The Bellman-Ford Algorithm

Detecting Negative Cycles

Algorithms: Design and Analysis, Part II

Checking for a Negative Cycle

Question: What if the input graph G has a negative cycle?
[want algorithm to report this fact]

Claim:
G has no negative-cost cycle (that is reachable from s) $\Longleftrightarrow$ in the (extended) Bellman-Ford algorithm, $A[n-1,v] = A[n,v]$ for all $v \in V$.

Consequence: Can check for a negative cycle just by running Bellman-Ford for one extra iteration (running time still $O(mn)$).

Proof of Claim

($\Rightarrow$) already proved in correctness of Bellman-Ford

($\Leftarrow$) Assume $A[n-1,v] = A[n,v]$ for all $v \in V$. (assume also these are finite ($< +\infty$))

Let $d(v)$ denote the common value of $A[n-1,v]$ and $A[n,v]$.

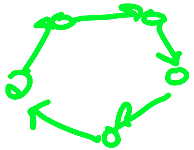
Recall algorithm: $\underbrace{A[n,v]}_{d(v)} = \min \begin{cases} A[n-1,v] \\ \min_{(w,v)\in E} \{ \underbrace{A[n-1,w]}_{d(w)} + c_{wv} \} \end{cases}$

Thus: $d(v) \le d(w) + c_{wv}$ for all edges $(w,v) \in E$

Equivalently: $d(v) - d(w) \le c_{wv}$

Now: consider an arbitrary cycle C.

$$\sum_{(w,v)\in C} c_{wv} \ge \sum_{(w,v)\in C} (d(w) - d(v)) = 0.$$



QED!

Tim Roughgarden