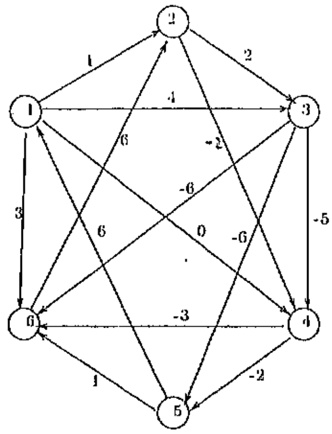
Network Flows - 14.2.03

Rafi Hassin

1. Consider the network in the following figure:
   (i) Use successive approximations to compute the shortest path tree rooted at node 1.
   (ii) Compute an equivalent network with non-negative arc lengths.



2. A network is given with bounds (l_{ij}, u_{ij}). Suppose that there is no feasible circulation with respect to these bounds. It is possible however to change the bounds. The cost of increasing u_{ij} is q_{ij} per unit of change. Similarly, the cost of deceasing l_{ij} is p_{ij} per each unit of change. The problem is to make changes of minimum cost so that the network with the new bounds will have a feasible circulation.
   (i) Formulate a linear program for this problem.
   (ii) Show that it is equivalent to a flow problem.

3. Given a min cost flow problem with demand v and integer capacities u_{ij}. (The lower bounds are 0). Suppose that x is an optimal solution. Suppose now that for one arc e, it turns out that the true bound is $u_e + 1$ (and not u_e).
   (i) Show how to check whether x is still optimal.
   (ii) If x is not optimal now, how can we find efficiently the new optimal solution. In particular, is it sufficient to change the flow along one cycle?

4. (i) Describe how to use the algorithm for checking the existence of a perfect matching, to compute a matching with a maximum number of edges.
   (ii) Describe how to use the algorithm for a minimum cost perfect matching, to compute a matching with maximum cost.

5. (i) Suggest an efficient algorithm to check whether the minimum s-t cut is unique.
   (ii) Which minimum cut will the labeling algorithm find if the minimum cut is not unique?

6. Suppose we want to compute a shortest $s - t$ path in an undirected graph $G = (V, E)$ with non-negative edge lengths $a_{i,j}$ $(i, j) \in E$. Suppose that we executed several iterations of Dijkstra's algorithm and stopped after computing for every node i in a subset $S \subset V$ the shortest $s - i$ path, and its length $u_{s,i}$. Similarly, using t as the source node, we executed the algorithm for several iterations and stopped after computing the shortest $t - j$ path, of length $u_{t,j} = u_{j,t}$ for every $j \in T$. Suppose that $S \cap T \neq \emptyset$.
   (i) Prove that there exists a shortest $s - t$ path that uses only nodes from $S \cup T$.
   (ii) Show how to compute $u_{s,t}$ using the above information. Prove that your answer is correct.