Problem 1

The image of Homer Simpson1 below was drawn using 25 complete set of double-9 dominos in the following way: the (high resolution) target image was converted to grayscale; with each pixel's intensity rounded to a value between 0 (black) and 255 (white). The image was then divided in a rectangular grid with a total number of 2 × 25 × 55 cells. (25 for the number of domino sets; 55 is the number of dominos in a set of double-9 dominos.) Each domino has to cover exactly two adjacent cells of this grid. The mean gray value of all the pixels contained in each grid cell was computed and rescaled and rounded from [0, 255] to the target intensity {0,…, 9} to match the number of dots of our dominos. Now, when a domino is placed on a pair of cells, the error (or "cost") is computed in each cell as the absolute difference between the target intensity value of the cell and the number of dots on the half-domino covering this cell. The goal is to find a placement of all the dominos on this grid such that the overall error (the sum of the error of all cells) is minimum. Formulate this as an integer linear program with binary variables.

minimize

$$\sum_{l,r,i,j} \left(\left| T_{i,j} - C_{l,r,o,i,j} \right| * D_{l,r,o,i,j} \right)$$

subject to

$$\sum_{o,i,j} D_{l,r,o,i,j} = 25$$

$$\sum_{l,r} D_{l,r,o,i,j} = 1$$

$$D_{l,r,o,i,j} \ \in \{0,1\}$$

$$T_{i,j} \ \in \{1,2, \dots ,9\}$$

$$C_{l,r,o,i,j} \in \{1,2, \dots ,9\}$$

where

$T_{i,j}$ – the target intensity value (1-9) of cell i, j ($i =$ row and $j =$ column)

$C_{l,r,o,i,j}$ – the number of dots (1-9) on the left/upper half of domino l, r in cell i, j with orientation o

$D_{l,r,o,i,j}$ – 1 if the left/upper half of domino l, r is placed in cell i, j with orientation o, 0 if not

Problem 2

Hypothetical City is reviewing its public safety spending, starting with the Fire Department budget. The map shows the location of currently operational fire stations. The sizes of the circles indicate the number of engine-runs each station had last year.

The city would like to close at least three stations. For safety reasons, no two stations can be simultaneously closed if either one is the other's closest neighbor. Also, purely for safety reasons, the station closest to the mayor's house can only be closed if the station closest to the governor's house is also closed.

Four of the stations were recently remodeled. For PR reasons, no more than two of those can be closed.

You are given the ungrateful task of deciding which fire stations to close. Formulate an integer program that minimizes the total number of engine-runs that the closed stations currently have, while satisfying the above constraints.

The map is for illustration only; assume that you are given the actual locations of the stations along with the data on the runs, the locations of all other buildings relevant to the problem, and everything else you might need. Clearly define your notation for the data used, your variables, and your constraints.

minimize

$$e_1 S_1 + e_2 S_2 + e_3 S_3 + \cdots$$

subject to

$$\sum S_i \geq 3$$

$$S_i + S_{c_i} \geq 1$$

$$S_m \geq S_g$$

$$\sum S_{i_r} \geq 2$$

$$S_i \in \{0,1\}, e_i > 0$$

where

- $S_i = 1$ if station i is shut down, 0 if not
- c_i – the number of the station that corresponds to the closest neighbor of station number i
- m – the number of the station closest to the mayor's house
- g – the number of the station closest to the governor's house
- i_r – the station numbers ($r = 1 - 4$) of the 4 recently remodeled stations
- e_i – the number of engine-runs of station i

Problem 3

Your machine shop has five jobs to do today. Each job must be done by exactly one of your three machines, but a machine can do more than one job, one after another. For each machine, there is a setup cost for the regular work day. Once this cost is paid, you can use the machine for up to 8 hours; if you do not pay the setup cost, you cannot use the machine today. If you set up a machine for the regular work day, you may also elect to set up the machine for overtime, in which case there is an additional cost and you may use the machine for up to 2 more hours. The costs are as follows:

	Setup cost ($)	Overtime cost ($)
Machine 1	600	190
Machine 2	585	150
Machine 3	700	200

Thus, if you pay $600, you can use machine 1 for up to 8 hours, and if you pay $600+$190=$790, you may use the machine 1 for up to 8+2=10 hours. The number of hours each job takes on each machine is as follows:

