PAGERANK ON AN EVOLVING GRAPH

Bahman Bahmani(Stanford)
Ravi Kumar(Google)
Mohammad Mahdian(Google)
Eli Upfal(Brown)

Present by

Yanzhao Yang

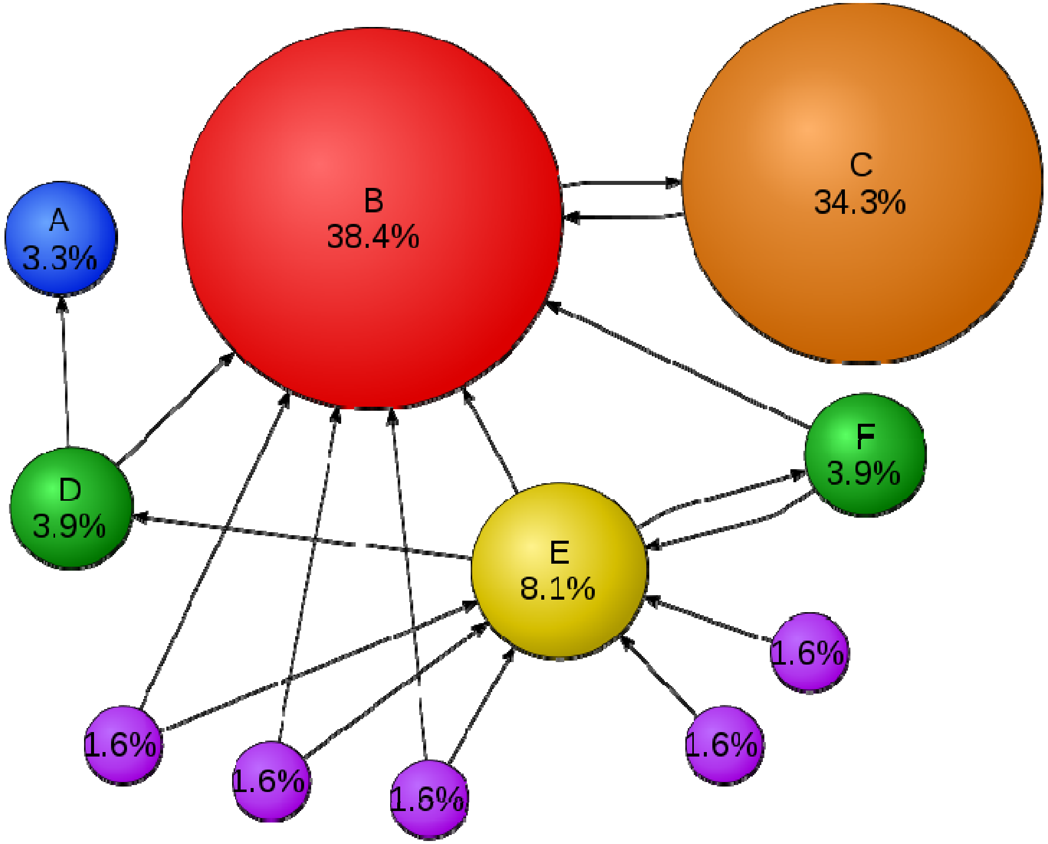
Evolving Graph(Web Graph)

- The directed links between web pages
- Used for computing the PageRank of the WWW pages [4]

Page Rank

- Classic link analysis algorithm based on the web graph
- A page that is linked to by many pages receives a high rank itself. Otherwise, it receives a low rank.
- The rank value indicates an importance of a particular page. [5]
- Very effective measure of reputation for both web graphs and social networks.

Example



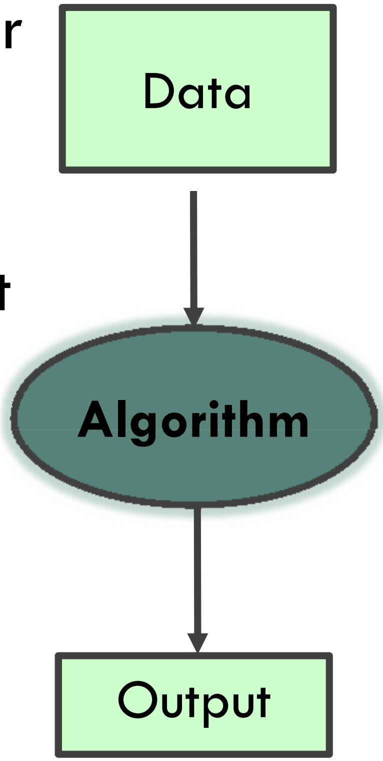
Problem

- Traditional algorithm paradigm is inadequate for evolving data

Traditional Paradigm

6

- Stationary dataset input- inadequate for current social networks
- It is necessary for algorithm to probe the input continually and produce solutions at any point in time that are close to the correct solution for the then-current input.



Motivating examples

- Web pages
  - Millions of hyperlinks modified each day
  - Which portions of the web should a crawler *focus* most?
- Social networks
  - Millions of social links modified each day
  - Which users should a third-party site track in order to *recompute*, eg, global reputation?

