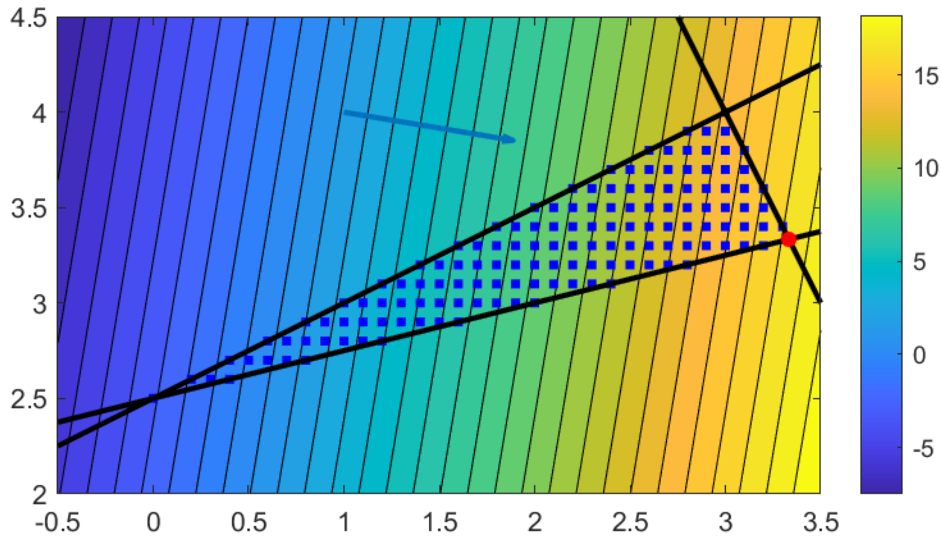
MA3077 (DLI) Operational Research
Dr Marco Fasondini – m.fasondini@leicester.ac.uk

Model Solutions to Problem Sheet 1

Note: The following model solutions indicate how the problem may have been solved. Alternative solutions are often also possible.

1)

a) Using Matlab's script `q1a_feasible_region.m`, we obtain



The solution is denoted in red in the previous picture ($x = 10/3, y = 10/3$). The active constraints are $4y - x = 10$ and $2x + y = 10$.

b) The code reads:

```
f = [-6;1];
A = [1, -4; 2, 1; -1, 2];
b = [-10 10 5];
x = linprog(f,A,b);
```

[4 Marks] Let $c = (6, -1)$, $A = \begin{pmatrix} 1 & -4 \\ 2 & 1 \\ -1 & 2 \end{pmatrix}$ and $b = (-10, 10, 5)^T$ and rewrite the problem as

$$\max c\ x \ \ s.t. Ax \leq b, x \in \mathbb{R}^2,$$

The Lagrangian is $L: \mathbb{R}^2 \times \mathbb{R}^3_+ \to \mathbb{R}$ defined by $L(x, y) \coloneqq c\ x + y^T(b - Ax)$.
The dual function is $g: \mathbb{R}^3_+ \to \mathbb{R}$ defined by

$$g(y) \coloneqq \max_x L(x, y) = \max_x (c - y^T A)x + y^T b \ = \begin{cases} y^T b, & if\ A^T y = c^T, \\ \infty, & otherwise. \end{cases}$$

Hence, the dual problem is $\min y^T b\ s.t. A^T y = c^T, y \geq 0$.

2) Because we want $|x| \in 0\ \cup [1,2]$, we can add the constraint $|x| \leq 2$. Let $t \coloneqq |x|$. As seen in class, this equality can be modeled by introducing two auxiliary scalars $a, b \geq 0$, a discrete variable $z \in \{0,1\}$, and imposing

$$x = a - b, \qquad t = a + b, \qquad a \le 2z, \qquad b \le 2(1 - z).$$

Finally, the condition $t \in 0 \ \cup [1,2]$ can be modelled by introduction $y \in \{0,1\}$ and imposing $y \le t \le 2y$. To sum up, the linear problem reads

$$\max x \ \ s.t. \begin{cases} x = a - b \\ t = a + b \\ a \le 2z \\ b \le 2(1 - z) \\ t \ge y \\ t \le 2y \\ a, b \ge 0, y, z \in \{0,1\}, x \in \mathbb{R} \end{cases}$$

3) Let $y = v - w$, with $v, w \ge 0$, and let $z^T \coloneqq (s, v, w)$. Then, the problem is equivalent to

$$-\min(0, -b^T, b^T)z \ \ s.t. [1, A^T, -A^T]z = c, z \ge 0,$$

which (except for the minus in front of min, although this doesn't not affect the solution) is in standard form (here 1 denotes the identity matrix). By Farkas' lemma, this problem is infeasible iff there is a vector n such that $n^T[1, A^T, -A^T] \le 0$ and $n^T c > 0$. Splitting $n^T[1, A^T, -A^T] \le 0$ into three inequalities we obtain

