MAS210 Graph Theory Exercises 8 Solutions

Q1 *Let $K_{5,5}$ be the complete bipartite graph with bipartition $X = \{x_1, x_2, x_3, x_4, x_5\}$ and $Y = \{y_1, y_2, y_3, y_4, y_5\}$. Let N be the network obtained from $K_{5,5}$ by giving its edges the weights shown in the following table.*

	y_1	y_2	y_3	y_4	y_5
x_1	5	1	2	3	3
x_2	4	3	4	4	3
x_3	3	2	5	6	2
x_4	1	2	3	2	1
x_5	1	2	1	2	1

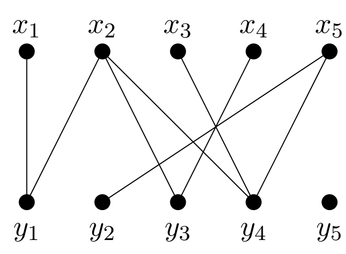
Use the Hungarian method to construct a maximum weight perfect matching and a minimum size feasible vertex labelling for N. Justify the facts that your perfect matching has maximum weight and your feasible vertex labelling has minimum size.

First iteration

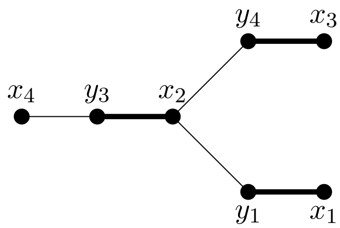
We first construct the feasible vertex labelling ℓ_1 below.

	y_1	y_2	y_3	y_4	y_5	
x_1	5	1	2	3	3	5
x_2	4	3	4	4	3	4
x_3	3	2	5	6	2	6
x_4	1	2	3	2	1	3
x_5	1	2	1	2	1	2
	0	0	0	0	0	ℓ_1

The equality subgraph $G = G(\ell_1)$ for ℓ_1 is shown below.



Apply König's Algorithm to G starting with the matching $M_1 = \{x_1y_1, x_2y_3, x_3y_4, x_5y_2\}$, which we choose greedily. We obtain the following M_1-alternating forest F rooted at the M_1-unsaturated vertex x_4.

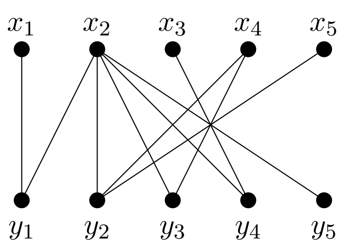


We deduce that M_1 is a maximum matching. Let $S = X \cap V(F) = \{x_4, x_2, x_3, x_1\}$. Then $\Gamma_G(S) = Y \cap V(F) = \{y_3, y_4, y_1\}$. This gives $\alpha = 1$ and we construct a new feasible vertex labelling ℓ_2 for G as below.

	y_1	y_2	y_3	y_4	y_5		
x_1	5	1	2	3	3	5	4
x_2	4	3	4	4	3	4	3
x_3	3	2	5	6	2	6	5
x_4	1	2	3	2	1	3	2
x_5	1	2	1	2	1	2	2
	0	0	0	0	0	ℓ_1	
	1	0	1	1	0		ℓ_2

Second iteration

We construct the equality subgraph $G = G(\ell_2)$ as below.



We apply König's Algorithm to G starting with the matching M_1 from iteration 1 and construct the perfect matching $M = \{x_1y_1, x_2y_5, x_3y_4, x_4y_3, x_5y_2\}$. We have $w(M) = 19 = size(\ell_2)$. Thus M is a maximum weight perfect matching in N, and ℓ_2 is a minimum size feasible vertex labelling for N.

Q2 *Let* $\ell_1, \ell_2, \ldots$ *be a sequence of vertex labellings constructed when the Hungarian method is applied to a network* N*. Prove that the following statements are valid.*
(a) Each of the vertex labellings ℓ_i*,* $i \geq 1$ *are feasible.*
(b) $size(\ell_{i+1}) < size(\ell_i)$ *for all* $i \geq 1$*.*

(a) We use induction on i. The fact that ℓ_1 is feasible follows immediately from the definition of ℓ_1. Assume, inductively, that ℓ_i is feasible for some $i \geq 1$. Suppose that ℓ_{i+1} is not a feasible vertex labelling of N. Then we have $\ell_{i+1}(x) + \ell_{i+1}(y) < w(xy)$ for some $x \in X$ and $y \in Y$. Since ℓ_i is a feasible vertex labelling of N, we must have $x \in S$ and $y \in Y - \Gamma_G(S)$. But then the definition of α implies that $\ell_i(x) + \ell_i(y) - w(xy) \geq \alpha$ and hence $\ell_{i+1}(x) + \ell_{i+1}(y) - w(xy) \geq 0$. Thus ℓ_{i+1} is a feasible vertex labelling of N.
(b) Since $\alpha > 0$ and $|S| > |\Gamma_G(S)|$, we have $size(\ell_{i+1}) = size(\ell_i) - \alpha(|S| - |\Gamma_G(S)|) < size(\ell_i)$.

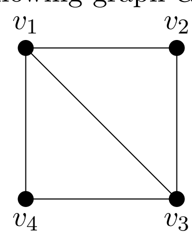
Q3*(a) Determine the number of different perfect matchings in* $K_{m,m}$*.*
(b) Deduce that, if N *is the network obtained from* $K_{m,m}$ *by giving its edges integer weights, then the 'brute force' algorithm of enumerating all perfect matchings and choosing one with the largest weight is not a polynomial algorithm.*

(a) Let $M = \{x_1y_{i_1}, x_2y_{i_2}, \ldots, x_my_{i_m}\}$ be a perfect matching in $K_{m,m}$. We have m different choices for the vertex y_{i_1}. Once we have chosen y_{i_1}, we have $m-1$ different choices for the vertex y_{i_2}. Once we have chosen y_{i_1} and y_{i_2}, we have $m-2$ different choices for the vertex y_{i_3}, and so on. It follows that the number of perfect matchings in $K_{m,m}$ is $m \times (m-1) \times (m-2) \ldots \times 2 = m!$.
(b) The time taken to enumerate all perfect matchings in N and choose one with the largest weight will be at least $c \times m!$, where c is a constant representing the time it takes to calculate the weight of any given perfect matching. Since $c \times m! > m^k$ for any fixed k, when m is large enough, the brute force algorithm is not a polynomial algorithm.

Q4 *Construct a graph G which is not bipartite and still satisfies*

$$match(G) = cov(G).$$

Consider the following graph G



Then $M = \{v_1v_2, v_3v_4\}$ in G and $U = \{v_1, v_3\}$ is a cover for G. Since $|M| = 2 = |U|$ we have $match(G) = 2 = cov(G)$.