On single source fault tolerant approximate shortest path

Keerti Choudhary
(Weizmann Institute)

Surender Baswana
(I.I.T. Kanpur)

Moazzam Hussain
(WorldQuant Research)

Liam Roditty
(Bar Ilan University)
REAL WORLD NETWORKS

Faults

- Inevitable
- Few in numbers
- Transient in nature

failures of nodes or links
1-FAULT TOLERANT MODEL

Time $t=0$
1-FAULT TOLERANT MODEL

Time $t=1$
1-FAULT TOLERANT MODEL

Time $t=2$
1-FAULT TOLERANT MODEL

Time $t=3$
FOCUS..

Preserving / computing \((1+\varepsilon)\)-approx-distances from fixed source after single failure
**$(1+\epsilon)$-Preserver:**

Given $G$, source $s$, a subgraph $H$ is said to be a $(1+\epsilon)$-preserver if for each $x, v \in V$, 

$$\text{dist}_{H\mid x}(s, v) \leq (1+\epsilon) \text{dist}_{G\mid x}(s, v)$$

**Goal:** $H$ is sparse

---

**$(1+\epsilon)$-Oracle:**

Given $G$, source $s$, a $(1+\epsilon)$-oracle is a data-structure that for each $x, v \in V$, 

Report: $\text{dist}_{G\mid x}(s, v)$ within $(1+\epsilon)$ factor.\(^1\)

**Goals:**
- Oracle is small in size
- Reporting time is small

**Path Reporting:**
Oracle also reports $(1+\epsilon)$-stretched-path

---

\(^1\) Distances are never under-estimated.
WHAT IS KNOWN?

- Exact Preserver, Oracles need $Ω(n^{3/2})$ space for unweighted and $Ω(m)$ for weighted graphs

(1+ε) Preserver and Oracle for Undirected graphs

<table>
<thead>
<tr>
<th>Problem</th>
<th>Failure</th>
<th>Space</th>
<th>Time</th>
<th>Path Reporting</th>
<th>Weights</th>
<th>Ref</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1+ε) Preserver</td>
<td>edge/vertex</td>
<td>$O(n \log n/ε^2)$</td>
<td>——</td>
<td>——</td>
<td>Weighted</td>
<td>[BGLP14]</td>
</tr>
<tr>
<td>(1+ε) Oracle</td>
<td>vertex</td>
<td>$O(n \log n + n/ε^3)$</td>
<td>$O(1)$</td>
<td>Yes</td>
<td>Unweighted</td>
<td>[BK13]</td>
</tr>
<tr>
<td>(1+ε) Oracle</td>
<td>edge</td>
<td>$O(n/ε \log 1/ε)$</td>
<td>$O(1/ε \log n \log 1/ε )$</td>
<td>No</td>
<td>Weighted</td>
<td>[BGLP16]</td>
</tr>
</tbody>
</table>

Questions

1) Extension to Directed Graphs?? ✓

2) Can we make dependence on $ε$ to be $(1/ε)$? ✓

3) Distance Oracle for handling vertex failure when $G$ is weighted? ✓

4) A ‘path-reporting-oracle’ for handling vertex(weighted case) and edge failures? ✓
OUR CONTRIBUTION

Directed (or undirected) graphs and edge/vertex failure

(i) (1+ε) Preserver: size = \(\tilde{O}(n/\varepsilon \log M)\)
   - Each vertex has in-degree at most \(\tilde{O}(1/\varepsilon \log M)\)

(ii) (1+ε) Oracle: size = \(\tilde{O}(n/\varepsilon \log M)\) and time = \(O(1/\varepsilon \log \log M)\)
   - Reports path in time = \(O(1/\varepsilon \log \log M + \text{number\_of\_edges\_in\_path})\)

(iii) Distributed implementations: labeling & routing scheme.

(iv) **Space is optimal** upto logarithmic factors for directed graphs.

---

1 \(\tilde{O}(\cdot)\) hides poly-logarithmic factors.

2 \(M = \text{diameter of graph (assuming all edge weights are at least 1)}\).
Sketch of $(1+\varepsilon)$-Preserver:

**Detour of $y$:** A path $D$ that
(i) Starts from ancestor of $y$
(ii) Is internally disjoint with tree-path $T(s,y)$

If $(\text{wt}(D') \leq (1+\varepsilon) \text{wt}(D))$ then

$D'$ gives a $(1+\varepsilon)$ shortest path to $v$

For any $y$

we need not store all the detours
Sketch of $(1+\varepsilon)$-Preserver:

**Detour of $y$:** A path $D$ that
(i) Starts from ancestor of $y$
(ii) Is internally disjoint with tree-path $T(s,y)$

If $(wt(D') \leq (1+\varepsilon) wt(D))$ then

$D'$ gives a $(1+\varepsilon)$ shortest path to $v$

For any $y$

we need not store all the detours

Consider only Logarithmic number of detours
Sketch of \((1+\varepsilon)\)-Preserver:

For each \(\alpha = (1+\varepsilon)^i\)

\[\text{HD}_\alpha(y) = \text{Highest Detour, weight \leq \alpha}\]