Motivating examples

- In fact, Pagerank may be always imprecise.

  e.g. Learn about changes->

  crawling webs->

  limit of crawling capacity->

  stale image of graph ->

  graph structure->

  Pagerank

Objective Algorithm

- Design an algorithm that decides which pages to crawl and computes the PageRank using the obtained information
- Maintains a good approximation of the true PageRank values of the underlying evolving graph
- Which pages to crawl
- The error is bounded at any point in time

Page Rank algorithm categories

- Linear algebraic methods[3]

  -Power iteration speed up.

  E.g, web graph.
- Monte carlo methods[6]

  -efficient and highly scalable

  E.g, data streaming anfd map reduce.

Evolving graph model

- A sequence of directed graphs over time
  - $G_t = (V, E_t)$ = graph at time t
  - Nodes do not change (for simplicity)
- Assume $|E_{t+1} - E_t|$ is small
  - Choose t fine enough
  - No change model assumed
- At time t, algorithm can probe a node u to get N(u), i.e, all edges in E_t of the form (u, v)
- No constraints on running time or storage space
- Probing strategy focus on which node to probe each time

PageRank on evolving graphs

- Teleport probability-ε
  - Probability of jumping to a random node
- Stationary distribution of random walk:

  -walk with ε: move to a node chosen uniformly at random

  -walk with 1-ε:choose one of the outgoing edges of the current node uniformly at random and move to the head of that edge
- π_u^t is PageRank of node u in G
- in_u^t is in-degree of node u
- out_u^t is out-degree of node u

Baseline probing methods

- Random probing(randomized)

Probe a node v chosen *uniformly* at random at each time step

- Round-robin probing(deterministic)

Cycle through all nodes and probe each in a round-robin manner

We can *vary* the ratio of change rate and probing rate

Propotional Probing

- At each step t, pick a node v to probe with probability proportional to the PageRank of v in the algorithm's current image of the graph.
- The output is the PageRank vector of the current image.

Priority Probing

Algorithm 1 Priority Probing

for all nodes u **do**
  $\text{priority}_u \leftarrow 0$
for every time step t **do**
  $v \leftarrow \arg\max_u \text{priority}_u$
  Probe v
  Let H^t be the current image of the graph
  Output the PageRank vector ϕ^t of H^t
  $\text{priority}_v \leftarrow 0$
  for all nodes $u \neq v$ **do**
    $\text{priority}_u \leftarrow \text{priority}_u + \phi_u^t$

for all node u do
$\text{Priority}_u \leftarrow 0$
for every time step t do

Experiment

16

- Dataset
  - AS(Autonom ous Systems, graph of routers)
  - CAIDA(communication patterns of the routers)
  - RAND (generated randomly)

Dataset	max #nodes (n)	#initial edges	#temporal edges	%edge additions
AS	7,716	10,696	488,986	0.516
CAIDA	31,379	65,911	1,084,388	0.518
RAND	100	715	250,000	0.5

Table 1: Details of the datasets used.

Experiment

- Random Probing serves as a baseline for Proportional Probing
- Round-Robin serves as a baseline for Priority Probing
- Hybird algorithm between Proportional Probing and Round-Robin Probing is parametrized by
- Metric

$$\beta \in [0,1]$$

$$\mathrm{L}_\infty \text{ metric } \mathrm{L}_\infty(\pi^{\mathrm{t}}, \phi^t) = \max_{u \in V} |\pi^t(u) - \phi^t(u)|$$

