6.854J / 18.415J Advanced Algorithms
Fall 2008

18.415/6.854 Advanced Algorithms October 15, 2001

Lecture 9

Lecturer: Michel X. Goemans *Scribe: Jasper Lin*

1 Min Cost Circulation Problem

Given a directed graph $G = (V, E)$, an anti-symmetric[1] cost function $c : E \to \mathbb{Z}$, and a capacity function $u : E \to \mathbb{Z}$.

Definition 1 *A flow is a mapping $f : E \to \mathbb{R}$ that is anti-symmetric[1] and $f(v, w) \leq u(v, w)$, $\forall (v, w) \in E$.*

Definition 2 *The cost of a flow f is defined to be*

$$c \cdot f = \sum_{(v,w) \in E} c(v, w) f(v, w)$$

Definition 3 *A circulation is a flow f such that*

$$\sum_{w} f(v, w) = 0, \ \forall v \in V$$

The min cost circulation problem is to find a circulation of minimum cost.

1.1 Special Cases of Min Cost Circulation Problem

The min cost circulation problem is actually fairly general. In includes the following important problems as special cases:

- Max-Flow
- Min Cost Flow
- Min Weighted Bipartite Matching
- Shortest Path
  Given a directed graph $G = (V, E)$, a function $l : E \to \mathbb{R}$ that specifies the lengths of the edges, and a fixed vertex $s \in V$, find the shortest simple[2] path from s to t, for every vertex $t \in V$. If the length of edges are nonnegative numbers, then Dijkstra's Algorithm may be used. Dijkstra's Algorithm has different running times depending on how the priority queue is implemented. Using a Fibonacci Heap, the running time is $O(m + n \log n)$, where $m = |E|$ and $n = |V|$. In general, if l can be negative, the problem is NP-complete. However, it is common to consider the case when there is no negative length cycles. Then the Bellman-Ford Algorithm can be used. This algorithm either detects that there is a negative length cycle or computes, in $O(mn)$, the length of the shortest path between s and t for every vertex $t \in V$.

[1] A function $f(x, y)$ is anti-symmetric iff $f(x, y) = -f(y, x)$.
[2] A path is simple if it visit each vertex at most once.

Remark 1 *The algorithms mentioned here do not in general apply directly to undirected graphs as one would think. The naive approach would be to turn every edge in the undirected graph into two edges going both ways in the digraph. This approach works if the lengths of the edges are nonnegative numbers. However, if an undirected graph has an edge with a negative length, doubling it would create a negative length cycle where one might not have existed in the first place. As it turns out, in this case, the problem in undirected graphs is much more difficult than the digraph shortest path problem and some underlying assumptions that we make about digraphs do not hold for undirected graphs. There is an algorithm for this problem using non-bipartite matching algorithm.*

1.2 Strongly Polynomial Running Time

Definition 4 *The running time of an algorithm is strongly polynomial iff*

- *the number of operations required is bounded by a polynomial of the number of data items in the input, and not the size of the numerical values in the input.*
- *all arithmetic operations are performed on numbers whose size is polynomial in input size.*

If the running time of an algorithm is polynomial but not strongly polynomial, then we say that it is weakly polynomial.

Our goal is to search for a strongly polynomial algorithm to solve the min cost circulation problem. Not all polynomial time algorithms are strongly polynomial. The algorithms for Linear Programming and some versions of Dijkstra's Algorithm are not strongly polynomial in running time. For example, using different priority queues, both $O(n \log\log L)$ and $O(m + n\sqrt{\log L})$ are possible, where L is the maximum length in the graph.

1.3 Searching for an Algorithm

Our first goal is to devise a check that given a flow f determines whether or not f is optimal.