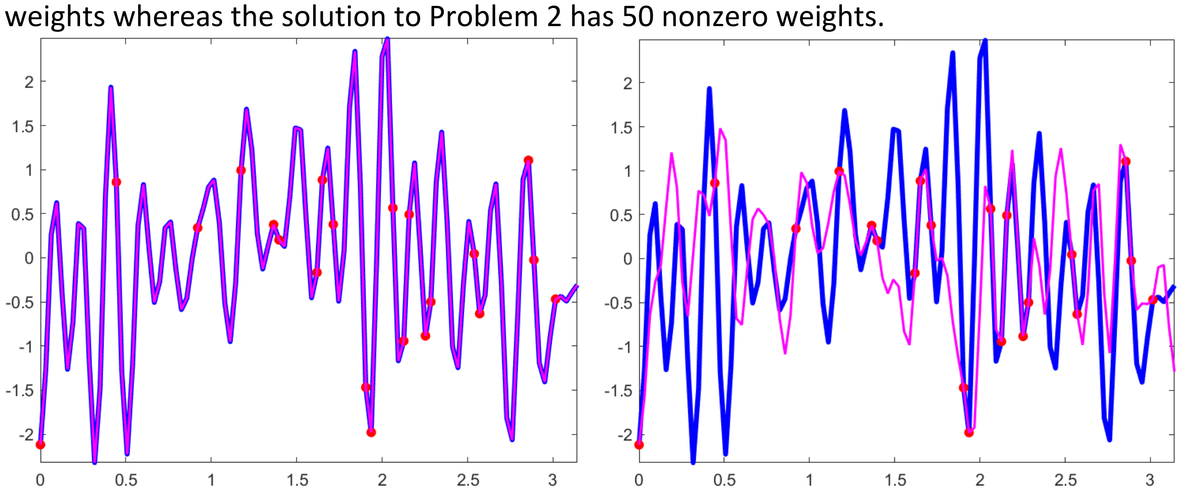
$$n \le 0, \qquad An \le 0, \qquad -An \le 0,$$

which implies $An = 0$. Finally, defining $x := -n$ implies that the original programming problem is infeasible iff there is $x \ge 0$ such that $Ax = 0$ and $c^T x < 0$.

4) Let $\mathbf{1} = (1, 1, \dots, 1)^T \in \mathbb{R}^{50}$, then Problem 2 can be formulated as a linear programming problem as follows:

$$\begin{array}{cc} min & t \\ s.t. & -t\mathbf{1} \le w \le t\mathbf{1} \\ & Aw = y \\ & w \in \mathbb{R}^{50},\ t \ge 0 \end{array}$$

(t is a scalar). For the m-file solving this problem, see PS1_q4.m on Blackboard. The solution to Problem 1 (left) recovers the signal exactly whereas the solution to Problem 2 (right) is not close to recovering original signal. The solution to Problem 1 has 5 nonzero weights whereas the solution to Problem 2 has 50 nonzero weights.



5)

a) First, we set $K_0 = \{O\}$, $T = (K_0, \emptyset)$. Then, in each step we identify the cut-set $C(N, K_k)$, identify an edge e in $C(N, K_k)$ with minimal weight, expand K_k by adding a new node from this edge, and add the edge to the tree. Here is the generated sequence and the resulting graph.

i) $C(N, K_0) = \{OA, OB, OC\}, e = OB, K_1 = \{O, B\}, T_1 = \{K_1, \{OB\})$

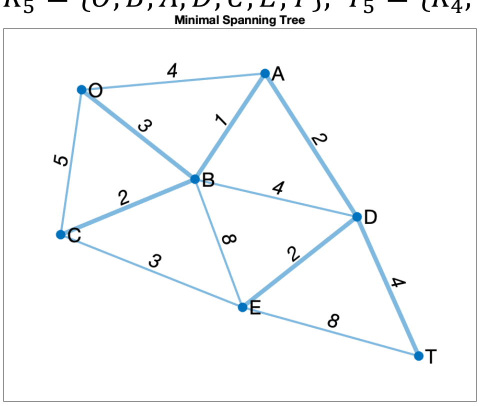
ii) $C(N, K_1) = \{OA, OC, BD, BE, BC\}, e = BA, K_2 = \{O, B, A\}, T_2 = \{K_2, \{OB, BA\})$

iii) $C(N, K_2) = \{OC, AD, BD, BE, BC\}, e = AD,$
$K_3 = \{O, B, A, D\},\ T_3 = \{K_3, \{OB, BA, AD\})$

iv) $C(N, K_3) = \{OC, BC, BE, DE, DT\}, e = BC,$
$K_4 = \{O, B, A, D, C\},\ T_4 = \{K_4, \{OB, BA, AD, BC\})$

v) $C(N, K_4) = \{CE, BE, DE, DT\}, e = DE,$
$K_5 = \{O, B, A, D, C, E\},\ T_4 = \{K_4, \{OB, BA, AD, BC, DE\})$

vi) $C(N, K_5) = \{DT, ET\}, e = DT,$
$K_5 = \{O, B, A, D, C, E, T\},\ T_5 = \{K_4, \{OB, BA, AD, BC, DE, DT\})$



b) The general form of the minimal spanning tree problem as a linear programming problem reads $\min \sum_{e \in E} w(e) x_e \ \ s.t. \begin{cases} \sum_{e \in E} x_e = |V| - 1 \\ \sum_{e \in E'} x_e \leq |V'| - 1 \ \ \forall \text{ full subgraph } (V', E') \\ x_e \in \{0,1\} \ \forall e \in E \end{cases}$