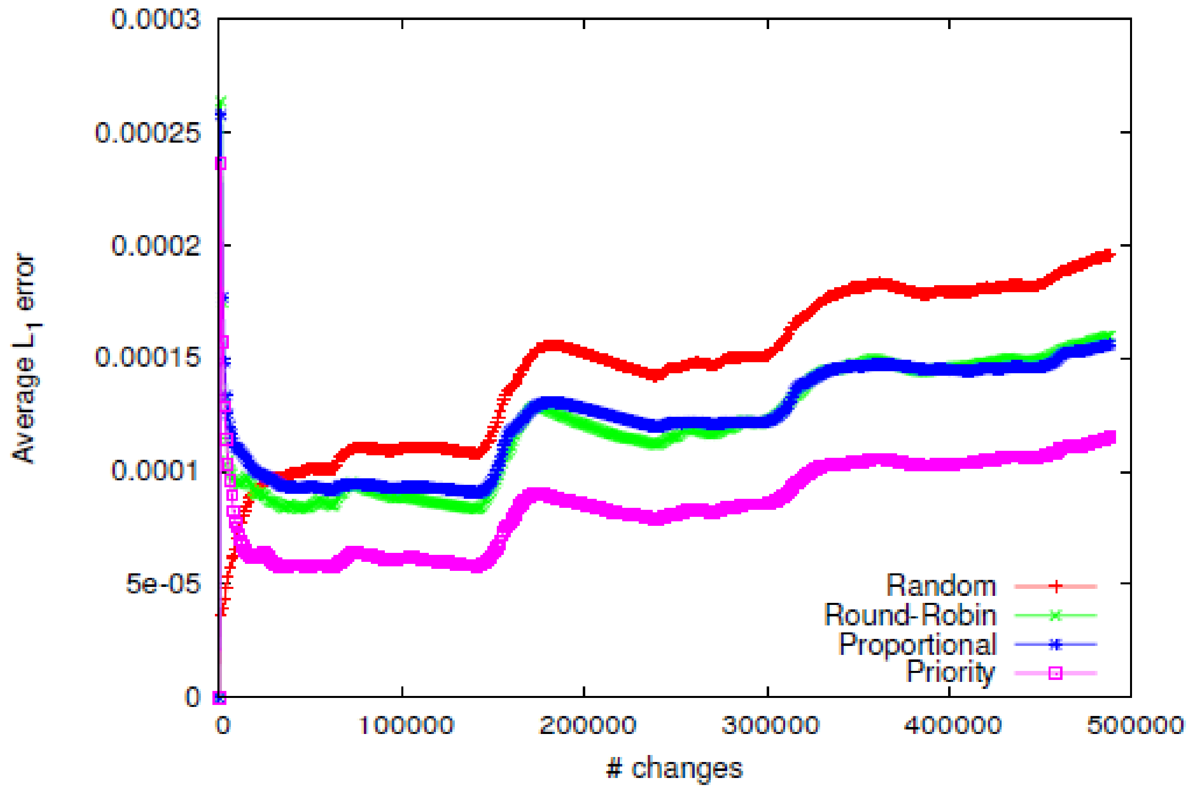
$$\mathrm{L}_1 \text{ metric } \mathrm{L}_1(\pi^{\mathrm{t}}, \phi^t) = \Sigma_{u \in V} |\pi^t(u) - \phi^t(u)|$$

Results(AS & CAIDA)

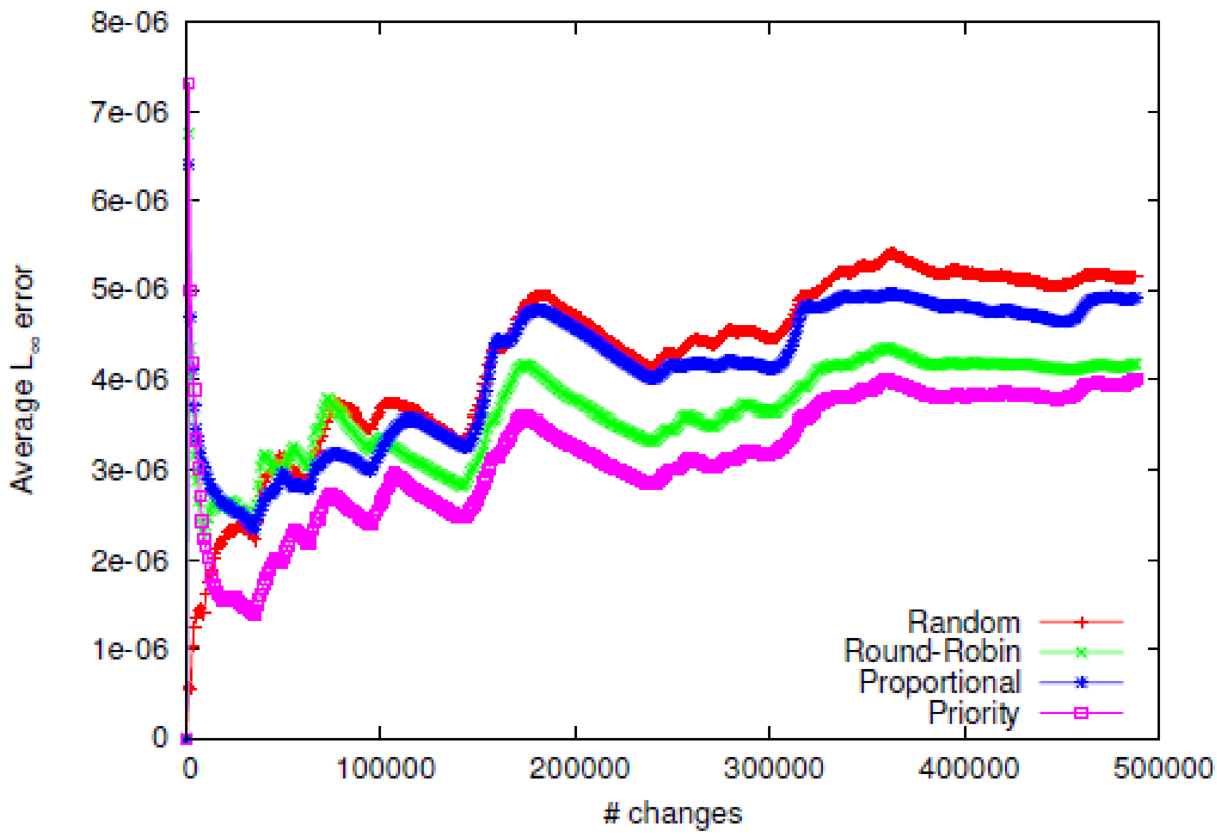
- Propotional Probing is better than Random Probing
- Priority Probing is better than Round-Robin Probing
- The algorithm perform better when they probe more frequently