	Job 1	Job 2	Job 3	Job 4	Job 5
Machine 1	2.0	2.0	4.5	1.0	4.1
Machine 2	2.5	2.5	4.5	1.5	4.7
Machine 3	2.1	2.1	4.0	2.5	4.5

a) Formulate a mixed-integer linear optimization (MILP) model to find the cheapest way to get all 5 jobs done today.

minimize

$$600m_1 + 585m_2 + 700m_3 + 190o_1 + 150o_2 + 200o_3$$

subject to

$$\sum_i x_{i,1} = 1, \sum_i x_{i,2} = 1, \sum_i x_{i,3} = 1, \sum_i x_{i,4} = 1, \sum_i x_{i,5} = 1$$

$$2x_{1,1} + 2x_{1,2} + 4.5x_{1,3} + 1x_{1,4} + 4.1x_{1,5} \leq 8m_1 + 2o_1$$

$$2.5x_{2,1} + 2.5x_{2,2} + 4.5x_{2,3} + 1.5x_{2,4} + 4.7x_{2,5} \leq 8m_2 + 2o_2$$

$$2.1x_{3,1} + 2.1x_{3,2} + 4x_{3,3} + 2.5x_{3,4} + 4.5x_{3,5} \leq 8m_3 + 2o_3$$

$$m_i \leq o_i$$

$$m_i \in \{0,1\}, o_i \in \{0,1\}, x_{i,j} \in \{0,1\}$$

where

m_i – 1 if machine number i $\{1-3\}$ is used, 0 if it is unused

o_i – 1 if machine number i $\{1-3\}$ is used for overtime, 0 if it is unused for overtime

$x_{i,j}$ – 1 if machine i $\{1-3\}$ does job j $\{1-5\}$, 0 if not used

b) Compute the optimal solution. Input and output must be submitted.

Matlab input:

```
f      = [600;585;700;190;150;200;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
intcon = [1:21];
Aineq =  zeros(6,21);
    Aineq(1,[7,8,9,10,11,1,4])=[2,2,4.5,1,4.1,-8,-2];
    Aineq(2,[12,13,14,15,16,2,5])=[2.5,2.5,4.5,1.5,4.7,-8,-2];
    Aineq(3,[17,18,19,20,21,3,6])=[2.1,2.1,4,2.5,4.5,-8,-2];
    Aineq(4,[1,4])=[1,-1];
    Aineq(5,[2,5])=[1,-1];
    Aineq(6,[3,6])=[1,-1];
bineq = [0;0;0;0;0;0];
Aeq   = zeros(5,21);
    Aeq(1,[7,12,17])=[1,1,1];
    Aeq(2,[8,13,18])=[1,1,1];
    Aeq(3,[9,14,19])=[1,1,1];
    Aeq(4,[10,15,20])=[1,1,1];
    Aeq(5,[11,16,21])=[1,1,1];
beq   = [1;1;1;1;1];
lb    = [0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0];
ub    = [1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1;1];
x0    = []; % no initial point
opts  = optimoptions('intlinprog','Display','iter'); % just an example
                                                     % see `doc linprog' for
                                                     % more options

[x,fval,status] = intlinprog(f,intcon,Aineq,bineq,Aeq,beq,lb,ub,x0,opts)
% status == 1 means the certified optimal solution was found
% other possible outcomes?
```

Matlab output:

```
LP:                   Optimal objective value is 1049.650000.

Cut Generation:       Applied 5 implication cuts, 7 mir cuts,
                      2 strong CG cuts, 8 cover cuts,
                      and 2 Gomory cuts.
                      Lower bound is 1177.382177.

Heuristics:           Found 1 solution using total rounding.
                      Upper bound is 1825.000000.
                      Relative gap is 35.32%.

Heuristics:           Found 1 solution using rss.
                      Upper bound is 1525.000000.
                      Relative gap is 22.61%.

Branch and Bound:
```

```
   nodes      total   num int        integer       relative
explored   time (s)  solution           fval        gap (%)
      37       0.08         2   1.525000e+03   0.000000e+00

Optimal solution found.

Intlinprog stopped because the objective value is within a gap tolerance of
the
optimal value, options.AbsoluteGapTolerance = 0 (the default value). The
intcon
variables are integer within tolerance, options.IntegerTolerance = 1e-05 (the
default value).


x =

    1.0000
    1.0000
         0
    1.0000
    1.0000
         0
    1.0000
         0
         0
         0
    1.0000
         0
    1.0000
    1.0000
    1.0000
         0
         0
         0
         0
         0
         0


fval =

   1.5250e+03


status =

     1
```