Definition 5 *Given a graph $G = (V, E)$ and a circulation f, we define the residual graph of f to be $G_f = (V, E_f)$, where $E_f = \{(v, w) \in E | f(v, w) < u(v, w)\}$, and the capacity function is replaced by the residual capacity function $u_f(u, v) = u(u, v) - f(u, v)$.*

The important property of the residual graph is that if g is a circulation in the residual graph G_f, then $f + g$ is a valid circulation in G, since $f + g \leq f + u - f = u$.

A vertex potential p is a real-valued function defined on the set of vertices ($p : V \to \mathbb{R}$). Corresponding to a potential p, we can define a reduced cost function $c_p(v, w) = c(v, w) + p(v) - p(w)$.

Lemma 1 *For every directed cycle C, $c_p(C) = c(C)$, where the notation $c_p(C) = \sum_C c_p(v, w)$.*

Proof of Lemma 1: $c_p(C) = \sum_C c_p(v, w) = \sum_C (c(v, w) + p(v) - p(w)) = \sum_C c(v, w) + \sum_C (p(v) - p(w)) = c(C)$, since every $v \in C$ appears again as a $w \in C$ the same amount of times since C is a cycle. □

Lemma 2 *For every circulation f and potential p,*

$$c \cdot f = \sum_{(v,w) \in E} c(v, w) \cdot f(v, w) = c_p \cdot f$$

Proof of Lemma 2: One way is to show that a circulation can be decomposed into a sum of many cycles, and then use Lemma 1. Here we give a direct proof:

$$c_p \cdot f = \sum_{(v,w)\in E} c_p(v,w) \cdot f(v,w) = \sum_{(v,w)\in E} (c(v,w) + p(v) - p(w)) f(v,w)$$

$$= c \cdot f + \sum_{v\in V} p(v) \cdot \sum_{(v,w)\in E} f(v,w) - \sum_{w\in V} p(w) \cdot \sum_{(v,w)\in E} f(v,w)$$

$$= c \cdot f, \text{ since } \sum_{(v,w)\in E} f(v,w) = 0$$

□

Theorem 3 *(Optimality Conditions) Let f be a circulation. The following are equivalent:*

(i) *f is a min-cost circulation*

(ii) *G_f has no directed cycles of negative cost*

(iii) *There is a potential p such that for every $(v,w) \in E_f$, $c_p(v,w) \geq 0$.*

Proof of Theorem 3:

($\neg i \to \neg iii$) Assume there is a circulation f^* with $c \cdot f^* < c \cdot f$ and a potential p with $c_p(v,w) \geq 0$ for every $(v,w) \in E_f$. By Lemma 2, $c_p \cdot f^* < c_p \cdot f$. Then

$$0 > c_p \cdot f^* - c_p \cdot f = \sum_{(v,w)\in E} c_p(v,w) \cdot (f^*(v,w) - f(v,w)) = 2 \sum_{(v,w)\in E_f} c_p(v,w) \cdot (f^*(v,w) - f(v,w))$$

However, $(v,w) \in E_f$ implies that $f^*(v,w) > f(v,w)$. Therefore, $2\sum_{(v,w)\in E_f} c_p(v,w) \cdot (f^*(v,w) - f(v,w)) \geq 0$, which is a contradiction.

($i \to ii$) Suppose f is of min-cost and there exists a cycle C in G_f such that $c(C) < 0$. Form f', where $f'(u,v) = f(u,v) + \epsilon$ when $(u,v) \in C$, $f'(u,v) = f(u,v) - \epsilon$ when $(v,w) \in C$, $f'(u,v) = f(u,v)$ otherwise, and $\epsilon = \min_{e\in C} c_f(e) > 0$. Therefore, $c(f') = c(f) + 2\epsilon \cdot c(C) < c(f)$, a contradiction.

($ii \to iii$) Will be proved next lecture.

□

Remark 2 *We now have an algorithm for finding a min-cost circulation:*

1. *Construct the residual graph.*
2. *Check if there is any negative cost cycles. If none, end.*
3. *If one is found, push flow and repeat.*

Step 3 can be done in many different ways. They key is to choose intelligently to reduce the number of iterations necessary.