       0       5       0
M9        4       5       0       0       0
M10       0       8       4       0       0
M11       0       0       2       7       0
M12       5       7       0       0       0
M13       0       4       5       3       3
M14       0       0       3       3       1    ;


VARIABLES
Z       Objective function variable
y(i)    Whether or not mission i will be funded ;
BINARY VARIABLES y(i) ;


EQUATIONS
OBJ            Objective function
BUDGET         Budget constraints   ;

OBJ..          Z =E= SUM(i, V(i)*y(i)) ;
BUDGET(j)..    SUM(i, C(i,j)*y(i)) =L= B(j) ;

Model SPACE /all/ ;
Solve SPACE MAXIMIZING Z USING MIP ;
```

```
Proven optimal solution
MIP Solution:          703.000000    (4 iterations, 0 nodes)
Final Solve:           703.000000    (0 iterations)

---- EQU BUDGET  Budget constraints
          LOWER          LEVEL          UPPER         MARGINAL
S1        -INF          10.0000        10.0000          .
S2        -INF           9.0000        12.0000          .
S3        -INF          10.0000        14.0000          .
S4        -INF          13.0000        14.0000          .
S5        -INF          14.0000        14.0000          .

                           LOWER          LEVEL          UPPER          MARGINAL
---- VAR Z                 -INF          703.0000        +INF              .
         Z  Objective function variable

---- VAR y  Whether or not mission i will be funded
           LOWER          LEVEL          UPPER         MARGINAL
M1           .            1.0000         1.0000        200.0000
M2           .              .            1.0000          3.0000
M3           .              .            1.0000         20.0000
M4           .            1.0000         1.0000         50.0000
M5           .              .            1.0000         70.0000
M6           .              .            1.0000         20.0000
M7           .              .            1.0000          5.0000
M8           .              .            1.0000         10.0000
M9           .            1.0000         1.0000        200.0000
M10          .              .            1.0000        150.0000
M11          .            1.0000         1.0000         18.0000
M12          .              .            1.0000          8.0000
```

```
M13           .            1.0000         1.0000        185.0000
M14           .            1.0000         1.0000         50.0000
```

5. Define the mutual exclusivity constraint for this problem.
   The mutual exclusivity constraints are based on the "Not With" column in the table. This column indicates that you cannot complete missions 4 and 5, 8 and 11, or 9 and 13 at the same time. These constraints can be written mathematically as:

$$y_4 + y_5 \leq 1$$

$$y_8 + y_{11} \leq 1$$

$$y_9 + y_{13} \leq 1$$

6. Add the mutual exclusivity constraints to GAMS and re-run the problem. Comment on the results and how adding the mutual exclusivity constraints affect your solution.
   Only the `EQUATIONS` section of the GAMS code will change from adding the mutual exclusivity constraints. Additions to the base case are highlighted with red text.

```
EQUATIONS
OBJ            Objective function
BUDGET         Budget constraints
ME1            Mutual exclusivity constraint 1
ME2            Mutual exclusivity constraint 2
ME3            Mutual exclusivity constraint 3 ;

OBJ..          Z =E= SUM(i, V(i)*y(i)) ;
BUDGET(j)..    SUM(i, C(i,j)*y(i)) =L= B(j) ;
ME1..          y('M4') + y('M5')  =L= 1 ;
ME2..          y('M8') + y('M11') =L= 1 ;
ME3..          y('M9') + y('M13') =L= 1 ;
```

```
Proven optimal solution
MIP Solution:          653.000000    (5 iterations, 0 nodes)
Final Solve:           653.000000    (0 iterations)


---- EQU BUDGET  Budget constraints
          LOWER          LEVEL          UPPER         MARGINAL
S1        -INF            6.0000        10.0000          .
S2        -INF           12.0000        12.0000          .
S3        -INF           14.0000        14.0000          .
S4        -INF           13.0000        14.0000          .
S5        -INF           14.0000        14.0000          .


