On Scalable Computation of Graph Eccentricities

Wentao Li, Miao Qiao, Lu Qin, Lijun Chang, Ying Zhang, Xuemin Lin

June 15, 2022

Eccentricity Distribution

Given a graph $G(V, E)$, the **eccentricity** of a node v:

$$ecc(v) = \max_{u \in V} dist(v, u).$$

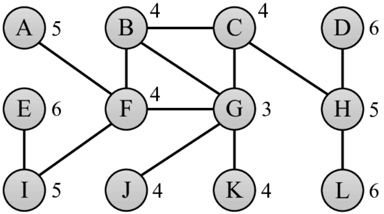
Eccentricity distribution is the eccentricities of **all** the nodes in the graph.

Eccentricity Distribution

Given a graph $G(V, E)$, the **eccentricity** of a node v:

$$ecc(v) = \max_{u \in V} dist(v, u).$$

Eccentricity distribution is the eccentricities of **all** the nodes in the graph.



Eccentricity Distribution

Given a graph $G(V, E)$, the **eccentricity** of a node v:

$$ecc(v) = \max_{u \in V} dist(v, u).$$

Eccentricity distribution is the eccentricities of **all** the nodes in the graph.

- Diameter $\max_{v \in V} ecc(v)$. Radius $\min_{v \in V} ecc(v)$.
- Node centrality measure: graph analysis, features for prediction.
- For each $v \in V$, compute single source shortest path. quadratic computation

Eccentricity Distribution

Given a graph $G(V, E)$, the **eccentricity** of a node v:

$$ecc(v) = \max_{u \in V} dist(v, u).$$

Eccentricity distribution is the eccentricities of **all** the nodes in the graph.

- Diameter $\max_{v \in V} ecc(v)$. Radius $\min_{v \in V} ecc(v)$.
- Node centrality measure: graph analysis, features for prediction.
- For each $v \in V$, compute single source shortest path. quadratic computation

Can we break the quadratic barrier?

Eccentricity Distribution

Given a graph $G(V, E)$, the **eccentricity** of a node v:

$$ecc(v) = \max_{u \in V} dist(v, u).$$

Eccentricity distribution is the eccentricities of **all** the nodes in the graph.

- Diameter $\max_{v \in V} ecc(v)$. Radius $\min_{v \in V} ecc(v)$.
- Node centrality measure: graph analysis, features for prediction.
- For each $v \in V$, compute single source shortest path. quadratic computation

Can we break the quadratic barrier? ✗

Eccentricity Distribution

Given a graph $G(V, E)$, the **eccentricity** of a node v:

$$ecc(v) = \max_{u \in V} dist(v, u).$$

Eccentricity distribution is the eccentricities of **all** the nodes in the graph.

- Diameter $\max_{v \in V} ecc(v)$. Radius $\min_{v \in V} ecc(v)$.
- Node centrality measure: graph analysis, features for prediction.
- For each $v \in V$, compute single source shortest path. quadratic computation

Can we break the quadratic barrier? ✗

- Any approach computing eccentricity distribution will either be quadratic or introducing a $\geq 1/3$ relative error under strong exponential hypotheses [Roditty and Williams, 2013].

Our Approach IFECC

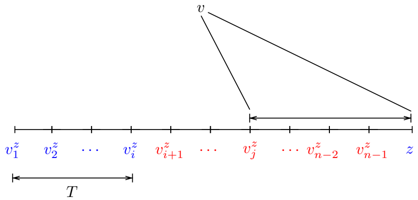
Theoretically refined (introducing new parameters), 69.9 x faster, $\frac{1}{37}$ space consumption

Our Approach IFECC

Theoretically refined (introducing new parameters), 69.9 x faster, $\frac{1}{37}$ space consumption

