Dr. Neslihan Suzen – ns553@leicester.ac.uk

Model Solutions to Problem Sheet 2

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

1)

a. **[2 marks]** 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 A, B, D, C, and we can quickly verify that indeed **[+1]**.

$$f(SC) = 4 = f(CT), \qquad f(SA) = 1 = f(AB) = f(BD) = f(DT)$$

It's value is $f(SA) + f(SB) + f(SC) = 4 + 1 = 5$. **[+1]**.

b. **[2 marks]** The path $\{S, B, D, T\}$ is f-augmenting **[+1]**. Indeed, its capacity is [+1] $\varepsilon(\{SBDT\}) = \min\{w(SB) - f(SB), w(BD) - f(BD), w(DT) - f(DT)\}$ $= \min\{7 - 0, 3 - 1, 6 - 1\} = 2 > 0.$

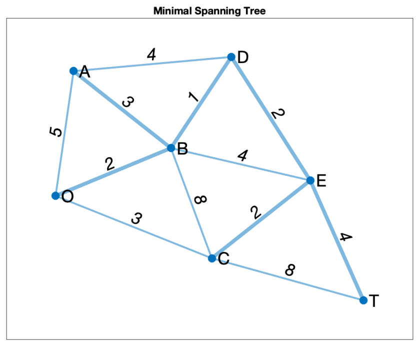
c. **[2 marks]** The value of the maximal flow is bounded by the capacity of an S-T cut. **[+1, even if implicit]** Since the capacity of the S-T cut induced by $\{T\}$ is 4+6 = 10, the value of the maximal flow is not greater than 10. **[+1]**

2)

a. **[3 marks]** 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.

   a. $C(N, K_0) = \{OA, OB, OC\}, e = OB, K_1 = \{O, B\}, T_1 = \{K_1, \{OB\})$
   b. $C(N, K_1) = \{OA, OC, BA, BD, BE, BC\}, e = BD, K_2 = \{O, B, D\}, T_2 = \{K_2, \{OB, BD\})$
   c. $C(N, K_2) = \{OC, BC, BE, DE, DA, BA, OA\}, e = DE,$ $K_3 = \{O, B, D, E\},\ T_3 = \{K_3, \{OB, BD, DE\})$
   d. $C(N, K_3) = \{DA, BA, OA, OC, BC, EC, ET\}, e = EC,$ $K_4 = \{O, B, D, E, C\},\ T_4 = \{K_4, \{OB, BD, DE, EC\})$
   e. $C(N, K_4) = \{DA, BA, OA, ET, CT\}, e = BA,$ $K_5 = \{O, B, D, E, C, A\},\ T_4 = \{K_4, \{OB, BD, DE, EC, BA\})$

f. $C(N, K_5) = \{ET, CT\}, e = ET,$
$K_5 = \{O, B, D, E, C, A, T\},\ T_5 = \{K_4, \{OB, BD, DE, EC, ET\})$



b. **[5 marks]** 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 follows:

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

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

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

where $w^T = (5, 2, 3, 3, 8)$. **[+3]** This can be solved in Matlab with the code

```
w= [5,2,3,3,8].'; 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=(0,1,1,1,0)` **[+2]**

3)

a) **[3 Marks]** Strategies 1 and 3 are both optimal for player 1, strategies 3 and 4 are both optimal for player 2, because. **[+1]**

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

The game is not stable because the maximin and minimax values are not equal. In lay terms, if player 1 picks strategy 1, then player 2 picks strategy 4, but then player 1 picks strategy 3, and then player 2 picks strategy 1, in which case player 1 picks strategy 2, and then player 2 switches to strategy 4, in which case player 1 picks strategy 3, and then player 2 switches to strategy 1, and then player 1 picks strategy 2, and so on. **[+2]**

b) **[4 Marks]** In class we saw that the optimal strategy x^* of player 1 is characterised by the optimisation problem

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

We can solve this in matlab with the commands

```
A = [0 2 1 -1; 3 4 0 -5;-1 3 0 2; -2 -1 2 1];
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 $x^* = (\frac{1}{16}, \frac{1}{4}, \frac{11}{16}, 0)$.

4) Consider a random variable T with exponential distribution alpha.

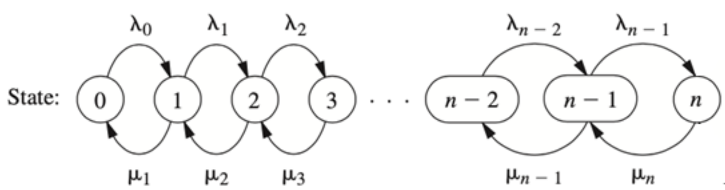
a. **[1 mark]** $E[T] = \alpha^{-1} = 1$, $Var[T] = \alpha^{-2} = 1$

b. **[3 marks]** Since T and V are independent, **[+1]**

$P[1 \le T \le 3 | V \le 3] = P[1 \le T \le 3] = P[T \le 3] - P[T \le 1]$
$= (1 - \exp(-1 * 3)) - (1 - \exp(-1 * 1)) = \exp(-1) - \exp(-3) \approx 0.32$

The sum $T + V + W$ has Erlang distribution with parameters (3,1). Therefore, **[+2]**

$$P[T + V + W \le 1] = 1 - \sum_{k=0}^{2} \frac{(1 * 1)^k \exp(-1 * 1)}{k!} = 1 - \frac{5}{2}\exp(-1).$$



