Shortest Paths in Near-Linear Time

(FOCS 22 paper of Bernstein, Nanongkai and Wulff-Nielsen).

- Integer Weights
- Single Source Shortest Paths
- Negative weights (but no neg. cycles, etc.)

. Non negative weights : Dijkstra $O(m + n\log n)$
+ Fib Heaps.

. Negative weights : Bellman-Ford-Moore $O(mn)$.

. Better for integer weights of small magnitude (see our old notes for Andrew Goldberg's algo in $O(m\sqrt{n}\lg C)$)
↑ max neg weight in absolute value.

. This paper : $O(m \operatorname{polylog}(n))$!!!

randomized also using ideas we've seen ~~(but~~

(but using them in interesting ways).
↑ extending, even ↑ but simple

. Ingredients

- Dijkstra's Algorithm (~~but~~ but the k-fold version)
- Bellman-Ford-Moore (but run for smaller # of iterations).
- Low-diameter decompositions (but on directed graphs)
- Prices / Feasible potentials
- SSSP on Dags in Linear time

Other refs : see Youtube talks by Danupon N. ← my lecture follows this talk fairly closely
Aaron B.

Recall: prices / potentials φ

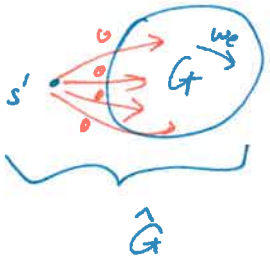
give new weights $w_\varphi(u,v) = \varphi(u) + w(u,v) - \varphi(v)$.

(i) w_φ has same shortest paths as w, just different lengths.

(ii) if $w_\varphi \geq 0$ then can use Dijkstra.

"feasible" potentials

(iii) one possible φ obtained by computing SPs in graph $\hat{G}$ and setting $\varphi(v) = \hat{d}(s',v)$



b/c. $\varphi(v) = \hat{d}(s',v) \leq \hat{d}(s',u) + w_{uv}$

$= \varphi(u) + w_{uv}$

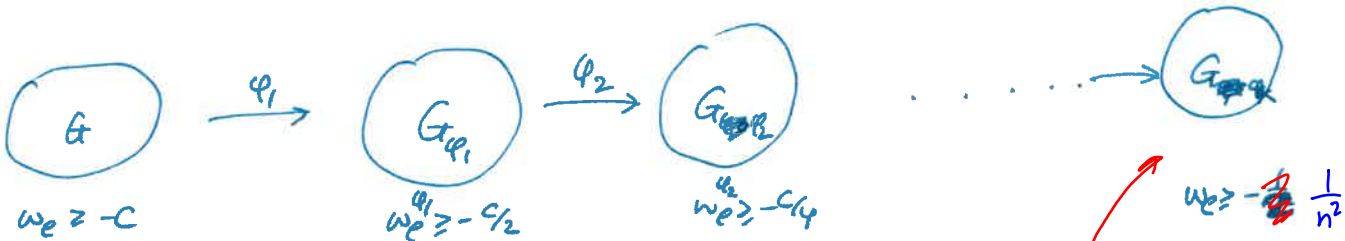
$\Rightarrow \varphi$ is feasible potential.

feasible prices $\Downarrow \Uparrow$ SSSP

But may take too much time, since need SSSP on $\hat{G}$ (chicken & egg).

So use a more gradual approach.

"scaling"-based approach



~~Now since weights are integers, $w_e \geq -1/2$ means $w_e \geq 0$ and we're all set in $O(\log C)$ rounds~~

See later for what to do here

Our ~~current~~ ultimate goal: $w_e \geq -C$ $\xrightarrow[\text{find } \varphi]{}$ $w_e^\varphi \geq -C/2$

current goal: $w_e \geq -2$ $\xrightarrow{\text{find } \varphi}$ $w_e^\varphi \geq -1$.

(will extend to the ultimate goal easily)

(I) Bellman-Ford-Moore(K)

$d(s)=0$ $d(u)=\infty$ $\forall u \neq s$

for $i = 1$ to K

for $e \in$ edges of G, $e=(u,v)$

$d(v) \leftarrow \min\{d(v), d(u)+w_{uv}\}$

Thm 1: if all shortest paths have at most k edges then BFM(K) correctly returns the shortest path labels (i.e. $d(u) = dist(s,u)$ $\forall u$).

But, of course, shortest paths may be longer than polylog(n)!

(II) Dijkstra's Algorithm (K)

$d(s)=0$, $d(u)=\infty$ $\forall u \neq s$

for $i = 1$ to $K+1$

all vertices are unmarked.

pick v with smallest label $d(v)$, v is unmarked

mark v, "relax" all outgoing edges $d(u) \leftarrow \min\{d(u), d(v)+w_{vu}\}$ $\forall u$ s.t. $(v \to u) \in E$

Thm 2: if all shortest paths have at most K negative weight edges then Dijkstra(K) correctly returns SSSP labels, runs in time $\tilde{O}(mK)$ (HW0)

Thm 3: Let $\eta(v)$ be # of neg-weight edges on shortest path from s to v.