1. Let z be the node with the highest degree in G.
2. Compute $ecc(z)$ and the farthest first order $L^z = \langle v_1^z, v_2^z, \cdots, v_{n-1}^z \rangle$ of z
3. Update the upper ecc bound and lower ecc bound of every node v with
   $\underline{ecc}(v) = \max\{\underline{ecc}(v), dist(v, z), ecc(z) - dist(v, z)\}$
   $\overline{ecc}(v) = \min\{\overline{ecc}(v), ecc(z) + dist(v, z)\}$
4. For each v_i^z, perform BFS from v_i^z, and update, for each node v

$\underline{ecc}(v) = \max\{\underline{ecc}(v), dist(v, v_i^z)\}$
$\overline{ecc}(v) =$
$\min\{\overline{ecc}(v), \max\{\underline{ecc}(v), dist(v_i^z, z) + dist(z, v)\}\}$
if $\underline{ecc}(v) = \overline{ecc}(v)$ then $ecc(v)$ is finalized;



Terminates early when all nodes' eccs are finalized.

On Scalable Computation of Graph Eccentricities

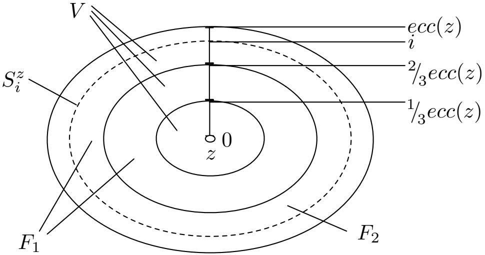


Figure: Stratified Graph from z

On Scalable Computation of Graph Eccentricities

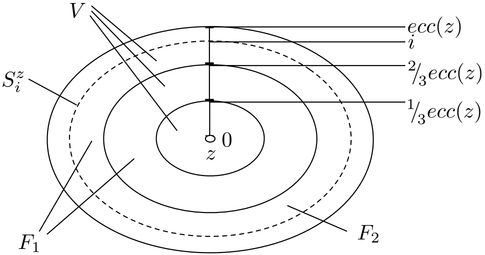


Figure: Stratified Graph from z

Theorems

1. $\widetilde{O}(|F_1|(m+n))$: Performing BFSs from nodes in F_1 computes the eccentricity distribution of the graph.
2. $\widetilde{O}(|F_2|(m+n))$: Performing BFSs from nodes in F_2 computes i) $ecc(v)$ for each v in F_2 and ii) an estimation $\widetilde{ecc}(v) = \max\{dist_{max}(v, F_2), dist(v, z) + \frac{1}{4}ecc(z)\}$ for each node $v \notin F_2$ such that $\frac{7}{12} \leq \frac{\widetilde{ecc}(v)}{ecc(v)} \leq \frac{3}{2}$.

Graph Data

Name	Dataset	n	m	Type
DBLP	DBLP	317,080	1,049,866	Social
GP	GPlus	201,949	1,133,956	Social
YOUT	Youtube	1,134,890	2,987,624	Social
DIGG	Digg	770,799	5,907,132	Social
SKIT	Skitter	1,694,616	11,094,209	Internet
DBPE	Dbpedia	3,915,921	12,577,253	Web
HUDO	Hudong	1,962,418	14,419,760	Web
TPD	UK-Tpd	1,766,010	15,283,718	Web
FLIC	Flickr	1,624,992	15,476,835	Social
BAID	Baidu	2,107,689	16,996,139	Web
TOPC	Topcats	1,791,489	25,444,207	Web
STAC	Stackoverflow	2,572,345	28,177,464	Contact
UK02	UK02	18,459,128	261,556,721	Web
ABRA	Arabic	22,634,275	552,231,867	Web
IT	IT-2004	41,290,577	1,027,474,895	Web
TWIT	Twitter	41,652,230	1,202,513,046	Social
FRIE	Friendster	65,608,366	1,806,067,135	Social
SK	SK	50,634,118	1,810,050,743	Web
UK07	UK07	104,288,749	3,293,805,080	Web
UKUN	UKUN	130,831,972	4,653,174,411	Web

