

6.889 — Lecture 25: Single-Source Shortest Paths with Negative Lengths in Minor-Free Graphs

Christian Sommer csom@mit.edu

December 12, 2011

Setting directed graph $G = (V, A)$, underlying undirected graph is H -minor-free, arbitrary real arc lengths $\ell : A \rightarrow \mathbb{R}$ or integer arc lengths $\ell : A \rightarrow \mathbb{Z}$ such that $\forall a : \ell(a) \geq -L$ for some $L \geq 0$

Single-Source Shortest Path cannot use Dijkstra's algorithm since arc lengths may be negative

General		H -Minor-Free	Planar
$\mathcal{O}(mn)$	Bellman-Ford	$\mathcal{O}(n^{3/2})$	$\mathcal{O}(n \log^2 n)$ Lecture 14
$\mathcal{O}(m\sqrt{n} \log L)$	Goldberg	$\mathcal{O}(n^{4/3} \log n \log L)$ today	$\mathcal{O}(n^{5/4+\epsilon} \log L)$

High-Level Algorithm (framework) based on separators and r -divisions

1. compute r -division
2. FOR EACH piece P , compute feasible price function φ_P (Bellman-Ford, or Goldberg, or recurse)
 - \rightsquigarrow non-negative edge lengths in P (reduced lengths $\ell_{\varphi_P}(u, v) = \ell(u, v) + \varphi_P(u) - \varphi_P(v)$, details in L. 14)
3. FOR EACH piece P , compute dense distance graph $DDG_P(\partial P)$
 - all-pairs distances among boundary nodes ∂P
 - distances with respect to subgraph induced by P , reduced lengths defined by ℓ and φ_P
4. run Goldberg's algorithm on graph with vertex set $\bigcup_P \partial P$ and arc set $\bigcup_P DDG_P(\partial P)$
 - \rightsquigarrow know distances from s to all boundary nodes
5. FOR EACH piece P , compute distances from ∂P to all $v \in P$

Correctness proof similar to proof in Lecture 14

Running time analysis let $r := (n \log L)^{2/3}$ (for recursive version set $r := n^{1-\epsilon}$)

1. r -division in $\mathcal{O}(n \log n)$
assuming "fast" $\mathcal{O}(\sqrt{n})$ -separator!
2. for $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$, run Goldberg's algorithm in $\mathcal{O}(r\sqrt{r} \log L)$, overall $\mathcal{O}(n\sqrt{r} \log L)$
 (could recurse instead of running Goldberg's algorithm)
3. for $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$, run $\mathcal{O}(\sqrt{r})$ non-negative SSSP computations, overall $\mathcal{O}(n\sqrt{r})$
 (can potentially be done faster using a distance oracle, see for example Lecture 12)
4. run Goldberg's algorithm on graph with $\mathcal{O}(n/\sqrt{r})$ vertices and $\mathcal{O}(n)$ arcs, time $\mathcal{O}(n^{3/2}r^{-1/4} \log L)$
5. for $\mathcal{O}(n/r)$ pieces, do $\mathcal{O}(\log r)$ iterations of Bellman-Ford in time $\mathcal{O}(r \log r)$, overall $\mathcal{O}(n \log r)$
 non-trivial, uses shortcuts such that shortest paths have at most $\mathcal{O}(\log r)$ arcs, proof in papers

Goldberg's Algorithm

see also: references,

slides of Andrew V. Goldberg: <http://www.avglab.com/andrew/pub/path05-slides.pdf>, and
lecture notes of Uri Zwick: <http://www.cs.tau.ac.il/~zwick/grad-algo-09/short-path.pdf>

Scaling

given a directed graph $G = (V, A)$ with length function $\ell : A \rightarrow \mathbb{Z}$,

- IF $\exists a : \ell(a) < -1$ THEN

1. recurse with $\ell'(a) := \lceil \ell(a)/2 \rceil$, recursive call computes feasible price function φ'

2. adjust price function to ℓ :

reduced lengths are non-negative for ℓ' , initialize $\varphi(v) := 2\varphi'(v)$

$$\begin{aligned} \ell'_{\varphi'}(u, v) &= \ell'(u, v) + \varphi'(u) - \varphi'(v) \geq 0 \\ 2\ell'_{\varphi'}(u, v) &= 2\ell'(u, v) + 2\varphi'(u) - 2\varphi'(v) \geq 0 \\ &= (2\ell'(u, v) - 1) + 2\varphi'(u) - 2\varphi'(v) \geq -1 \\ \ell_{\varphi}(u, v) &= \ell(u, v) + \varphi(u) - \varphi(v) \geq -1 \end{aligned}$$

find feasible price function or negative-length cycle in graph with $\forall a : \ell_{\varphi}(a) \geq -1$

- ELSE find feasible price function or negative-length cycle in graph with $\forall a : \ell(a) \geq -1$

SSSP algorithm for integral arc lengths ≥ -1

Thm. Given a (general) directed graph with integer arc lengths ≥ -1 , there is an algorithm computing either a negative-length cycle or a feasible price function in time $\mathcal{O}(m\sqrt{n})$.

