6.854J / 18.415J Advanced Algorithms
Fall 2008

18.415/6.854 Advanced Algorithms

Problem Set 3

1. Consider the following optimization problem:

   Given $c \in \mathbb{R}^n$, $c \geq 0$, n even, find

$$\min\{c^T x : \begin{array}{ll} \sum_{i \in S} x_i \geq 1 & \forall S \subset \{1, \ldots, n\}, |S| = \frac{n}{2}, \\ x_j \geq 0 & \forall j\}. \end{array}$$

   In class, it was shown that this can be solved by the ellipsoid method because there is an efficient separation algorithm. However, this problem has a more straightforward solution.

   Develop an algorithm which finds the optimum in $O(n \log n)$ time. Prove its correctness.

2. Fill a gap in the analysis of the interior point algorithm:

   Suppose that (x, y, s) is a feasible vector, i.e. $x > 0$, $s > 0$,

$$Ax = b,$$

$$A^T y + s = c$$

   and we perform one Newton step by solving for $\Delta x, \Delta y, \Delta s$:

$$A \Delta x = 0$$

$$A^T \Delta y + \Delta s = 0$$

$$\forall j; \;\; x_j s_j + \Delta x_j s_j + x_j \Delta s_j = \mu$$

   where $\mu > 0$. The proximity function is defined as

$$\sigma(x, s, \mu) = \sqrt{\sum_j (\frac{x_j s_j}{\mu} - 1)^2}.$$

   Prove that if

$$\sigma(x + \Delta x, s + \Delta s, \mu) < 1$$

   then $(x + \Delta x, y + \Delta y, s + \Delta s)$ is a feasible vector for $Ax = b, x > 0$ and $A^T y + s = c, s > 0$.

3. Given a directed graph $G = (V, E)$ and two vertices s and t, we would like to find the maximum number of edge-disjoint paths between s and t (two paths are edge-disjoint if they don't share an edge). Denote the number of vertices by n and the number of edges by m.

   (a) Argue that this problem can be solved as a maximum flow problem with unit capacities. Explain.

   (b) Consider now the maximum flow problem on directed graphs $G = (V, E)$ with unit capacity edges (although some of the questions below would also apply to the more general case).

   Given a feasible flow f, we can construct the *residual network* $G_f = (V, E_f)$ where

   $$E_f = \{(i, j) : ((i, j) \in E \ \&\ f_{ij} < u_{ij}) \text{ or } ((j, i) \in E \ \&\ f_{ji} > 0)\}.$$

   The residual capacity of an edge $(i, j) \in E_f$ is equal to $u_{ij} - f_{ij}$ or to f_{ji} depending on the case above. Since we are dealing with the unit capacity case, all the u_{ij}'s are 1 and therefore for $0-1$ flows f (i.e. flows for which the value on any edge is 0 or 1), all residual capacities will be 1.

   We define the distance of a vertex $l_f(v)$ as the length of the shortest path from s to v in E_f (∞ for vertices which are not reachable from s in E_f). Further, define the *levelled residual network* as

   $$E_f^l = \{(i, j) \in E_f : l_f(j) = l_f(i) + 1\}$$

   and a *saturating flow* g in E_f^l as a flow in E_f^l (with capacities being the residual capacities) such that every directed $s-t$ path in E_f^l has at least one saturated edge (i.e. an edge whose flow equals the residual capacity).

   For a unit capacity graph and a given $0-1$ flow f, show how we can find the levelled residual network and a saturating flow in $O(m)$ time.

   (c) Prove that if the levelled residual network has no path from s to t ($l_f(t) = \infty$), then the flow f is maximum.

   (d) For a flow f, define

   $$d(f) = l_f(t)$$

   (the distance from s to t in the residual network). Prove that if g is a saturating flow for f then

   $$d(f + g) > d(f),$$

   where $f + g$ denotes the flow obtained from f by either increasing the flow f_{ij} by g_{ij} or decreasing the flow f_{ji} by g_{ij} for every edge $(i, j) \in G_f$.

(e) Prove that if f is a feasible $0-1$ flow with distance $d = d(f)$ and f^* is an optimum flow, then

$$\text{value}(f^*) \leq \text{value}(f) + \frac{m}{d}$$

and also

$$\text{value}(f^*) \leq \text{value}(f) + \frac{n^2}{d^2}.$$

(f) Design a maximum flow algorithm (for unit capacities) which proceeds by finding a saturating flow repeatedly. Try to optimize its running time. Using the observations above, you should achieve a running time bounded by $O(\min(mn^{2/3}, m^{3/2}))$.

(g) Can we now justify that, for $0-1$ capacities, there is always an optimum flow that takes values 0 or 1 on every edge?