MATH 3330 — Applied Graph Theory

Assignment 9 – Solutions

1. Suppose a tourist wants to travel from LA to New York city, in stages. In order to plan the most touristy route, a travel agent has identified a large number of cities that could be used as stopping places along the way, and assessed a number of routes to travel from city to city, with an assessment of "benefit" (scale 1-10) assigned to each route. For example: LA-Las Vegas, benefit 8. LA-San Fransisco, benefit 7. San Fransisco-Las Vegas, benefit 5, etc. The aim of the travel agent is to find a route from LA to New York that will maximize the benefit for the tourist.

   (a) Formulate this problem as a shortest path problem.

   Form a graph with all cities as vertices. Add edges for each proposed route for which a benefit is assigned, with cost equal to minus benefit. The shortest path (i.e. path of lowest cost) from LA to NYC in this graph will be the route of maximum benefit.

   (b) Why can Dijkstra's algorithm not be used to solve this problem?

   Dijkstra's algorithm only works for non-negative costs.

   (c) Which algorithm can be used? Describe the algorithm.

   Floyd-Warshall can be used. Description: Start with a matrix $M^{(0)}$, where $M_{ij}^{(0)}$ equals minus the benefit of going from city i to city j, or infinity if no benefit was assigned. For $k = 1, \ldots, n$, form $M^{(k)}$ from $M^{(k-1)}$ as follows:

   $$M_{ij}^{(k)} = \min\{M_{ij}^{(k-1)}, M_{ik}^{(k-1)} + M_{kj}^{(k-1)}\}.$$

   The last matrix, $M^{(n)}$, will contain the cost of the shortest paths between each pair of vertices.

   To find the path, maintain a matrix E. Initialize with all entries set to zero. Whenever the minimum in the equation above is achieved by the second term, set $E_{ij} = k$. Trace back throught E to find all intermediate vertices that make up the path.

   (d) What happens if the graph formed in (a) contains a directed cycle?

   This can be detected if one of the elements on the diagonal of $M^{(k)}$ becomes negative. For this problem, that would mean that there *is a cycle of positive benefit, so in principle, the traveller can keep travelling around this cycle to increase her benefit. (So in order to make the shortest path approach work, positive benefit cycles should be avoided.)*

2. Solve the following shortest path problem with the Floyd-Warshall algorithm. The table below represents the weights of the edges. If the entry is $-$, no edge exists. Using the matrix E, find the shortest (cheapest) path from v_1 to v_3.

	v_1	v_2	v_3	v_4
v_1	-	2	3	-
v_2	3	-	-3	4
v_3	-	5	-	3
v_4	1	1	1	-

$$M^{(0)} = \begin{bmatrix} \infty & 2 & 3 & \infty \\ 3 & \infty & -3 & 4 \\ \infty & 5 & \infty & 3 \\ 1 & 1 & 1 & \infty \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M^{(1)} = \begin{bmatrix} \infty & 2 & 3 & \infty \\ 3 & 5 & -3 & 4 \\ \infty & 5 & \infty & 3 \\ 1 & 1 & 1 & \infty \end{bmatrix} \quad E = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$M^{(2)} = \begin{bmatrix} 5 & 2 & -1 & 6 \\ 3 & 5 & -3 & 4 \\ 7 & 5 & 2 & 3 \\ 1 & 1 & -2 & 5 \end{bmatrix} \quad E = \begin{bmatrix} 2 & 0 & 2 & 2 \\ 0 & 1 & 0 & 0 \\ 2 & 0 & 2 & 0 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

$$M^{(3)} = \begin{bmatrix} 5 & 2 & -1 & 2 \\ 3 & 2 & -3 & 0 \\ 7 & 5 & 2 & 3 \\ 1 & 1 & -2 & 1 \end{bmatrix} \quad E = \begin{bmatrix} 2 & 0 & 2 & 3 \\ 0 & 3 & 0 & 3 \\ 2 & 0 & 2 & 0 \\ 0 & 0 & 2 & 3 \end{bmatrix}$$

$$M^{(4)} = \begin{bmatrix} 3 & 2 & -1 & 2 \\ 1 & 1 & -3 & 0 \\ 4 & 4 & 1 & 3 \\ 1 & 1 & -2 & 1 \end{bmatrix} \qquad E = \begin{bmatrix} 4 & 0 & 2 & 3 \\ 4 & 4 & 0 & 3 \\ 4 & 4 & 4 & 0 \\ 0 & 0 & 2 & 3 \end{bmatrix}$$

Cheapest path from v_1 to v_3 has cost -1. Reconstruct path from E: $v_1 - v_2 - v_3$.

3. Describe the problem of finding a maximum weight matching (find a matching so that the sum of the weights of the edges in the matching is maximized) as a minimum cost flow problem. Carefully describe the transformation.

   Suppose a bipartite graph G is given with partite sets X and Y. Form an $s - t$ network with edge capacities and edge costs as follows:

   - *Add a vertex s, and edges from s to every vertex in X, with capacity 1 and cost 0.*
   - *Add a vertex t, and edges from every vertex in Y to t, with capacity 1 and cost 0.*
   - *For all the edges in the graph, direct them from X to Y, keep the cost the same, and give them capacity ∞.*
   - *Any (integer) max flow of value k in this graph corresponds to a size k matching in G, as seen before (edges in matching are those $X - Y$ edges in the network that have nonzero flow).*

4. Determine the maximum flow from s to t of minimum cost in the following network. *Show your work.*

   Method of doing this: start with zero flow. In each iteration, form the auxiliary graph as in the max flow algorithm. For each forward edge, assign its original cost, for each backward edge, give it the negative of its original cost. Find the cheapest path from s to t in this graph, and use it to augment the flow. For this example, min cost max flow has value 11, cost 249

5. Suppose cars have to be transported from 3 manufacturing plants w_1, w_2 and w_3 to dealerships s_1, s_2, and s_3. The number of cars available at each plant is 900, 1500 and 700, respectively. The number of cars required at each dealership is 500, 1300 and 700. The cost of shipping one car from plant to dealership is given in the table below. The problem is to find the cheapest way of shipping cars from manufacturers to dealers, so that as much of the demand for cars is satisfied (however, you may not deliver more cars than the dealership requested, because of storage problems). Describe a way to solve this problem using minimum cost maximum flow.

   Form a graph as follows. Make vertices for each of the three plants and each of the dealerships, and a vertex s and vertex t. Add edges from s to each plant vertex, with capacity equal to the number of cars available at that plant: $cap(sw_1) = 900$, $cap(sw_2) = 1500$, $cap(sw_3) = 700$. *Cost of each edge is zero. Add edges from each plant to each dealership, with capacity infinity, and cost equal to the cost of transporting one car from plant to dealership. Add edges from each dealership to t with capacity equal to the demand at the dealership, and costs zero.*