Algorithm main loop: work on *admissible graph*, fix vertices v with negative-length edges entering v

- compute admissible graph (non-positive subgraph) $G^- = (V, A^-)$ where $A^- = \{a \in A : \ell(a) \leq 0\}$

- compute strongly connected components of G^-

IF there exists a component that contains arc a with $\ell(a) = -1$, then there is a negative-length cycle

ELSE contract each connected component into one node (only zero-length arcs) \rightsquigarrow acyclic graph

- define vertex v to be *negative* if there is a negative-length arc (u, v) entering v

repeatedly “fix” negative vertices (fixing vertex v means fixing all arcs (u, v) entering v)

decreasing potential of v by 1 fixes all -1 arcs entering v , which now have reduced length ≥ 0

Problem arcs (v, w) leaving v could get worse (fortunately working on a DAG)

decrease potential of all nodes in $R(v) := \{w : \text{reachable from } v\}$, fixes v without new problems

computing $R(v)$ and decreasing potential may require $\Omega(m)$ time \rightsquigarrow fix many vertices simultaneously

- connect super-source s by zero-length arc to each v in G^- , compute distance $d_{G^-}(s, v)$ (à la BFS)
let level $L_i := \{v : d_{G^-}(s, v) = -i\}$, let L_r be last level. suppose there are k negative vertices.

- IF $r \leq \sqrt{k}$ THEN there is a layer L_i with $\geq \sqrt{k}$ negative vertices

fix all of them simultaneously: decrease potential by 1 for all $v \in \bigcup_{j \geq i} L_j$

- ELSE ($r > \sqrt{k}$) fix all $> \sqrt{k}$ vertices on a shortest path from s to any node $t \in L_r$ simultaneously
let (w_1, w_2, \dots, w_r) be negative vertices on path; either fix all w_i or find negative-length cycle:
in G , let $\ell^+(a) = \max\{\ell(a), 0\}$, connect super-source s' by arc to each w_i , length $\ell^+(s', w_i) := r - i$
and $\ell^+(s', v) = r$ for all other $v \in V$; compute distances from s' , define price func. $\varphi(v) = d(s', v)$

Running time analysis time $\mathcal{O}(m)$ for each iteration of main loop, $\mathcal{O}(\sqrt{n})$ iterations (in each iteration, total number of negative vertices decreases from k to at most $k - \sqrt{k}$)

Separators for H -Minor-Free Graphs

Problem for H -minor-free graphs, we do not know how to compute $\mathcal{O}(\sqrt{n})$ -separators in linear time (recall separator algorithm discussed in L. 6, which runs in $\mathcal{O}(n^{3/2})$) \rightsquigarrow cannot compute “standard” r -division \rightsquigarrow work with (r, s) -division, a partition into $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$ with $\mathcal{O}(s)$ boundary nodes (planar: $s = \sqrt{r}$, here: $s = r^\gamma$ for some $\gamma \in [1/2, 1)$)

Running time analysis using (r, s) -division, let $s := r^\gamma$

1. (r, s) -division in time $\mathcal{O}(T(r, \gamma))$
2. for $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$, run Goldberg’s algorithm in $\mathcal{O}(r\sqrt{r} \log L)$, overall $\mathcal{O}(n\sqrt{r} \log L)$
3. for $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$, run $\mathcal{O}(r^\gamma)$ SSSP computations, overall $\mathcal{O}(nr^\gamma)$
4. run Goldberg’s algorithm on graph with $\mathcal{O}(nr^{\gamma-1})$ vertices and $\mathcal{O}(nr^{2\gamma-1})$ arcs, time $\mathcal{O}(n^{3/2}r^{5\gamma/2-3/2} \log L)$
5. for $\mathcal{O}(n/r)$ pieces, do $\mathcal{O}(\log r)$ iterations of Bellman-Ford in time $\mathcal{O}(r \log r)$, overall $\mathcal{O}(n \log r)$

Overall time (up to constant factors)

$$T(r, \gamma) + nr^\gamma + n^{3/2}r^{(\gamma-1)/2} \log L$$

Balanced running times of step (3) and step (4) if

$$\begin{aligned} nr^\gamma &= n^{3/2}r^{5\gamma/2-3/2} \log L \\ r^{3-3\gamma} &= n \log L \end{aligned}$$