AS graph (L1 errors)

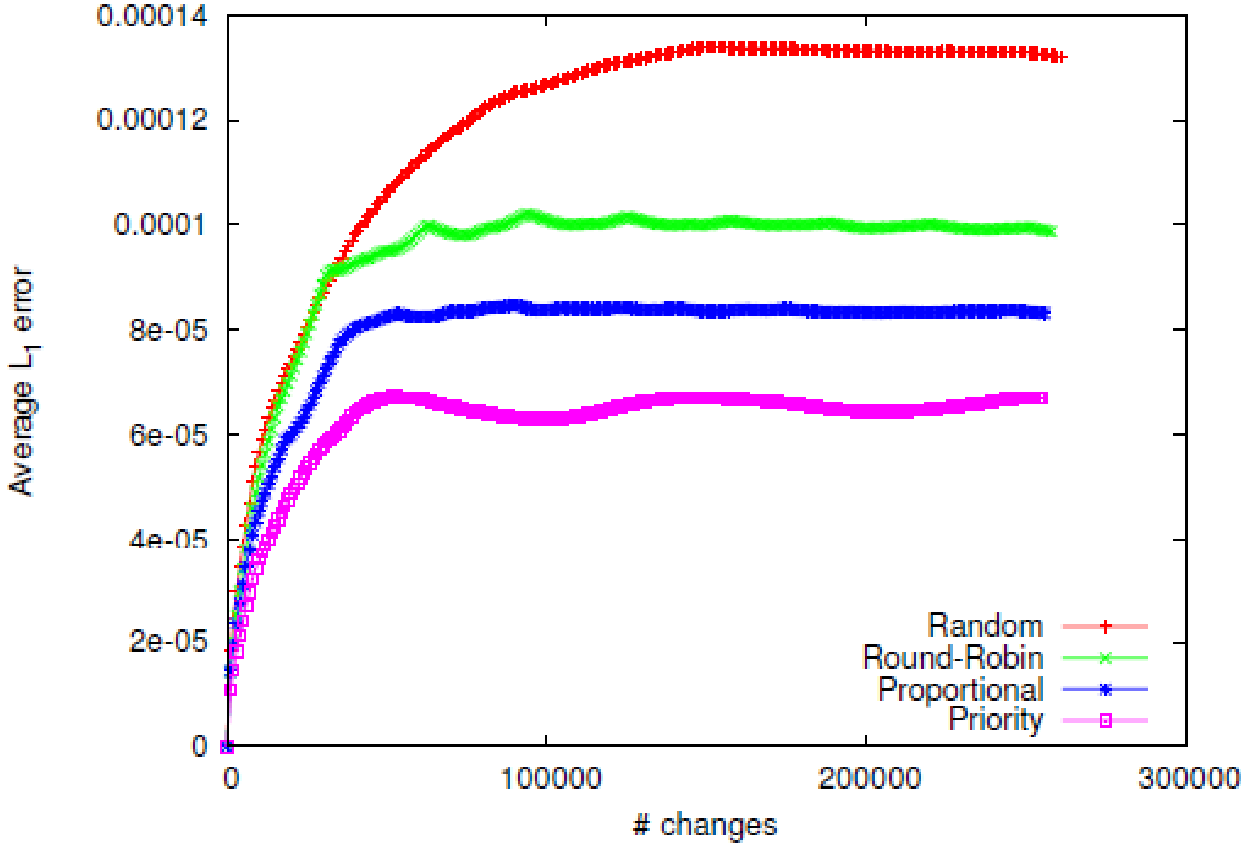
19



AS graph (L∞ errors)

