A SINGLE SOURCE SHORTEST PATH ALGORITHM FOR GRAPHS WITH SEPARATORS

K. Mehlhorn B. H. Schmidt

Fachbereich 10 - Angewandte
Mathematik und Informatik
Universität des Saarlandes
6600 Saarbrücken/West Germany

Abstract: We show how to solve a single source shortest path problem on a planar network in time $O(n^{3/2}\log n)$. The algorithm works for arbitrary edge weights (positive and negative) and is based on the planar separator theorem. More generally, the algorithm works in time $O(n^{a+b}\log n + n^{3a} + n^{d})$ on graphs $G=(V, E)$ which have a separator of size n^{a}, have at most n^{b} edges and where the separator can be found in time n^{d}.

1. Introduction

A network $N = (V, E, C)$ is a directed graph $G = (V, E)$ (where $|V| = n$, $|E| = e$) together with a real-valued function $C : E \to \mathbb{R}$. The length of a path $w = v_1 \ldots v_m$ is defined

$$l(w) = \sum_{i=1}^{m-1} C(v_i, v_{i+1}).$$

Let us denote the minimal length of any path from x to y (with respect to $W \subset V$) by

$$\mathrm{Min}(x,y;W) = \inf \{ l(w) \; ; \; w=v_1 \ldots v_n \text{ is a path from x to y and for } 1 \leq i \leq n \text{ we have } v_i \in W \}.$$

The single source shortest path problem is defined by

(SSSP): Let $N = (V, E, C)$ be a directed network and let s be a designated vertex in V.
Compute $\mathrm{Min}(s,x;V)$ for every vertex $x \in V$.

The most popular solution to SSSP is the Bellman/Ford algorithm. It solves SSSP in time $O(n \cdot e)$.

A more efficient solution is known if edge weights are non-negative, i.e. $C(x,y) \geq 0$ for all edges (x,y). In this case Dijkstra's algorithm runs in time $O(\min(n^2, e \cdot \log n))$. Finally we should mention that the all-pair shortest path problem, i.e. compute $\mathrm{Min}(x,y;V)$ for all $x,y \in V$, can be solved in time $O(\min(n^3, n \cdot e \cdot \log n)$; here n^3 comes from Kleene's algorithm and $n \cdot e \cdot \log n$ comes from an observation of Edmonds/Karp. They have shown that having solved SSSP once for a single source s one can transform the network N into a network $N' = (V, E, C')$ such that N' has non-negative weights, yet the same shortest paths as N. Hence the remaining n-1 SSSP's can be solved for a cost of $O(n \cdot e \cdot \log n)$ by Dijkstra's algorithm.

In this paper we show how to solve SSSP on planar graphs in time $O(n^{3/2} \log n)$. This should be seen in contrast to the running time of the Bellman/ford algorithm. More generally we show how to improve upon the Bellman/Ford algorithm for all graphs having a separator property.

Definition: Let $a, b \in \mathbb{R}$, $0 < a < 1 \leq b$.
A family F of graphs has the (a,b)-separator property if for every $G = (V, E) \in F$, $n = |V|$, we have

(1) $|E| = O(n^b)$

(2) there is a partition of V into three pairwise disjoint sets V_1, S, V_2 such that

(2.1) $|S| = O(n^a)$

(2.2) $|V_1 \cup S| \leq 2n/3$, $|V_2 \cup S| \leq 2n/3$

(2.3) there are no edges $(v_1, v_2) \in E$ with $v_1 \in V_1$, $v_2 \in V_2$, i.e. S separates V_1 from V_2.

(2.4) $G_i = (V_i \cup S, E_i)$, the subgraph spanned by $V_i \cup S$, belongs to F for i=1,2.

Theorem: Let F be a family of graphs having an (a,b)-separator property such that separators can be found in time n^d and space n^e for a graph G in F with n nodes.
Then the single single source shortest path problem on graphs in F can be solved in time $O(n^{a+b} \log n + n^{3a} + n^d)$
and space $O(n^{a+1} + n^e)$.

Together with the planar separator theorem of Lipton/Tarjan we get the following

Corollary: SSSP on planar networks can be solved in time $O(n^{3/2}\log n)$ and space $O(n^{3/2})$.

For reasons of fairness we want to stress that our algorithm uses more space than the Bellman/Ford algorithm. Bellman/Ford uses $O(n)$.

2. The Algorithm

Let $N = (V, E, C)$ be a network out of a family F of graphs having the (a,b)-separator property. If $W \subset V$ then the network induced by W is $(W, (W\times W)\cap E, C)$.

Let us first present the following observation due to Edmonds/Karp as

Lemma 1: Let $N = (V, E, C)$ be a network, $s \in V$.
Suppose we have computed $\mathrm{Min}(s,x;V)$ for all $x \in V$.
Define $\overline{C} : E \to \mathbb{R}$ by
$\overline{C}(u,v) = C(u,v) + \mathrm{Min}(s,u;V) - \mathrm{Min}(s,v;V)$ then

a) $\overline{C}(u,v) \geq 0$ for all $(u,v) \in E$

b) if p is a path from u to v, and $l(\overline{l})$ is the length of p with respect to C ($\overline{C}$) then $\overline{l}(p) = l(p) + \mathrm{Min}(s,u;V) - \mathrm{Min}(s,v;V)$

Remark: The Lemma 1 shows that shortest paths with respect to C are identical to shortest paths with respect to $\overline{C}$.