The two parameters

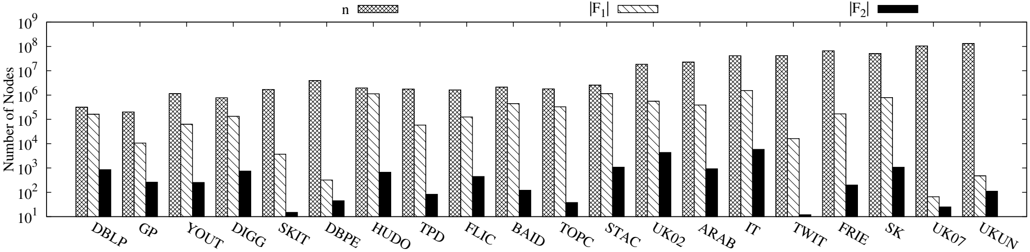


Figure: Size of $|F_1|$ and $|F_2|$

$|F_1|$ is on average $0.1n$, $|F_2|$ is on average $3 \times 10^{-4} n$.

Experimental Results

- Our approach: IFECC;
- The state-of-the-art: PLLECC [Li, et. al., 2018];
- Baseline: BoundECC [Takes and Kosters, 2013].

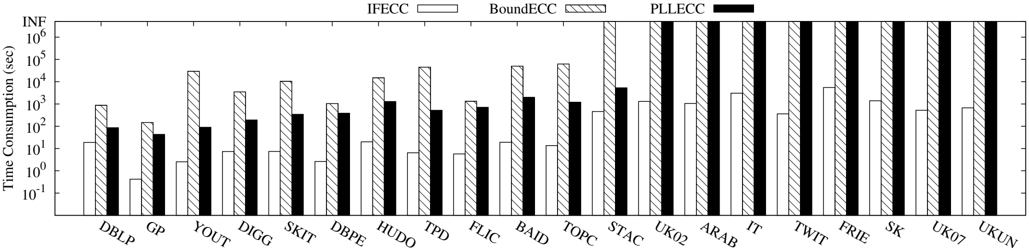


Figure: Exact ED Computation Efficiency

- IFECC completes the computation in 2h on UKUN (4.6 billion edges) and is
  - 69.9x faster than PLLECC on the first 12 graphs,
  - 2675.3x faster than BoundECC on the first 11 graphs.

Experimental Results

- Our approach: IFECC
- The state-of-the-art: PLLECC [Li, et. al., 2018]

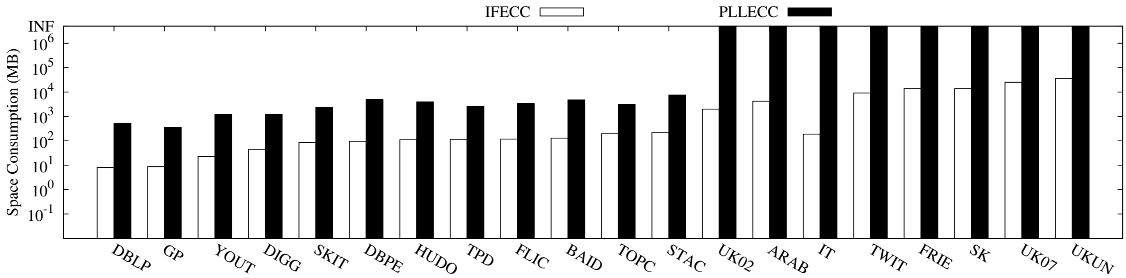


Figure: Exact ED Space Comparison

- IFECC's space consumption is 1/37 that of PLLECC.

Conclusions

IFECC: a concise algorithm for exact eccentricity distribution computation

- Theoretical analysis
- Extensive experiments
  - Time efficient: $69.9\times$ faster than PLLECC
  - Space efficient: $1/37$ that of PLLECC
  - Scalability: billion-scale networks
- Approximation adaption