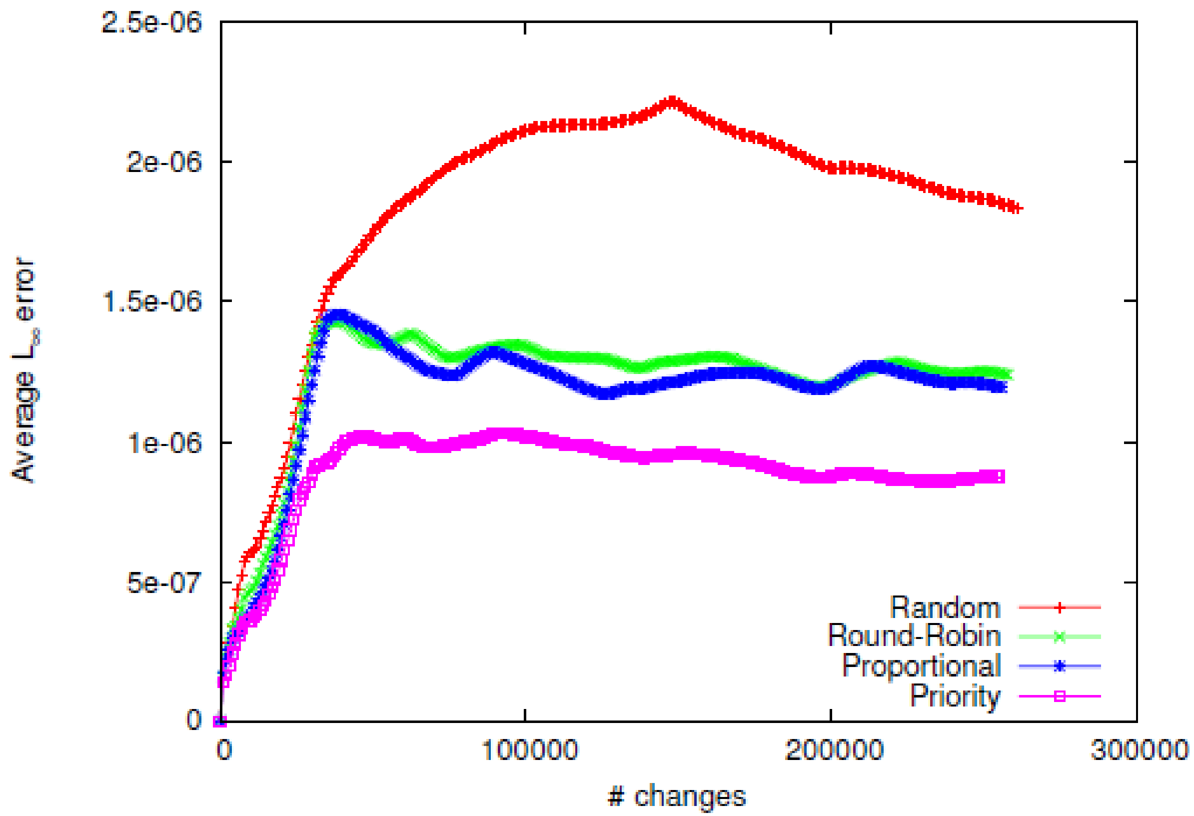


CAIDA graph (L1 errors)



CAIDA graph (L∞ errors)

22



