**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. **[2 marks]** The function is a flow because for every edge , 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]**.  
       
     It’s value is . **[+1]**.
  2. **[2 marks]** The path is -augmenting **[+1]**. Indeed, its capacity is **[+1**] *.*
  3. **[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 is 4+6 = 10, the value of the maximal flow is not greater than 10. **[+1]**
  4. **[3 marks]**  First, we set . Then, in each step we identify the cut-set , identify an edge in with minimal weight, expand by adding a new node from this edge, and add the edge to the tree. Here is the generated sequence and the resulting graph.



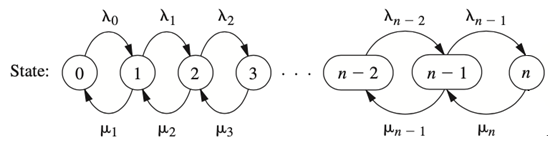
     4. 
  5. **[5 marks]**  The general form of the minimal spanning tree problem as a linear programming problem reads  
       
     In this case we have 5 edges, which we can label as follows:  
       
     For the second constraint, we need to consider systematically every full subgraph of (V,E) (note that we can ignore nonconnected subgraphs).  
     Full subgraphs with two nodes just lead to the constraints: .  
     Full subgraphs with three nodes lead to the constraints:  
       
     OAB

OAC   
OBC   
ABC   
  
The full subgraph with four nodes is equivalent to the original subgraph.

To sum up, the linear program reads:

where . **[+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]**

* 1. **[3 Marks]** Strategies 1 and 3 are both optimal for player 1, strategies 3 and 4 are both optimal for player 2, because. **[+1]**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]**
  2. **[4 Marks]** In class we saw that the optimal strategy of player 1 is characterised by the optimisation problem   
       
     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 .

1. Consider a random variable T with exponential distribution alpha.
   1. **[1 mark]** ,
   2. **[3 marks]** Since and are independent, **[+1]**The sum has Erlang distribution with parameters (3,1). Therefore, **[+2]**
2. 
   1. **[3 marks]** Draw the above picture with and .The values and are the parameters of the exponentially distributed interarrival and service times when there are -many customers in the system, respectively **[+2]**. This means that, if there are -many customers, the expected waiting time before a new customer arrives is (alternatively, the expected interarrival rate is ) and the expected service time is (alternatively, the expected service rate is ). There are servers, each with parameter and the capacity of the queueing system is . **[+1]**
   2. **[3 marks]** We have that for **[+1]** and

**[+2]**

1. **[3 Marks]** The steady state probabilities exist if (otherwise for every , but this is incompatible with the constraint ). This implies that we need to determine such that if converges. In this case, if is even and if is odd. Therefore,  
   which converges iff .
2. We note that .
   1. **[2 Marks]** The derivative of is The stationary points satisfy . This means they occur at
   2. **[3 Marks]** The second derivative of is Therefore, .
   3. **[3 Marks]** To compute the smallest optimal step size, we need to solve   
        
      Clearly, the minima are and ..Using , we obtain .
3. **[4 Marks]** After deriving the formulas and , we compute .
4. **[4 Marks]** We define the Lagrangian Then, where solves the state constraint (note that is invertible for any ) and solves the adjoint equation is , i.e.,