A variant of Dijkstra(K) correctly computes SSSP, runs in time

$$\tilde{O}\left(\sum_{v \in V} \eta(v)\right)$$

↑ subsumes Thm 2. (pf: maybe later)

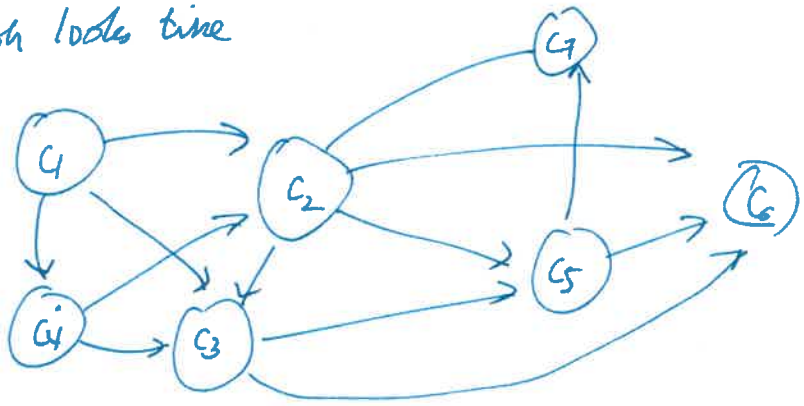
FixDag(G):

Recall the Strongly connected component decomposition of a Digraph G.

SCCs (C is an SCC if $\forall u,v \in C$, $\exists$ a dipath from u to v & from v to u)

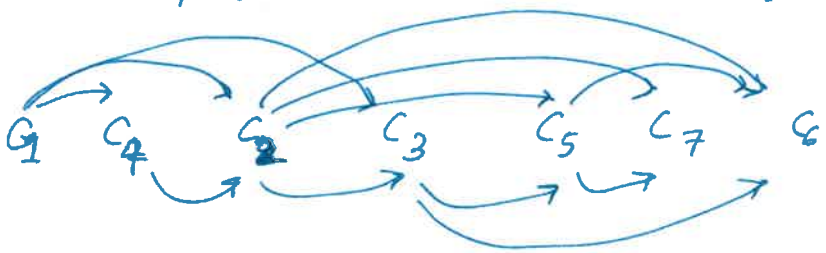
this gives an equivalence relation on V

So the graph looks like



The edges outside the SCCs form a DAG (directed acyclic graph)

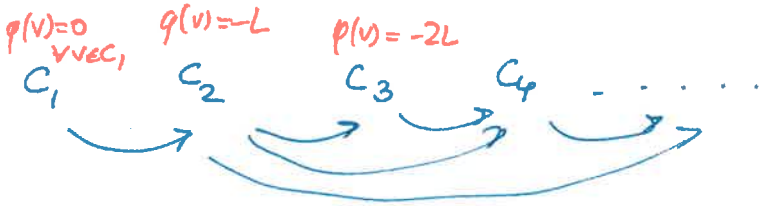
Can put these in topological order so that edges only go from Left → Right



can find this in linear time.

Now can make all reduced costs of these DAG-edges ≥ 0.

How: set $p(v) = 0$ $\forall v \in C_1$, $p(v) = -L$, $p(v) = -2L$



then $w_p(u,v) = (-iL) + w_{uv} - (-jL) = (j-i)L + w_{uv}$

for $u \in C_i$

and $v \in C_j$, $j > i$

make L large enough to make this ≥ 0.

Thm 4: Can "fix" all DAG edges of SCC in time $O(m)$.

IV Low diameter Decompositions

Recall the definition from the last lecture. But of course we now have digraphs, and negative weights. Can't handle negative wts, but can get analog for digraphs with $w_e \geq 0$.

Thm: Given H with $\ell_e \geq 0$ (directed), and parameter $D \geq 0$.

Then $\exists$ randomized algo that deletes edge set E_{del} s.t.

- $\forall$ strongly connected components C in $H \setminus E_{del}$, diameter$_\ell(C) \leq D$

  ↑ $\max_{u,v \in C} dist_\ell(u,v) \leq D$.

- $\forall e$, $\Pr[e \in E_{del}] \leq \frac{\ell_e}{D} \cdot O(\log^2 n)$

  ↑ was $O(\log n)$ for undir graphs

Pf: (omit at first reading)

OK, we have most components. Let's see the plan.

Given G, ~~exact~~ with $w(e) \geq -2$, want φ s.t. $w_\varphi(e) \geq -1$.

Plan [1st level]: define G^{+1} as graph with weights $w(e)+1$ ($w^{+1}(e) :=$), G^{+2} has $w(e)+2$ ($w^{+2}(e) :=$) $\forall e$.

- Find φ s.t. G^{+1}_φ has non negative lengths $\Rightarrow$ G_φ has $w_\varphi(e) \geq -1$.
  $w^{+1}_\varphi(e) \geq 0$.

How?

1. G^{+2} has $w^{+2}(e) \geq 0$ $\Rightarrow$ perform LDD with some param D.
   to delete E_{del}.

2. Now on G^{+1}/E_{del}
   (i) Find φ_1 to fix the SCC edges. (use that SCC have low diam).
   (ii) Find φ_2 to fix the DAG edges (use DAG fixing).