Effect of probing rate α

23

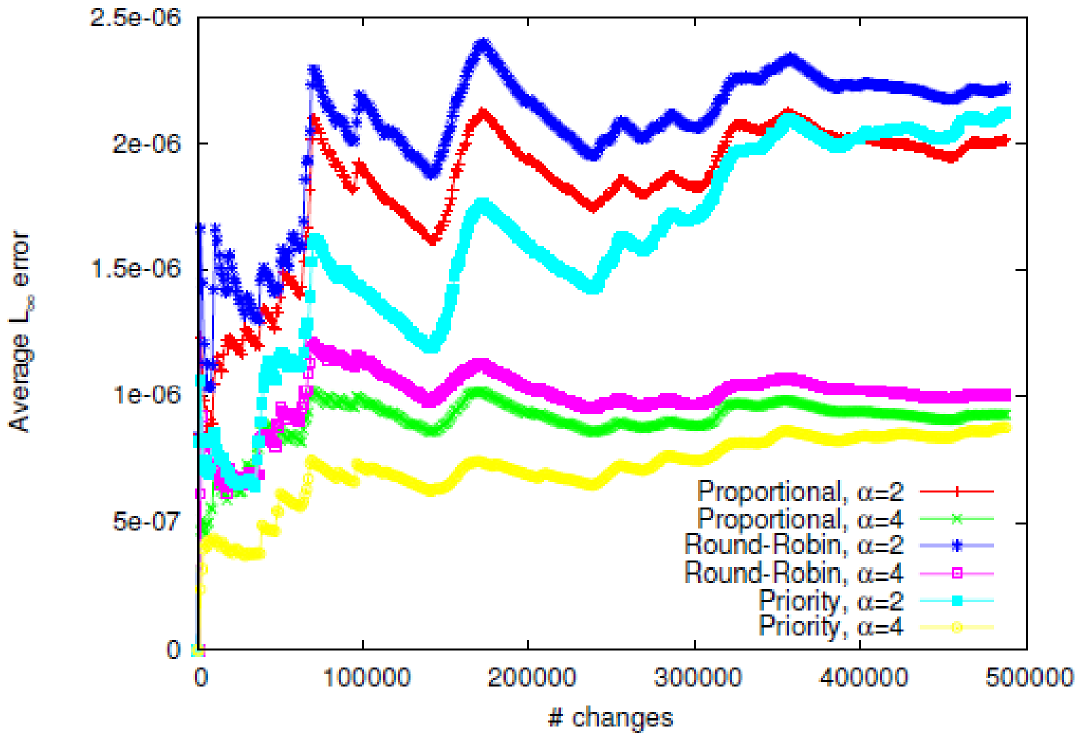
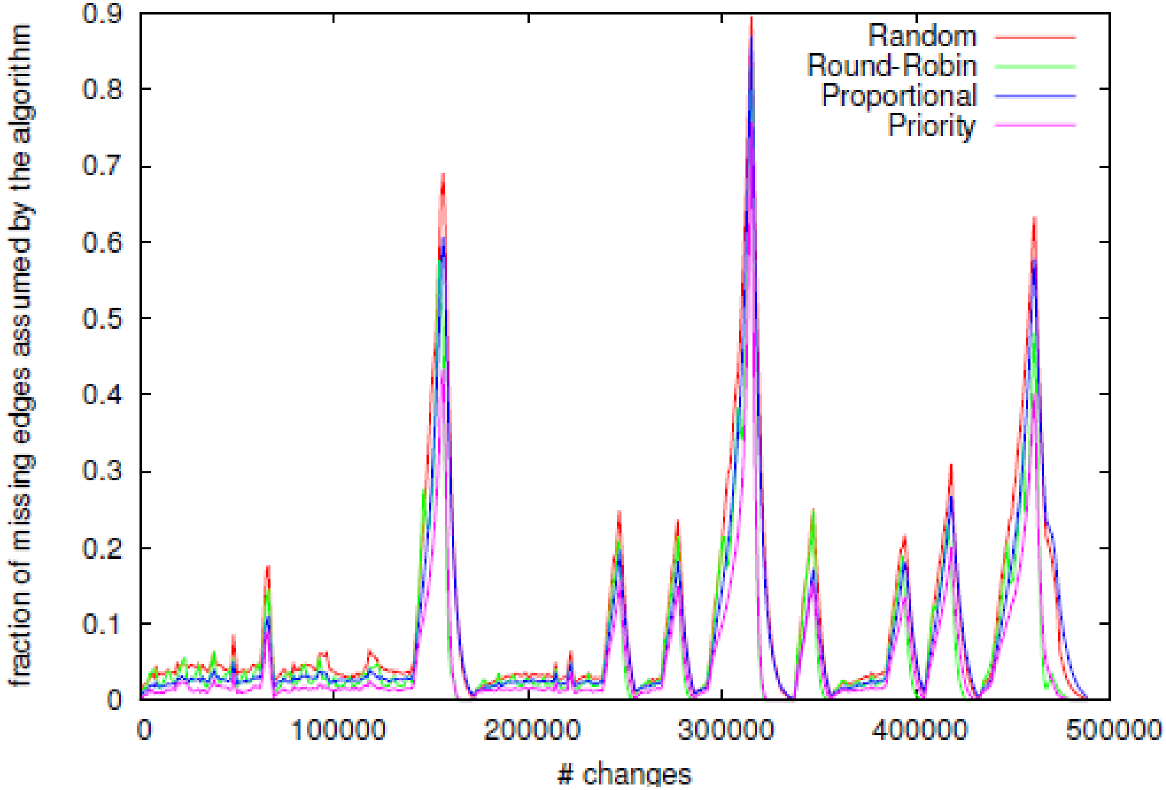