Proof of Lemma 1:

a) $\mathrm{Min}(s,u;V) + C(u,v) < \mathrm{Min}(s,v;V)$ is a contradiction to the definition of the function "Min".

b) Let $p = v_0 \ldots v_k$, $u = v_0$, $v = v_k$ then we have

$$\overline{l}(p) = \sum_{i=0}^{k-1} \overline{l}(v_i, v_{i+1})$$

$$= \sum_{i=0}^{k-1} l(v_i,v_{i+1}) - \mathrm{Min}(s,v_{i+1};V) + \mathrm{Min}(s,v_i;V)$$

$$= l(p) - \mathrm{Min}(s,v;V) + \mathrm{Min}(s,u;V)$$

Lemma 1 above can be used to derive an efficient many source shortest path algorithm. Let $S \subset V$. Suppose that we want to compute $Min(t,v;V)$ for all $t \in S$ and $v \in V$. This can be done as follows:

(1) Choose $s \in S$ arbitrary and compute $Min(s,v;V)$ for all $v \in V$.

(2) Transform edge weights as described in Lemma 1, solve $|S| - 1$ shortest path problems with non-negative edge costs and translate back the result.

The running time of this algorithm is the running time of one single source problem plus $O(|S| \cdot e \cdot \log n)$.

We are now in the position to describe our new algorithm.

Let $N = (V, E, C)$ be a network belonging to a family of graphs with an (a,b)-separator property

Algorithm SSSPS - Single Source Shortest Path on graphs with Separators

1. Find a separator S of N, i.e. find a partition V_1, S, V_2 of V which satisfies conditions (2.1) to (2.4). Set $S := S \cup \{s\}$.
   Remark: By assumption this step takes time $O(n^d)$ and space $O(n^e)$.

2. Compute $Min(t,v;V_1 \cup S)$ for all $t \in S$, $v \in V_1 \cup S$ and compute $Min(t,v;V_2 \cup S)$ for all $t \in S$, $v \in V_2 \cup S$.
   Remark: In this step we apply the many source algorithm described above. In order to solve the required single source problems on the subnetwork spanned by $V_1 \cup S$ ($V_2 \cup S$) we use our algorithm recursively. Thus running time of this step is
   $T(|V_1 \cup S|) + T(|V_2 \cup S|) + O(n^a n^b \cdot \log n)$
   where T is the running time (yet to be determined) of our algorithm. Similarly space requirement is
   $S(|V_1 \cup S|) + S(|V_2 \cup S|) + n^a n$

3. Let $\overline{N} = (S, S \times S, \overline{C})$ where $\overline{C}(r,t) = \min\{\infty, Min(r,t,V_1 \cup S), Min(r,t,V_2 \cup S)\}$
   Solve the single source shortest path problem on $\overline{N}$ with source $s \in S$.
   Call the solution $\overline{Min}(s,t;S)$.

Remark: Because $\overline{N}$ is a complete network we need time $O(n^{3a})$ and space $O(n^{2a})$. We show in claim 1 below that: $\overline{Min}(s,t;S) = Min(s,t;V)$ for all $t \in S$.

4. For each $v \in V$ compute

$$Min(s,v;V) = \begin{cases} \min\limits_{t \in S} \{ \overline{Min}(s,t;S) + Min(t,v;V_1 \cup S)\} & \text{if } v \in V_1 \\ \min\limits_{t \in S} \{ \overline{Min}(s,t;S) + Min(t,v;V_2 \cup S)\} & \text{if } v \in V_2 \end{cases}$$

Remark: To combine the intermediate results computed so far we need time $O(n^{1+a})$. Claim 2 below shows the correctness of the result.

Theorem 1: Let F be a family of graphs having an (a,b)-separator property. Let N = (V, E, C) be a network in F and let s be a fixed vertex of V.
Then algorithm SSSPS computes Min(s,x;V) for all $x \in V$.

Proof:
claim 1: After performing step 3 we have for $t \in S$
$\overline{Min}(s,t;S) = Min(s,t;V)$

Proof: Let t be any vertex in S. A shortest path w from s to t in N has the form

$$s = s_0 \xrightarrow{g_0^1} s_1 \xrightarrow{g_0^2} s_2 \xrightarrow{g_1^1} s_3 \xrightarrow{g_1^2} \ldots \rightarrow s_k = t$$

where g_i^j is a shortest path from vertex $s_{2i+j-1} \in S$ to vertex $s_{2i+j} \in S$ running completely within $S \cup V_j$ $(j=1,2)$.
Hence

$$Min(s,t;V) = l(w) = \sum_{i,j} l(g_i^j) = \sum_h \overline{C}(s_h, s_{h+1})$$

$$= \overline{Min}(s,t;S)$$

The third equality follows from the fact that $l(g_i^j) = \overline{C}(s_{2i+j-1}, s_{2i+j})$ by definition of $\overline{C}$.

claim 2: After step 4 we have computed Min(s,x;V) for each $x \in V$.
Proof: Let x be any vertex of N, $w = sv_1 \ldots v_m x$ the shortest path to x.
case 1: $x \in S$: claim 1 yields the stated result.

case 2: $x \in V_1$

On the path from s to x there exists at least one vertex $v_i \in S$ (possibly s itself).