                           LOWER          LEVEL          UPPER         MARGINAL
---- EQU ME1               -INF            1.0000         1.0000          .
---- EQU ME2               -INF            1.0000         1.0000          .
---- EQU ME3               -INF            1.0000         1.0000          .
  ME1  Mutual exclusivity constraint 1
  ME2  Mutual exclusivity constraint 2
  ME3  Mutual exclusivity constraint 3


                           LOWER          LEVEL          UPPER         MARGINAL
---- VAR Z                 -INF          653.0000        +INF             .
```

```
        Z  Objective function variable



---- VAR y  Whether or not mission i will be funded
           LOWER          LEVEL          UPPER         MARGINAL
M1           .            1.0000         1.0000        200.0000
M2           .              .            1.0000          3.0000
M3           .              .            1.0000         20.0000
M4           .            1.0000         1.0000         50.0000
M5           .              .            1.0000         70.0000
M6           .              .            1.0000         20.0000
M7           .              .            1.0000          5.0000
M8           .              .            1.0000         10.0000
M9           .              .            1.0000        200.0000
M10          .            1.0000         1.0000        150.0000
M11          .            1.0000         1.0000         18.0000
M12          .              .            1.0000          8.0000
M13          .            1.0000         1.0000        185.0000
M14          .            1.0000         1.0000         50.0000
```

7. Define the dependency constraints for this problem.
The dependency constraints are based on the "Depends On" column in the table, where it is indicated that: (a) missions 4, 5 and 6 depend on mission 3, (b) mission 11 depends on mission 2 and (c) mission 14 depends on completing mission 12. These dependency constraints can be written mathematically as:

$$y_4 \leq y_3$$

$$y_5 \leq y_3$$

$$y_6 \leq y_3$$

$$y_{11} \leq y_2$$

$$y_{14} \leq y_{12}$$

8. Add the dependency constraints to the **base case** in GAMS (that is, don't include the mutual exclusivity constraints) and re-run the problem. Comment on the results and how adding the dependence constraints affect your solution.
Only the `EQUATIONS` section of the GAMS code will change from adding the dependency constraints. Additions to the base case are highlighted with red text.

```
EQUATIONS
OBJ           Objective function
BUDGET        Budget constraints
DC1           Dependency constraint 1
DC2           Dependency constraint 2
DC3           Dependency constraint 3
DC4           Dependency constraint 4
DC5           Dependency constraint 5 ;

OBJ..         Z =E= SUM(i, V(i)*y(i)) ;
BUDGET(j)..   SUM(i, C(i,j)*y(i)) =L= B(j) ;
DC1..         y('M4')  =L= y('M3')  ;
DC2..         y('M5')  =L= y('M3')  ;
DC3..         y('M6')  =L= y('M3')  ;
DC4..         y('M11') =L= y('M2')  ;
DC5..         y('M14') =L= y('M12') ;
```

```
Proven optimal solution
MIP Solution:           595.000000    (6 iterations, 0 nodes)
Final Solve:            595.000000    (0 iterations)

---- EQU BUDGET  Budget constraints
          LOWER          LEVEL          UPPER         MARGINAL
S1        -INF          10.0000        10.0000          .
S2        -INF           9.0000        12.0000          .
S3        -INF           5.0000        14.0000          .
S4        -INF           8.0000        14.0000          .
S5        -INF           3.0000        14.0000          .

                           LOWER          LEVEL          UPPER         MARGINAL
---- EQU DC1               -INF             .              .              .
---- EQU DC2               -INF             .              .              .
---- EQU DC3               -INF             .              .              .
---- EQU DC4               -INF             .              .              .
---- EQU DC5               -INF             .              .              .
  DC1  Dependency constraint 1
  DC2  Dependency constraint 2
  DC3  Dependency constraint 3
  DC4  Dependency constraint 4
  DC5  Dependency constraint 5

                           LOWER          LEVEL          UPPER         MARGINAL
---- VAR Z                 -INF          595.0000        +INF             .
         Z  Objective function variable