In this case we have 5 edges, which we can label as follow:

$$e_1 = OA. e_2 = OB, e_3 = OC, e_4 = AB, e_5 = BC$$

For the second constraint, we need to consider systematically every full subgraph of (V,E) (note that we can ignore nonconnected subgraphs). Full subgraphs with two nodes just lead to the constraints: $x \leq 1$. Full subgraphs with three nodes lead to the constraints:

OAB $\quad x_1 + x_2 + x_4 \leq 2$

OAC $\quad x_1 + x_3 \leq 2$

OBC $\quad x_2 + x_3 + x_5 \leq 2$

ABC $\quad x_4 + x_5 \leq 2$

The full subgraphs with four nodes is equivalent to the original subgraph. To sum up, the linear program reads:

$$\min w^T x \; s.t. \begin{cases} x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 3 \\ x_1 + x_2 + x_4 \leq 2 \\ x_1 + x_3 \leq 2 \\ x_2 + x_3 + x_5 \leq 2 \\ x_4 + x_5 \leq 2 \\ x_e \in \{0,1\} \; \forall e \in E \end{cases}$$

where $w^T = (4, 3, 5, 1, 2)$. This can be solved in Matlab with the code

```
w= [4,3,5,1,2].'; t = ones(5,1);
A = [1,1,0,1,0; 1,0,1,0,0; 0,1,1,0,1; 0,0,0,1,1];
x = intlinprog(w, 1:5, A, 2*t(2:end), t', 3,0*t, t);
```

which correctly computes `x=(0,1,0,1,1)`

6) Consider the directed network $N = (V, E)$ with source S and sink T on the right. Let $f: E \to \mathbb{R}_+$ be defined by…

a) The function f is a flow because $f(e) \leq w(e)$ for every edge e, and for every node (excluding the source and the sink), the sum of incoming flows equals the sum of the outgoing flows. The nontrivial nodes are B, D, C, and we can quickly verify that indeed

$$f(SB) = 6 = 3 + 3 = f(BC) + f(BD)$$
$$f(BD) = 3 = 0 + 3 = f(DC) + f(DT)$$
$$f(BC) = 3 = 3 = f(CT)$$

b) The path $\{S, C, T\}$ is f-augmenting. Indeed, its capacity is

$$\varepsilon(\{SCT\}) = \min\{w(SC) - f(SC), w(CT) - f(CT)\} = \min\{4, 4 - 3\} = 1 > 0$$

c) The value of the maximal flow is bounded by the capacity of an S-T cut. Since the capacity of the S-T cut induced by $\{T\}$ is 4+6 = 10, the value of the maximal flow is not greater than 10.

7)

a) Strategy 1 and 2 are both optimal for player 1, strategy 1 and 4 are both optimal for player 2, because.

					min
	1	0	2	-1	$\mathbf{-1}$
	0	-1	3	2	$\mathbf{-1}$
	2	-2	-1	1	-2
	0	3	4	-5	-5
max	$\mathbf{2}$	3	4	$\mathbf{2}$	

The game is not stable because the payoffs are not equal. In lay terms, if player 1 picks strategy 1, then player 2 picks strategy 4, but then player 1 picks strategy 2, and then player 2 picks strategy 2, in which case player 1 picks strategy 4, and then player 2 switches to strategy 4, in which case player 1 picks strategy 2, and then player 2 switches to strategy 2, and then player 1 pick strategy 4, and so on.

b) In class we saw that the optimal strategy x^* of player 1 is characterized by the optimization problem

$$\begin{array}{rc} max & v \\ s.t. & A^T x \geq v\,(1, \dots, 1)^T \\ & (1, \dots, 1)^T x = 1 \\ & x \geq 0,\ v \in \mathbb{R} \end{array}$$

We can solve this in matlab with the commands

```
A = [1 0 2 -1; 0 -1 3 2; 2 -2 -1 1; 0 3 4 -5];
A_ = [ones(4,1), -A.'; zeros(4,1), -eye(4)];
f = [-1, zeros(1,4)]; b_ = zeros(8,1);
Aeq = [0, ones(1,4)]; beq = 1;
X = linprog(f,A_,b_,Aeq,beq);
```

The optimal strategy is stored in `X(2:end)` and reads $x^* = (\frac{1}{16}, \frac{11}{16}, 0, \frac{1}{4})$.