Break w into w = s...t...x where t is the last vertex lying on w with $t \in S$. Claim 1 yields that the shortest path from s to t is found. The choice of t and $x \in V_1$ yields for $tv_r \ldots v_m x$ that $v_r, \ldots, v_m \in V_1$. This part of w is computed in step 2 as $\mathrm{Min}(t, x; V_1 \cup S)$. Summing the two values gives the shortest path from s to x.

case 3: $x \in V_2$

The argumentation runs totally analogous to case 2 where V_1 is substituted by V_2.

We are now ready to give the main theorem

Theorem 2: Let F be a family of graphs having an (a,b)-separator property such that separators can be found in time $O(n^d)$ and space $O(n^e)$ for a graph G in F with n nodes. $(0<a<1\leq b)$

Then the single source shortest path problem on graphs G in F can be solved in

time $O(n^{a+b} \log n + n^{3a} + n^d)$ and

space $O(n^{a+1} + n^e)$.

Proof: Summing the time bounds already stated at each step of the algorithm we get:

$$T(n) = \max_{\substack{n_1, n_2 \leq 2n/3 \\ n_1 + n_2 = n + n^a}} \left[T(n_1) + T(n_2) + n^{a+b} \log n + n^{3a} + n^d \right]$$

and $T(1) = 1$.

Recall that $|V_1 \cup S| \leq 2n/3$, $|V_2 \cup S| \leq 2n/3$ and

$|V_1 \cup S| + |V_2 \cup S| \leq n + n^a$.

T(n) is clearly non-decreasing.

Let $U(n) = T(n)/n$.

Then $U(1) = 1$ and

$$U(n) = \max_{\substack{n_1, n_2 \leq 2n/3 \\ n_1 + n_2 = n + n^a}} \left[(n_1/n)U(n_1) + (n_2/n)U(n_2) + n^{a+b-1} \log n + n^{3a-1} + n^{d-1} \right]$$

$$\leq (1+n^{a-1})\ U(2n/3) + n^{a+b-1}\log n + n^{3a-1} + n^{d-1}$$

since $U(n_1) \leq U(n_2) \leq U(2n/3)$ and $n_1+n_2 = n + n^a$. Hence

$$U(n) \leq \sum_{i=0}^{(\log n)/\log(3/2)} \left[\prod_{j=0}^{i} (1+((2/3)^j n)^{a-1}) \right] \cdot f((2/3)^i n)$$

where $f(m) = m^{a+b-1}\log m + m^{3a-1} + m^{d-1}$.

$$\leq \sum_{i=0}^{(\log n)/\log(3/2)} \prod_{j=0}^{i} \left[1 + ((2/3)^j n)^{a-1} \right] \cdot (2/3)^{i\cdot c}$$

where $c = \mathrm{Max}(\ a+b-1,\ 3a-1,\ d-1) > 0$.

Next note that

$$\prod_{j=0}^{i} \left[1 + ((2/3)^j n)^{a-1} \right] = e^{\sum_{j=0}^{i} \ln(1+((2/3)^j n)^{a-1})}$$

$$\leq e^{\sum_{j=0}^{i} ((2/3)^j n)^{a-1}}$$

$$\leq e^{O(\, n^{a-1}(3/2)^{(1-a)\cdot i}\ )}$$

$$\leq e^{O(\, n^{a-1}\cdot n^{1-a})} = O(1)$$

since $i \leq (\log n)/\log(3/2)$ and hence $(3/2)^i \leq n$.

We can now finally conclude

$$U(n) = O(f(n))$$

and hence $$T(n) = O(n^{a+b}\log n + n^{3a} + n^d)$$

Corollary: The single source shortest path problem on planar graphs is solvable in

time $O(n^{3/2}\log n)$ and

space $O(n^{3/2})$.

Proof: From Lipton/Tarjan we get d=1, a=1/2 and b=1.

References:

Bellman R.E.
On a Routing Problem
Quart. Appl. Math. 16 (1958) pp. 87-90

Dijkstra E.W.
A Note on two Problems in Connection with Graphs
Num. Math. Vol. 1 (1959) pp. 269-271

Edmonds J., Karp R.M.
Theoretical Improvements in Algorithmic Efficiency for Network Flow Problems
J. ACM 5 (1962) p. 345

Lipton R., Tarjan R.E.
A Separator Theorem for Planar Graphs
Waterloo Conference on Theoretical Comp. Science Aug.77 pp. 1-10

Floyd F.W.
Algorithm 97: Shortest Path
C. ACM 5 (1962) p. 345