Discrete Optimization 2010

Hand-in Assignments (due Oct. 11, 2010)

Marc Uetz (m.uetz@utwente.nl), Applied Mathematics, University of Twente

The reading for Maximum Flows is as follows. Chapters 6.1–6.5, 6.7., and Chapters 7.1, 7.2. Moreover, Chapter 7.4 contains an $O(n^2m)$ implementation of the shortest augmenting path algorithm for Maximum Flows[1]. All chapter numbers refer to [1].

1. (4 points) Given is an undirected, connected graph $G = (V, E)$ with edge lengths $c_e > 0$, $e \in E$. We use Dijkstras algorithm to compute the shortest paths to all nodes $v \in V$, starting at some node s. Denote by $P_v \subseteq E$ the set of edges on a shortest (s, v)-path as computed by Dijkstra, and define $T_D := \bigcup_{v \in V} P_v$. Moreover, let T be an arbitrary minimum spanning tree of G.

   Prove that T and T_D always have at least one edge in common, i.e., $T_D \cap T \neq \emptyset$.

2. (5 points) Consider the feasible flow problem discussed in Application 6.1 (page 169) of [1]. Now let us suppose that, instead of ordinary flow capacity constraints $0 \leq x_{ij} \leq u_{ij}$, we have flow capacity constraints $\ell_{ij} \leq x_{ij} \leq u_{ij}$ for some nonnegative lower bound ℓ_{ij} on the arc flows.

   Prove that this feasibility problem can be solved by application of a standard maximum (s, t)-flow algorithm in an accordingly transformed network (with capacity constraints $0 \leq x_{ij} \leq u_{ij}$).

   (Hint: First get rid of the lower bounds on arc flows by modifying the balance constraints and arc capacities accordingly. Then get rid of all nodes with balance constraints $\neq 0$. Finally, argue how one maximum flow computation in the transformed network can be used to solve the original feasible flow problem.)

3. (5 points) Given is a directed graph $G = (V, A)$ with integer arc capacities $0 \leq x_{ij} \leq u_{ij}$, and a maximum (s, t)-flow x_{ij}, for certain $s, t \in V$.

   (i) Suggest an algorithm to compute a minimum (s, t)-cut, and analyze its computation time.

   (ii) Suppose the given maximum (s, t)-flow x_{ij} is not integral (i.e., some flow values x_{ij} are fractional). Suggest an algorithm to convert this flow into an integral one.

4. (4 points) Given is a directed graph $G = (V, A)$ with arc capacities $0 \leq x_{ij} \leq u_{ij}$, and $s, t \in V$. Give a proof or a counterexample.

   (i) If all arcs have different arc capacities u_{ij}, there is a unique minimum (s, t) cut.

   (ii) If all arc capacities are scaled by some $\delta > 0$, the minimum cut(s) remain unchanged.

References

[1] Ahuja, R. K., T. L. Magnanti, and J. B. Orlin, *Network Flows*, Prentice Hall, 1993.

[1] Note that the definition of distance labels is different in the sense that they measure the minimum distance from any node to the sink t, while in the lecture we used distance labels that measure the minimum distance from the source s to all other nodes. Moreover, the implementation differs from the presentation in the lecture in the following sense: At any iteration, not a complete BFS search is used to compute the distance labels, but the distance labels are updated in each step directly (therefore requiring a proof that this update produces correct distance labels!). The improvement in the computation time comes from the fact that it is explicitly exploited that the distance labels can only increase.