3. now: on G^{+1}
   - Find φ_3 to fix E^{+1}_{del} (use that SPs have few negative edges due to LDD; then use Dijkstra^{++}.

$\Rightarrow$ get φ s.t G^{+1}_φ has $w^{+1}_\varphi(e) \geq 0$ $\Rightarrow$ G_φ has $w_\varphi(e) \geq -1$.

———— × ————

This gives $O(m\sqrt{m})$ when $D = \sqrt{m}$.

Then (a) can recurse in step 2(i) instead.
(b) and choose better choice of D $\Rightarrow$ get $O(mpoly\lg m)$.

Then scale our distances to get $w(e) \geq -\Delta$ $\Rightarrow$ $w_\varphi(e) \geq -\Delta/2$. ~~Finish~~.

Beta?

Details: (1.) is easy from definition.

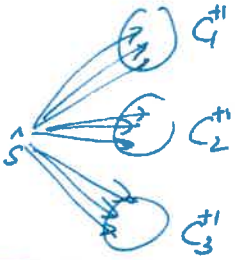
Step 2. for graph G^{+1}, define $\hat{s}+G^{+1}$

want to find SPs from $\hat{s}$ to get prices.

(i) drop $E_{del} \cup E_{dag}$.

Find SPs for each SCC separately.

Want to use BF (D rounds).



Alert! that uses that all SPs have few edges.

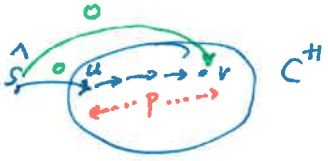
But we only have bounds on the weighted diameter!!

Main Lemma: sps C^{+1} has no negative cycles, and diameter$(C^{+1}) \le D$.

then $\hat{s}+C^{+1}$ has shortest paths with $\le D+1$ edges.

Pf:

sps. shortest path has $\ge D+2$ edges. from $\hat{s}$ to v



$\Rightarrow$ P (tail of path) has $\ge D+1$ edges.

note: $w^{+1}(P) \le 0$ b/c of construction of $\hat{s}+C^{+1}$

so $w(P) \le -(D+1)$ b/c $w^{+1} = w+1$.

but $d(v,u) \le D$ b/c diameter

$\Rightarrow$ neg. cycle, contradiction ☺.

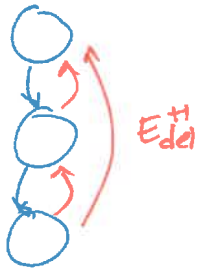
$\Rightarrow \hat{s}+C^{+1}$ has few edges in each SP. $(\le D+1)$

$\Rightarrow$ can just use BF $(D+1$ rounds.)

time = $O(mD)$

#2(ii) DAG edges put back in, and fixed in $O(m)$ time.

#3: How many ~~edges~~ negative edges in E^{+1}_{del} need to be fixed? (in expectation?)



Note: $\begin{cases} e \in E^{+1}_{del} \text{ and } e \text{ negative} \Rightarrow w^{+2}(e) \in [0,1]. \\ \text{b/c } w(e) \geq -2 \text{ and } w^{+1}(e) < 0 \end{cases}$

$\Rightarrow \Pr[e \in E_{del}] = \frac{w(e)}{D} \cdot O(\log^2 n) \leq \frac{O(\log^2 n)}{D}.$

$\Rightarrow E[\text{~~Edel~~} \mid E^{+1}_{del} \cap \text{negative in } w^{+1}] \leq \tilde{O}(\frac{m}{D}).$

$\Rightarrow E[\text{runtime of Dijkstra to fix all these}] \leq \tilde{O}(m) \times \tilde{O}(m/D).$
↑ repeated Dijkstra!

$\Rightarrow$ total time: $O(mD + m\alpha + m^2/D) \Rightarrow$ set $D = \sqrt{m}$ to get $O(m^{3/2})$.

Let's do better: ~~SSSP~~

SSSP(G, Δ) // invariant: (G has each SCC having diam $\leq \Delta$ and $w(e) \geq -2$)

(1) LDD with $D = \Delta/2^{-1}$, delete E_{del}

(2) (i) recurse on $(G \setminus E_{del}, \Delta/2)$ // invariant maintained,

get back φ s.t. ~~$w(e) \geq -1$~~ $w(e) \geq -1 \ \forall e \in G \setminus E_{del} \Rightarrow w^{\varphi}(e) \geq 0$

(3) Fix E^{+1}_{del}.

BTW: DAG fix in initial base case now, if $E_{del} = \emptyset$.

But now: Each vertex v has $\eta(v) \leq \sum^{+}_{e \in \text{esp.}} \frac{1}{(\Delta/2)} \cdot O(\log^2 n) \leq O(\log^2 n)$.

$\Rightarrow E \sum^{+} \eta(v) \leq \tilde{O}(m)$

Overall $\tilde{O}(m \log \Delta)$

Final Piece: Find φ s.t. $w(e) \geq -D \Rightarrow w_\varphi(e) \geq -D/2$