A detour to \(y\)

(i) from highest possible ancestor of \(y\)

(ii) of weight at most \(\alpha\)

So \(H\) is a \((1+\varepsilon)\)-Preserver:

\[H = T \cup \{\text{HD}_\alpha(y) \mid y \in V, \alpha = (1+\varepsilon)^i\}\]

Space = \((n) \times (\log(1+\varepsilon) M) \times (n)\)

Consider only Logarithmic number of detours

Costly!!
Sketch of \((1+\varepsilon)\)-Preserver:

**Algorithm (parameter \(k\))**:

1. \(H \leftarrow T\);
2. \(S_1 \supseteq S_2 \supseteq \cdots \supseteq S_k \leftarrow \text{Subsets of } V \text{ sampled with probability } n^{-1/k}\);
3. For each vertex \(y\) and \(\alpha \in \text{POWERS}(1+\varepsilon)\)
   
   If \((y \in S_i)\)
   
   Add last \(n^{i/k}\) non-tree edges of \(HD_\alpha(y)\) to \(H\);

\[|S_1| = n \quad |S_2| = n^{1/4} \quad |S_3| = n^{2/4} \quad |S_4| = n^{3/4}\]

\[|E(H)| = \tilde{O}(k \ n^{1 + 1/k}) \text{ edges}\]

\(k=4, \text{ probability } = n^{-1/4}\)
Correctness of $(1+\varepsilon)$-Preserver:

Last $n^{3/4}$ vertices contains a sampled vertex of $S_3$
Correctness of $$(1+\varepsilon)$$-Preserver:

- Path from $x$ to $y_1$ is not a detour!
  (detours of $y_2, y_3, y_4$ can intersect tree-path($s, y_1$))
Correctness of $(1+\varepsilon)$-Preserver:

- Path from $x$ to $y_1$ is not a detour!
  (detours of $y_2, y_3, y_4$ can intersect tree-path(s,y_1))
Correctness of $(1+\varepsilon)$-Preserver:

- Path from $x$ to $y_1$ is not a detour! (detours of $y_2, y_3, y_4$ can intersect tree-path$(s, y_1)$)

- **Lemma:** $H$ contains **2-vertex disjoints paths** from $x$ to $y_1$ of length $\leq (1+\varepsilon)^4 \text{ wt}(D)$

\[ k \text{ levels: stretch } \leq (1+\varepsilon)^k \]
Results for \((1+\varepsilon)\)-Preserver:

**Upper Bound:**

For any directed weighted graph \(G\) with source \(s\), we can compute a \((1+\varepsilon)\)-preserver of size \(O(n/\varepsilon \log M \log^3 n)\) resilient to an edge or a vertex failure.

**Lower bound:**

There exists directed weighted graphs whose optimal \((1+\varepsilon)\)-preserver has size at least \(\Omega(n/\varepsilon \log M)\).

\(M =\) diameter of graph (assuming all edge weights are at least 1).
Finding \((1+\varepsilon)\)-Path:

- First compute vertices \(y_1, y_2, y_3, y_4\)

- **Observation** :
  For any \(y_i\)
  
  1. either detour of \(y_i\), or
  2. tree-path\((s, y_i)\)
  
  will NOT contain the failed vertex.

- **Algorithm** :
  
  1. Traverse tree-path\((y_1, \nu)\);
  2. For \(i = 1\) to \(k\):
     
     If partial detour of \(y_i\) is intact, traverse it;
     
     Else traverse tree-path\((s, y_i)\) and break;
  3. Traverse tree-path\((s, x)\);
Finding \((1+\varepsilon)\)-Path:

**Theorem:**

For any directed weighted graph \(G\) with source \(s\), we can compute an oracle of size \(O(n/\varepsilon \log M \log^3 n)\) that after any edge/vertex failure can report \((1+\varepsilon)\)-stretched shortest paths from \(s\) in time which is \(O(\text{number\_of\_edges\_in\_path} + 1/\varepsilon \log \log M)\).

\(M = \text{diameter of graph (assuming all edge weights are at least 1)}.\)
(i) $(1+\epsilon)$-Preserver of size $\tilde{O}(n/\epsilon \log M)$.

(ii) $(1+\epsilon)$-Oracle of size $\tilde{O}(n/\epsilon \log M)$ that reports
- $(1+\epsilon)$-dist in time $= O(1/\epsilon \log \log M)$.
- $(1+\epsilon)$-path in time $= O(\text{number_of_edges_in_path} + 1/\epsilon \log \log M)$.

(iii) **Space is optimal** upto logarithmic factors.

(iv) Labeling and routing scheme
OPEN QUESTIONS

(i) Extension of (1+\(\epsilon\))-Preserver and (1+\(\epsilon\))-Oracle to multiple failures.

(ii) Extension to multi-source case and SxT case.

(iii) For single failure:

- Improving the pre-processing time, currently it is \(O(\frac{mn}{\epsilon \log M \text{ polylog } n})\).
- Obtaining constant query time.