Figure 4: Average L_∞ error of Round-Robin, Proportional, and Priority as a function of the probing rate α for the AS graph.

Algorithm's image vs truth(1)

24



Algorithm's image vs truth(2)

25

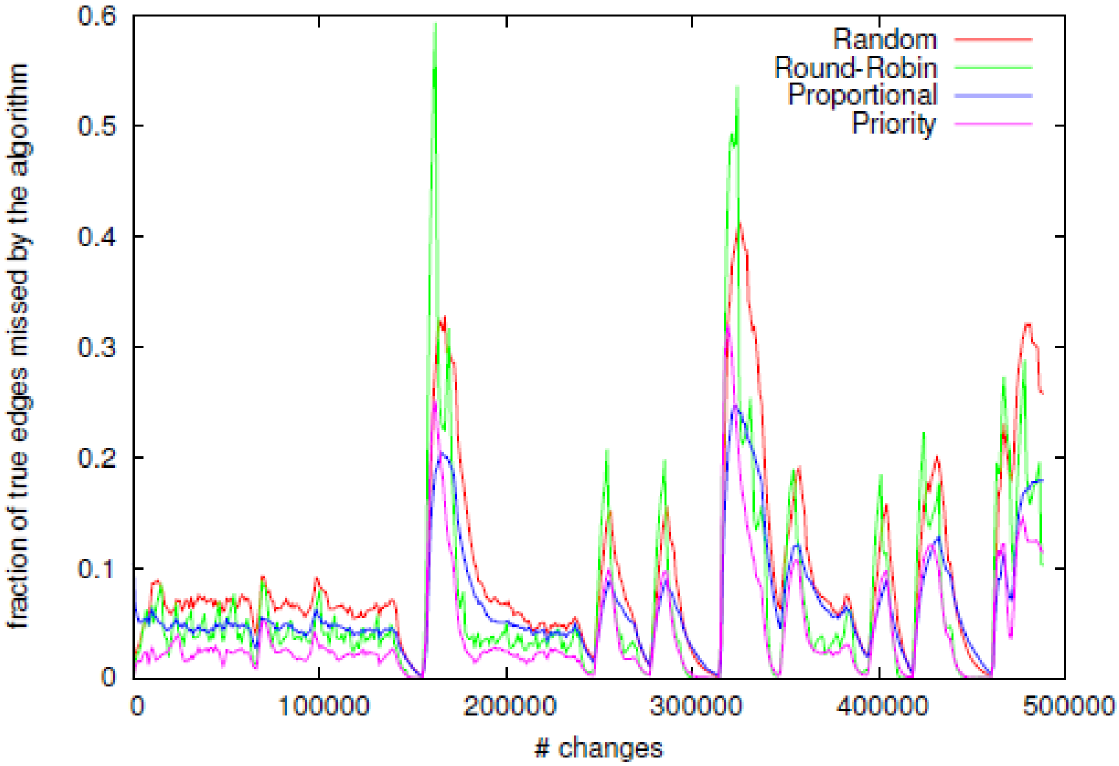
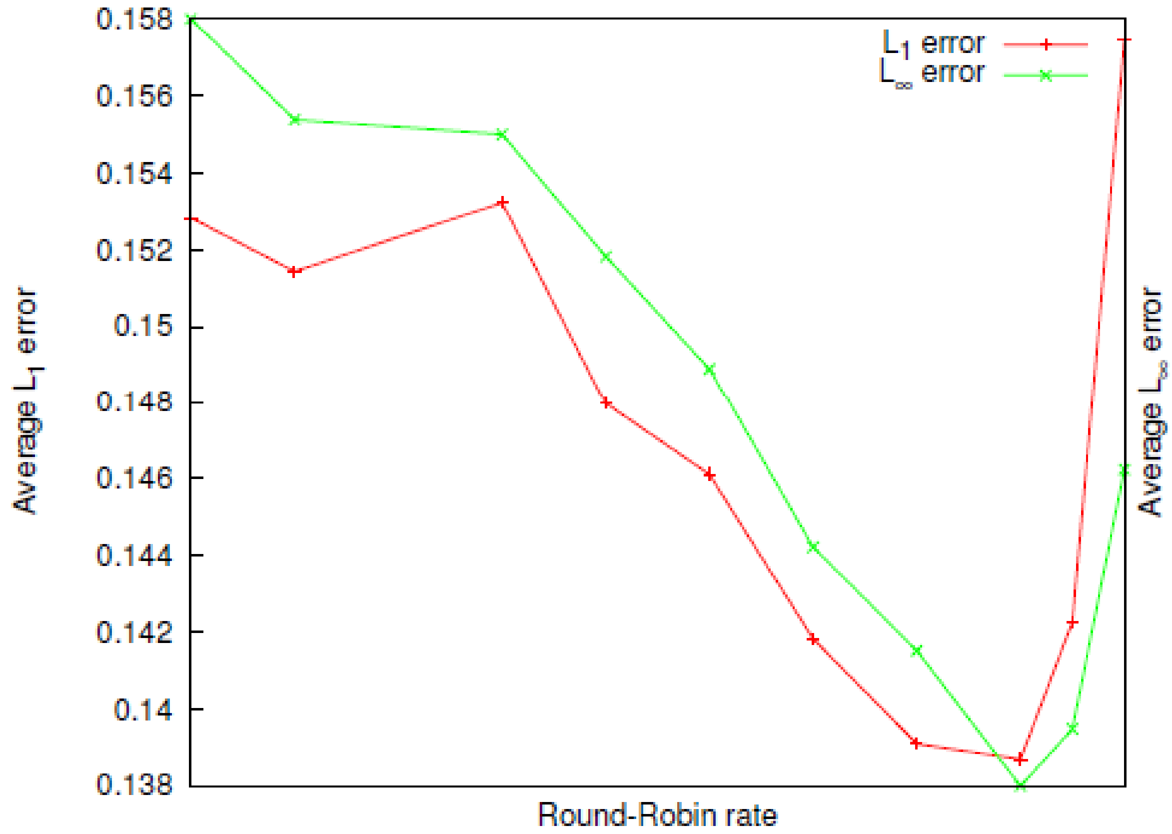


Figure 5: Staleness of the image of algorithms for the AS graph.

