ORIE 6330 Network Flows November 1, 2012

Problem Set 5

Due Date: November 15, 2012

1. In class, when considering the minimum-cost circulation problem we showed that we could cancel all negative-cost cycles in $O(m^2 n^2 \log(nC))$ time. When considering the generalized circulation problem, we can also cancel all flow generating cycles by using this algorithm. We do this by considering costs $c(i,j) = -\log \gamma(i,j)$ and cancelling negative-cost cycles. However, the costs $c(i,j)$ are no longer integral, which was an assumption needed to prove the running time above. Assume that the gains $\gamma(i,j)$ are ratios of integers that are bounded in absolute value by B. Show that in this case, the running time of the cycle cancelling algorithm we used in class is $O(m^2 n^3 \log(nB))$.

2. In the minimum-cost generalized circulation problem, we are given costs $c(i,j)$ in addition to gains $\gamma(i,j)$ and capacities $u(i,j)$. The goal is to find a *generalized circulation* f that minimizes $\sum_{(i,j)\in A} c(i,j)f(i,j)$. A generalized circulation is a generalized pseudoflow that has $e_f(i) = 0$ for all $i \in V$.

   For the generalized flow problem we considered in class, we showed that the flow is maximum if and only if there are no generalized augmenting paths. In the case of the minimum-cost generalized circulation problems, the objects of interest are *unit gain cycles* and *bicycles*. A unit gain cycle C has $\gamma(C) = 1$. A bicycle has a flow-generating cycle C_1 connected by a path (possibly trivial) to a flow-absorbing cycle C_2.

   Prove that a minimum-cost generalized circulation f is optimal if and only if there are no negative-cost unit gain cycles and no negative-cost bicycles in the residual graph G_f.

3. In Wallacher's algorithm, we cancel cycles that tradeoff cost versus residual capacity. In this exercise, we consider another way of doing this that uses some ideas from the capacity scaling algorithm discussed in class. One way to improve the situation is to make sure that every iteration of cycle canceling considers only arcs with "large enough" residual capacity. Given a circulation f, potentials p and a parameter Δ, let $A_f(\Delta) = \{(i,j) \in A_f : u_f(i,j) \geq \Delta\}$. Call an arc $(i,j) \in A_f$ *admissible* if $c_p(i,j) < 0$ and Δ-*admissible* if (i,j) is admissible and $(i,j) \in A_f(\Delta)$. Let's say that cycle Γ is a Δ-*cycle* if $\Gamma \subseteq A_f(\Delta)$, $c_p(i,j) \leq 0$ for all $(i,j) \in \Gamma$, and $c_p(i,j) < 0$ for some $(i,j) \in \Gamma$. Note that this implies $c_p(\Gamma) = c(\Gamma) < 0$. We give a procedure $\texttt{Find}\Delta\texttt{Cycle}(p,i,j)$ that takes as input node potentials p and some Δ-admissible arc (i,j), and uses them to find a Δ-cycle Γ.

   We can then use this subroutine in Algorithm 1, $\texttt{Cancel}\Delta\texttt{Cycles}$.

   (a) Prove that the subroutine $\texttt{Find}\Delta\texttt{Cycle}$ does not create any new Δ-admissible arcs.

   (b) Prove that if the subroutine $\texttt{Find}\Delta\texttt{Cycle}$ returns a cycle, it is a Δ-cycle.

Let S be the set of nodes reachable from j via arcs in $A_f(\Delta) - \{(j,i)\}$
if $i \notin S$ **then**

$$p(k) \leftarrow \begin{cases} p(k) + c_p(i,j) & \text{if } k \in S \\ p(k) & \text{otherwise} \end{cases}$$

else

Compute shortest j-k path distance $\tilde{p}(k)$ using arcs in $A_f(\Delta)$ and costs $\max(0, c_p(i,j))$

$\tilde{p}_{\max} = \max_{k \in S} \tilde{p}(k)$

$$p(k) \leftarrow \begin{cases} p(k) + \tilde{p}(k) - \tilde{p}_{\max} & k \in S \\ p(k) & \text{otherwise} \end{cases}$$

if $c_p(i,j) < 0$ **then**

Let $\Gamma = \{(i,j)\}$ + shortest path from j to i

return Γ, p'

Procedure FindΔCycle(p, i, j)

```
f ← 0
p ← 0
Δ ← 2^⌈log U⌉
while Δ ≥ 1 do
    while there is a Δ-admissible arc (i, j) do
        (Γ, p) ← FindΔCycle (p, i, j)
        if Γ ≠ ∅ then
            Cancel Γ
            Update f
    Δ ← Δ/2
return f
```

Algorithm 1: Another cycle-canceling algorithm, `Cancel∆Cycles`.

(c) Prove that either `Find∆Cycle` returns a cycle containing (i,j) or makes (i,j) inadmissible.

(d) Prove that at the start of the inner while loop, $u_f(i,j) < 2\Delta$ for each admissible arc (i,j), and that this remains true through the execution of the while loop.

(e) Prove that in each iteration of the inner while loop, the number of Δ-admissible arcs strictly decreases.

(f) Prove that if the algorithm terminates, it correctly returns a minimum-cost circulation.

(g) Suppose that we have an $O(m + n \log n)$ time algorithm for computing shortest paths in graphs with nonnegative edge lengths. Prove that the algorithm runs in time $O((m \log U)(m + n \log n))$.