negative-weight SSSP notes

May 23, 2023

Setting. The goal is to find potentials $\phi(v)$ such that each edge (u, v) has nonnegative modified weight: $w_\phi(u, v) := w(u, v) + \phi(u) - \phi(v) \geq 0$. This is done iteratively through scaling. Suppose the current weights are at least $-2B$. The goal is to produce weights of at least $-B$. This is done by first adding $+B$ to each edge and then finding a potential whose modified edge weights are all nonnegative. Then, undoing the added $+B$ per edge ensures each edge has weight $\geq -B$. To find the potentials on the graph with $+B$ edge weights, add a dummy source s with an edge (s, v) of weight 0 to all vertices v, and compute SSSP distances from s which become the potentials. Let G^{+B} be the graph G with $+B$ edge weights and including the dummy source s.

Parameter κ. One new idea is to consider the parameter κ, defined as the maximum number of edges along any $s \to *$ path of weight ≤ 0 in G^{+B}. **I'm trying to relax κ as much as possible to understand the algorithm better.** Note that κ can only decrease upon taking subgraphs.

Algorithm structure. The algorithm computes potentials on graph G^{+B} recursively. At a high level:

1. Partition the vertices (excluding s) into:
   (a) Clusters of size $\leq$ half,
   (b) A remaining cluster whose induced graph has parameter κ halved, and
   (c) Intercluster edges are DAG + few extra edges, so that any path $s \to *$ of weight ≤ 0 has $O(\log n)$ non-DAG edges in expectation.
2. For each cluster U, recursively solve $G^{+B}[U \cup \{s\}]$ to obtain potentials whose modified edge weights are all nonnegative.
3. Modify the potentials according to the DAG, so that modified DAG edges are all nonnegative.
4. To correct non-DAG edges, run Dijkstra + BF hybrid on modified graph and set new potentials to be the distances. The running time depends on the sum (over all v) of the number of non-DAG edges on the shortest $s \to v$ path, which is small enough.

Decomposition. The decomposition works on graph $G^{+B}_{\geq 0}$, the graph G^{+B} with negative weight edges increased to zero. It iteratively looks for in/out-balls of radius $\kappa B/4$ with $\leq$ half vertices as long as they exist. For such an in/out-ball, it samples a random radius $\leq \kappa B/4$ and carves out a ball of that radius. (More precisely, remove the in/out-boundary of this ball.) Suppose that no more such balls remain, and let R be the remaining vertices.

Claim 1. *Any two vertices in $G^{+B}_{\geq 0}[R]$ are within $\kappa B/2$ apart in either direction. The same is true for $G^{+B}[R]$ since its edge weights can only be smaller.*

Proof. For any two vertices $u, v \in R$, the out-ball of u and the in-ball of v (of radii $\kappa B/4$) must intersect, since they each contain more than half the vertices. So there is a path $u \to v$ of weight $\leq \kappa B/2$. □

Claim 2. *If G has no negative cycle, then $G^{+B}[R]$ has parameter $\kappa/2$. (We don't even need to assume that G^{+B} has parameter κ.)*

Proof. Suppose for contradiction that there exists an $s \to v$ path in $G^{+B}[R \cup s]$ of weight ≤ 0 and with $> \kappa/2$ edges. Let u be the first vertex on the path. Since edge (s, u) has weight 0, we have a $u \to v$ path in $G^{+B}[R]$ of weight ≤ 0 and with $> \kappa/2 - 1$ edges. Extend it by a $v \to u$ path of length $\leq \kappa B/2$ to form a cycle in $G^{+B}[R]$ of weight $\leq \kappa B/2$ and with $> \kappa/2$ edges. Since all edges have weight $+B$ in G^{+B}, this cycle has negative weight in G, a contradiction. □

Claim 3. *In G^{+B}, any $s \to *$ path of weight ≤ 0 has $O(\log n)$ non-DAG edges in expectation.*

Proof. Since G^{+B} has parameter κ, any $s \to v$ path of weight ≤ 0 has at most κ edges. Since edge weights are at least $-2B$ in G, they are at least $-B$ in G^{+B}, so they are increased by $\leq B$ from G^{+B} to $G^{+B}_{\geq 0}$. So the path has weight $\leq \kappa B$ in $G^{+B}_{\geq 0}$. Since the decomposition algorithm removes balls of radii $\sim \kappa B$, standard LDD analysis shows that in expectation, $O(\log n)$ edges of the path are on the boundary of these balls. □