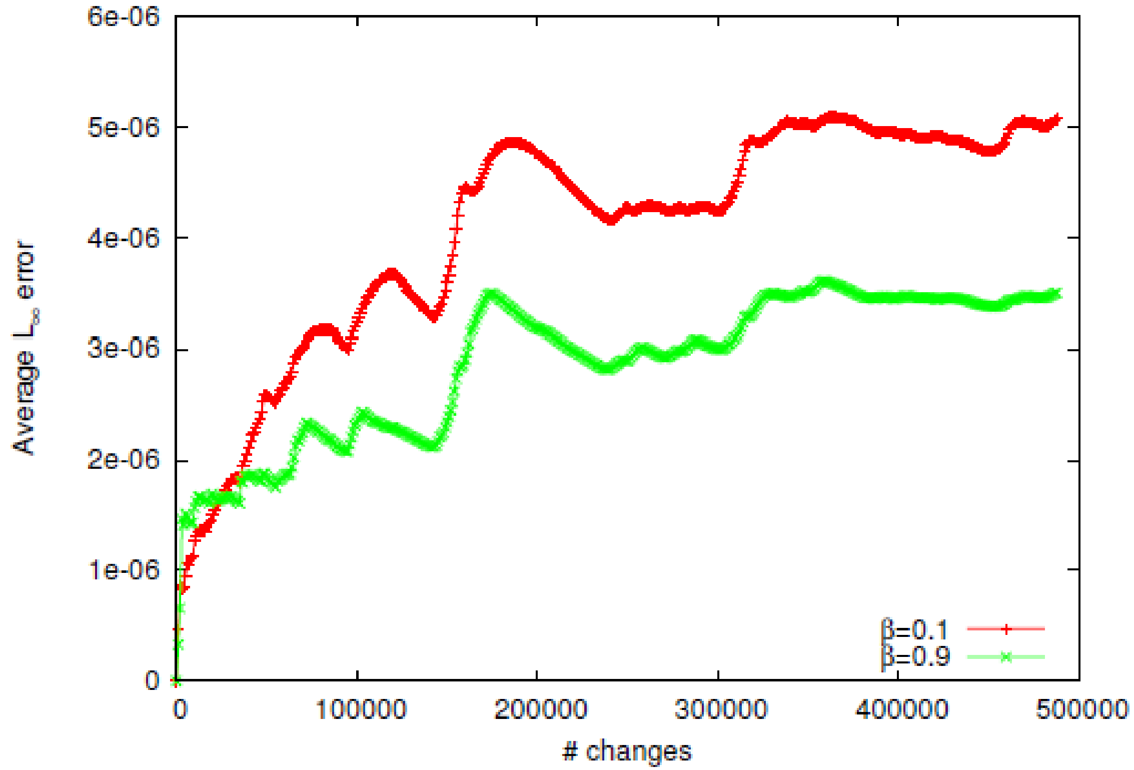
Hybird Algorithm (L1 &L∞)

26



Hybird Algorithm (β=01. or 0.9)

27



Analysis(1)

LEMMA 1. *Let $D(\pi^{t+1}, \pi^t)$ be the total variation distance between π^{t+1} and π^t. Then,*

$$E[D(\pi^{t+1}, \pi^t)] \leq \frac{1-\epsilon}{m\epsilon}.$$

LEMMA 2. *The expected PageRank of any node x at time $t+1$, conditioned on the graph at time t, satisfies*

$$\pi_x^t \left(1 - \frac{1}{\epsilon^2 m}\right) \leq E[\pi_x^{t+1} \mid G^t] \leq \pi_x^t \left(1 + \frac{1}{\epsilon^2 m}\right).$$

COROLLARY 3. *For any node x, time t, and time difference $\tau > 0$:*

$$\left(1 - \frac{1}{\epsilon^2 m}\right)^\tau \pi_x^t \leq E[\pi_x^{t+\tau} \mid G^t] \leq \left(1 + \frac{1}{\epsilon^2 m}\right)^\tau \pi_x^t.$$

Analysis(2)

THEOREM 4. *For a time instance t, assume that there exists an $\alpha > 0$ such that for all nodes $v \in V$ and all $t-2n \leq \tau \leq t-1$:*

$$(1-\alpha)\phi_v^\tau \leq E[\pi_v^\tau \mid G^{\tau-1}, H^\tau] \leq (1+\alpha)\phi_v^\tau.$$

Then, letting $\beta = (1-\epsilon)\frac{1+\alpha}{m}(1+\frac{1}{\epsilon^2 m})^{2n}$, we have for all $v \in V$:

$$(1-\beta)\phi_v^t \leq E[\pi_v^t \mid G^{t-1}, H^t] \leq (1+\beta)\phi_v^t.$$

COROLLARY 5. *In the steady state*

$$\left(1-O\left(\frac{1}{m}\right)\right)\phi^t \leq E[\pi^t \mid G^{t-1}, H^t] \leq \left(1+O\left(\frac{1}{m}\right)\right)\phi^t.$$

Conclusion

- Obtain simple effective algorithm
- Evaluate algorithms empirically on real and randomly generated datasets.
- Proved theoretical results in a simplified model
- Analyze the theoretical error bounds of the algorithm
- Challenge: extend our theoretical analysis to other models of graph evolution.

Reference

- 1. S. Brin, L. Page, Computer Networks and ISDN Systems 30, 107 (1998)
- 2. Glen Jeh and Jennifer Widom. 2003. *Scaling personalized web search*. WWW '03 http://doi.acm.org/10.1145/775152.775191
- 3. Lawrence Page, Sergey Brin, Rajeev Motwani, Terry Winograd. *The PageRank Citation Ranking: Bringing Order to the Web.*
- 4. http://en.wikipedia.org/wiki/Webgraph
- 5. http://en.wikipedia.org/wiki/PageRank#cite_note-1
- 6. K. Avrachenkov, N. Litvak, D. Nemirovsky, and N. Osipova. *Monte Carlo methods in Pagerank computation: When one iteration is sucient.* SIAM J.Numer. Anal., 45(2):890-904, 2007.

Thank you