Check: $\gamma = 1/2$ yields $r = (n \log L)^{2/3}$

Lemma (discussed in L. 5 & L. 6). *There is an algorithm that finds a separator of size $\mathcal{O}(n^{2/3})$ in linear time.*

Overall running time for $\gamma = 2/3$, no improvement *but* can do better by using a *time vs. γ* tradeoff

Lemma (Wulff-Nilsen). *For H -minor-free graphs, there is an algorithm that finds a separator of size $\mathcal{O}(\sqrt{n \log n})$ in time $\mathcal{O}(n^{5/4+\epsilon})$ for any constant $\epsilon > 0$.*

Running time analysis let $r := n^{2/3} \log n \log^{4/3} L$

1. r -division in time $\mathcal{O}(n^{5/4+\epsilon})$, boundary size $\mathcal{O}(\sqrt{r \log n})$
2. for $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$, run Goldberg’s algorithm in $\mathcal{O}(r\sqrt{r} \log L)$, overall $\mathcal{O}(n\sqrt{r} \log L)$
3. for $\mathcal{O}(n/r)$ pieces of size $\mathcal{O}(r)$, run $\mathcal{O}(\sqrt{r \log n})$ SSSP computations, overall $\mathcal{O}(n\sqrt{r \log n})$
4. run Goldberg’s algorithm on graph with $\mathcal{O}(n/\sqrt{r \log n})$ vertices and $\mathcal{O}(n \log n)$ arcs, time $\mathcal{O}(n^{3/2}r^{-1/4} \log^{5/4} n \log L)$
5. for $\mathcal{O}(n/r)$ pieces, do $\mathcal{O}(\log r)$ iterations of Bellman-Ford in time $\mathcal{O}(r \log r)$, overall $\mathcal{O}(n \log r)$

Overall time

$$\mathcal{O}(n\sqrt{r \log n}) + \mathcal{O}(n^{3/2}r^{-1/4} \log^{5/4} n \log L) = \mathcal{O}(n^{4/3} \log n \log^{2/3} L)$$

References

For general graphs, if arbitrary negative edge lengths are allowed, the fastest strongly-polynomial-time algorithm for shortest paths is due to Bellman and Ford [Bel58, For56, Bel67], and it runs in time $\mathcal{O}(mn)$.

For integral edge lengths, Goldberg [Gol95] found an efficient algorithm for the case where each edge has length at least -1 . Combined with a scaling framework, his algorithm computes shortest paths in time $\mathcal{O}(m\sqrt{n} \log L)$, where L denotes the absolute value of the smallest negative edge length.

For H -minor-free graphs, Alon and Yuster [AY10] extended the planar nested dissection result [LRT79] (result on shortest paths with negative lengths not stated explicitly). Yuster [Yus10] extended the separator-based framework of Henzinger, Klein, Rao, and Subramanian [HKRS97] to minor-free graphs, obtaining an algorithm running in time $\mathcal{O}(n^{\sqrt{23/2}-2} \log L)$. More recently, Wulff-Nilsen [WN11] obtained further improvements, giving an algorithm that runs in time $\mathcal{O}(n^{4/3} \log n \log L)$ (which, up to \log factors, matches the earlier bound for planar graphs in [HKRS97]).

- [AY10] Noga Alon and Raphael Yuster. Solving linear systems through nested dissection. In *51st IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 225–234, 2010.
- [Bel58] Richard Ernest Bellman. On a routing problem. *Quarterly of Applied Mathematics*, 16:87–90, 1958.
- [Bel67] Richard Ernest Bellman. *Dynamic Programming*. Princeton University Press, 1967.
- [For56] Lester R. Ford. Network flow theory. Report P-923, The Rand Corporation, 1956.
- [Gol95] Andrew V. Goldberg. Scaling algorithms for the shortest paths problem. *SIAM Journal on Computing*, 24(3):494–504, 1995. Announced at SODA 1993.
- [HKRS97] Monika Rauch Henzinger, Philip Nathan Klein, Satish Rao, and Sairam Subramanian. Faster shortest-path algorithms for planar graphs. *Journal of Computer and System Sciences*, 55(1):3–23, 1997. Announced at STOC 1994.
- [LRT79] Richard J. Lipton, Donald J. Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM Journal on Numerical Analysis*, 16:346–358, 1979.
- [WN11] Christian Wulff-Nilsen. Separator theorems for minor-free and shallow minor-free graphs with applications. In *52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, 2011. To appear, preprint available on the arXiv: <http://arxiv.org/abs/1107.1292>.
- [Yus10] Raphael Yuster. Single source shortest paths in H -minor free graphs. *Theoretical Computer Science*, 411(34–36):3042–3047, 2010.