5)

a. **[3 marks]** Draw the above picture with $n = K = 100, \lambda_0 = \lambda_1 = \cdots = \lambda_{99} = 99$ and $\mu_1 = \mu = 50, \mu_2 = \mu_3 = \cdots = \mu_{100} = 2\mu = 100$.
The values λ_i and μ_i are the parameters of the exponentially distributed interarrival and service times when there are i-many customers in the system, respectively **[+2]**. This means that, if there are i-many customers, the expected waiting time before a new customer arrives is λ_i^{-1} (alternatively, the expected interarrival rate is λ_i) and the expected service time is μ_i^{-1} (alternatively, the expected service rate is μ_i). There are $s = 2$ servers, each with parameter $\mu = 50$ and the capacity of the queueing system is $K = 100$. **[+1]**

b. **[3 marks]** We have that $c_n = \frac{\lambda_{n-1}\lambda_{n-2}\ldots\lambda_0}{\mu_n\mu_{n-1}\ldots\mu_1} = \frac{99}{50}\left(\frac{99}{100}\right)^{n-1}$ for $1 \le n \le 100$ **[+1]** and

$$p_0 = \left(1 + \sum_{n=1}^{100} c_n\right)^{-1} = \left(1 + \frac{99}{50}\sum_{n=1}^{100}\left(\frac{99}{100}\right)^{n-1}\right)^{-1}$$

$$= \left(1 + \frac{99}{50}\sum_{n=0}^{99}\left(\frac{99}{100}\right)^n\right)^{-1} = \left(1 + \frac{99}{50}\frac{\left(1 - \left(\frac{99}{100}\right)^{100}\right)}{1 - \frac{99}{100}}\right)^{-1}$$

$$= \left(1 + 2 \cdot 99 \cdot \left(1 - \left(\frac{99}{100}\right)^{100}\right)\right)^{-1}$$

$$= (1 + 2 \cdot 99 \cdot 0.633967 \ldots)^{-1} = (126.525596 \ldots)^{-1}$$

[+2]

6) **[3 Marks]** The steady state probabilities exist if $p_0 = (1 + \sum_{n=1}^{\infty} c_n)^{-1} > 0$ (otherwise $p_n = c_n p_0 = 0$ for every $n > 0$, but this is incompatible with the constraint $\sum_{n=0}^{\infty} p_n = 1$). This implies that we need to determine μ such that if $\sum_{n=1}^{\infty} c_n$ converges. In this case, $c_n = \frac{3^{\frac{n}{2}}}{\mu^n}$ if n is even and $c_n = \frac{3^{\frac{n+1}{2}}}{\mu^n}$ if n is odd. Therefore,

$$\sum_{n=1}^{\infty} c_n = \sum_{n=1}^{\infty} \frac{3^n}{\mu^{2n}} + \sum_{n=1}^{\infty} \frac{3^n}{\mu^{2n-1}} = (1 + \mu)\sum_{n=1}^{\infty} \frac{3^n}{\mu^{2n}} \; ,$$

which converges iff $\mu > \sqrt{3}$.

7) We note that $f(x) = x^2(x-1)^2$.
   a. **[2 Marks]** The derivative of f is $f'(x) = 2x(x-1)^2 + 2x^2(x-1) = 2x(x-1)(2x-1)$.
      The stationary points satisfy $f'(x) = 0$. This means they occur at $x = 0, 0.5, 1$
   b. **[3 Marks]** The second derivative of f is $f''(x) = 12x^2 - 12x + 2$.
      Therefore, $x_1 = -1 - \frac{f'(x_0)}{f''(x_0)} = -1 - \frac{-12}{26} = \frac{6}{13} - 1 = -\frac{7}{13} \cong -0.54$.
   c. **[3 Marks]** To compute the smallest optimal step size, we need to solve

$$\min_s g(s) = f(-1 + 12s) = (12s - 1)^2(12s - 2)^2$$

      Clearly, the minima are $s = \frac{1}{12}$ and $s = \frac{1}{6}$..Using $s = \frac{1}{12}$, we obtain $x_1 = -1 + \frac{12}{12} = 0$.

8) **[4 Marks]** After deriving the formulas $\nabla f(x) = 2A^TAx - 2A^Tb$ and $Hf(x) = 2A^TA$, we compute $x_1 = x_0 - Hf(x_0)^{-1}\nabla f(x_0) = x_0 - 0.5A^{-1}A^{-T}(2A^TAx_0 - 2A^Tb) = A^{-1}b$.

9) **[4 Marks]** We define the Lagrangian $L: \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$

$$L(x,u,p) = f(x,u) - p^T \left(\begin{pmatrix} 1+x^2 & x \\ x & 1 \end{pmatrix} u - \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right).$$

Then, $\frac{d}{dx} f(x,u) = \nabla_x L(x,u,p) = -p^T \begin{pmatrix} 2x & 1 \\ 1 & 0 \end{pmatrix} u$

where u solves the state constraint $\begin{pmatrix} 1+x^2 & x \\ x & 1 \end{pmatrix} u = \binom{1}{1}$ (note that $\begin{pmatrix} 1+x^2 & x \\ x & 1 \end{pmatrix}$ is invertible for any x) and p solves the adjoint equation is $\nabla_u L(x,u,p) = 0$, i.e., $\begin{pmatrix} 1+x^2 & x \\ x & 1 \end{pmatrix} p = \binom{1}{-\sin(u_2)}$.