Randomized Single Source Shortest Path with Negative Real Weights in $\widetilde{O}(mn^{\frac{8}{9}})$

Jeremy T. Fineman
Georgetown University

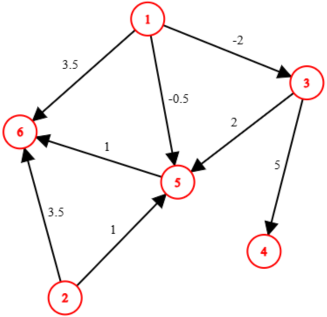
Presented by

Arman Keshazar

December 30, 2024

Problem Definition

- What's SSSP?
- What's $\widetilde{O}$?



Previous Related Works

Previous works:

- Dijkstra — Positive Weights — O(m+nlogn)
- Bellman-Ford — Negative Real Weights — O(mn)
- Johnson — Negative Real Weights — Bellman-Forde + Dijkstra — uses price function $dist(V, u)$
- Goldberg? — Negative Integer Weights — nearly linear but weakly-polynomial. Uses logW for scalling where W is the most negative edge.

This paper presents:

- Las Vegas $\widetilde{O}(mn^{\frac{8}{9}})$ with high probablity
- Best algorithm in negative real case
- Best strongly-polynomial even in negative integer case

Assumptions

- If u is a negative vertex, then $|out(u)| = 1$.
- So there are at most n negative edges.
- $deg(u) = O(\frac{m}{n})$

* All these can be acheived using some graph transformations that increase the graph size by at most a constant factor.

Preliminary Definitions - Price Function

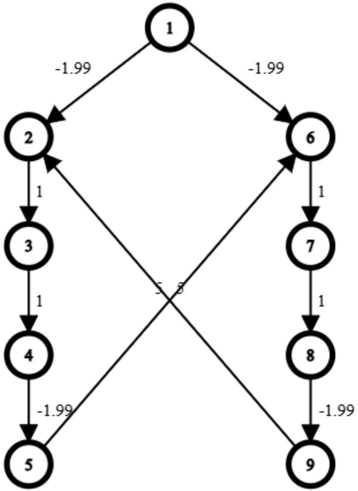
Price Function: ϕ

- $\phi : V \to \mathbb{R}$
- $w_\phi^G(u, v) := w^G(u, v) + \phi(u) - \phi(v)$
- $w_\phi^G(p) := w^G(p) + \phi(u) - \phi(v)$ for any path p
- So applying price functions on a graph does NOT change cycles' weights.
- ϕ is **valid** if non-negative edges remain non-negative.

Preliminary Definitions - Hop limited shortest path

Hop limited shortest path:

- p is a h-hop path if it only uses at most h negative edge.
- $dist_G^h(u, v)$ is the minimum of length of all h-hop paths between u and v.

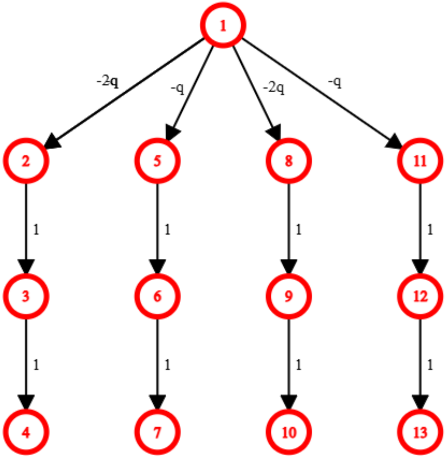


$dist^1(1, 8) = 0.01$
$dist^0(8, u) = +\infty$, u $\neq 8$

f(k)-elimination algorithm

A is an f(k)-elimination algorithm if A(G) outputs a valid reweighting ϕ, that eliminates f(k) of the negative edges, where k is the number of negative edges of G.

- Theorem. $O(k^{\frac{2}{3}})$ repetitions of a $\theta(k^{\frac{1}{3}})$-elimination algorithm reduces k by a constant factor.



$\phi(1) = q$
$\phi(u) = 0$, u $\neq 1$
is a f(k)-elimination routine for our family of *star*-graphs, where f(k) $=$ k/2.

New OBJECTIVE

Find a $\theta(k^{\frac{1}{3}})$-elimination algorithm that runs in $\widetilde{O}(mk^{\frac{2}{9}})$!

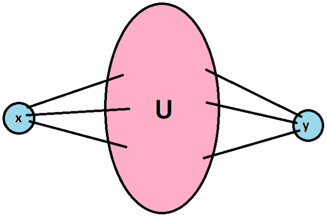
Sketch of Alogrithm

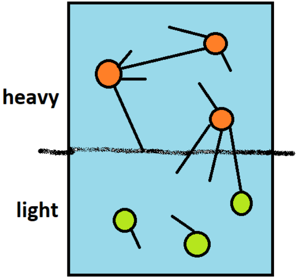
- Find a $\Omega(k^{\frac{1}{3}})$-size set of negative vertices which are *"far"* from rest of the graph.(*Remote Set)*
- Or find a $\Omega(k^{\frac{1}{3}})$-size set of negative vertices that are *"not close"* to eachother.(*1-hop Independent Set*)
- Consider the subgraph induced by these constructs and all postive edges.
- Apply a price function which eliminates all negative edges in the new graph.(Johnson's strategy but with better time)

Randomized Flavor - Betweenness Reduction

- $BW(u, v) = |\{x \in V | thru_{u,v}(x) < 0\}|$
- Let $thru_{u,v}(x_1) \leq thru_{u,v}(x_2) \leq ... \leq thru_{u,v}(x_q) < 0 < thru_{u,v}(x_{q+1}) \leq ... \leq thru_{u,v}(x_n)$
- By applying any price function ϕ we'll have:
  $thru_{u,v}(x_1) + \phi(u) - \phi(v) \leq ... \leq thru_{u,v}(x_n) + \phi(u) - \phi(v)$
- We choose a random sample of vertices with appropriate size like X, and compute a reweighting that makes $thru_{u,v}(x) \geq 0$ for all $u, v \in V$ and $x \in X$.
- Thus, with high probability, for every pair u, v we have that $BW(u, v)$ is small enough.

Randomized Flavor - Negative Sandwich





Negative Vertices

Negative Sandwich:

- U is subset of negative vertices
- $dist^1(x, u) < 0$ for all $u \in U$
- $dist^1(u, y) < 0$ for all $u \in U$

Can we find a large enough negative-sandwich?

Heavy-Light Decomposition:

- If $h \in heavy$, then we can use h and out(h) to find a negative-sandwich.
- If all vertices are light, we can probably find an 1-Hop-Independent-Set.

Fein!

src: Randomized Single Source Shortest Path with Negative Real Weights in $\widetilde{O}(mn^{\frac{8}{9}})$

Thanks for your attention :D