---- VAR y  Whether or not mission i will be funded
           LOWER          LEVEL          UPPER         MARGINAL
M1           .            1.0000         1.0000        200.0000
M2           .              .            1.0000          3.0000
M3           .              .            1.0000         20.0000
M4           .              .            1.0000         50.0000
M5           .              .            1.0000         70.0000
M6           .              .            1.0000         20.0000
M7           .              .            1.0000          5.0000
M8           .            1.0000         1.0000         10.0000
M9           .            1.0000         1.0000        200.0000
M10          .              .            1.0000        150.0000
M11          .              .            1.0000         18.0000
M12          .              .            1.0000          8.0000
M13          .            1.0000         1.0000        185.0000
M14          .              .            1.0000         50.0000
```

9. Re-run your GAMS code with *both* of the mutual/dependence constraints included. Comment on *this* solution.
Only the `EQUATIONS` section of the GAMS code will change from adding the mutual exclusivity constraints. Additions to the base case are highlighted with red text.

```
EQUATIONS
OBJ           Objective function
BUDGET        Budget constraints
ME1           Mutual exclusivity constraint 1
ME2           Mutual exclusivity constraint 2
ME3           Mutual exclusivity constraint 3
DC1           Dependency constraint 1
DC2           Dependency constraint 2
DC3           Dependency constraint 3
DC4           Dependency constraint 4
```

```
DC5            Dependency constraint 5 ;

OBJ..          Z =E= SUM(i, V(i)*y(i)) ;
BUDGET(j)..    SUM(i, C(i,j)*y(i)) =L= B(j) ;
ME1..          y('M4') + y('M5')  =L= 1 ;
ME2..          y('M8') + y('M11') =L= 1 ;
ME3..          y('M9') + y('M13') =L= 1 ;
DC1..          y('M4')  =L= y('M3')  ;
DC2..          y('M5')  =L= y('M3')  ;
DC3..          y('M6')  =L= y('M3')  ;
DC4..          y('M11') =L= y('M2')  ;
DC5..          y('M14') =L= y('M12') ;
```

```
Proven optimal solution
MIP Solution:          545.000000    (8 iterations, 0 nodes)
Final Solve:           545.000000    (0 iterations)

---- EQU BUDGET  Budget constraints
          LOWER          LEVEL          UPPER         MARGINAL
S1        -INF            6.0000        10.0000          .
S2        -INF           12.0000        12.0000          .
S3        -INF            9.0000        14.0000          .
S4        -INF            8.0000        14.0000          .
S5        -INF            3.0000        14.0000          .

                           LOWER          LEVEL          UPPER         MARGINAL
---- EQU ME1               -INF             .            1.0000           .
---- EQU ME2               -INF           1.0000         1.0000           .
---- EQU ME3               -INF           1.0000         1.0000           .
---- EQU DC1               -INF             .              .              .
---- EQU DC2               -INF             .              .              .
---- EQU DC3               -INF             .              .              .
---- EQU DC4               -INF             .              .              .
---- EQU DC5               -INF             .              .              .
  ME1  Mutual exclusivity constraint 1
  ME2  Mutual exclusivity constraint 2
  ME3  Mutual exclusivity constraint 3
  DC1  Dependency constraint 1
  DC2  Dependency constraint 2
  DC3  Dependency constraint 3
  DC4  Dependency constraint 4
  DC5  Dependency constraint 5

                           LOWER          LEVEL          UPPER         MARGINAL
---- VAR Z                 -INF          545.0000        +INF             .
         Z  Objective function variable


---- VAR y  Whether or not mission i will be funded
           LOWER          LEVEL          UPPER         MARGINAL
M1           .            1.0000         1.0000        200.0000
M2           .              .            1.0000          3.0000
M3           .              .            1.0000         20.0000
M4           .              .            1.0000         50.0000
M5           .              .            1.0000         70.0000
M6           .              .            1.0000         20.0000
M7           .              .            1.0000          5.0000
M8           .            1.0000         1.0000         10.0000
M9           .              .            1.0000        200.0000
M10          .            1.0000         1.0000        150.0000
M11          .              .            1.0000         18.0000
M12          .              .            1.0000          8.0000
M13          .            1.0000         1.0000        185.0000
```

M14	.	.	1.0000	50.0000

10. Feel free to play around with costs, budgets and mutual exclusivities to see how they affect the problem!