PageRank in Apache Flink

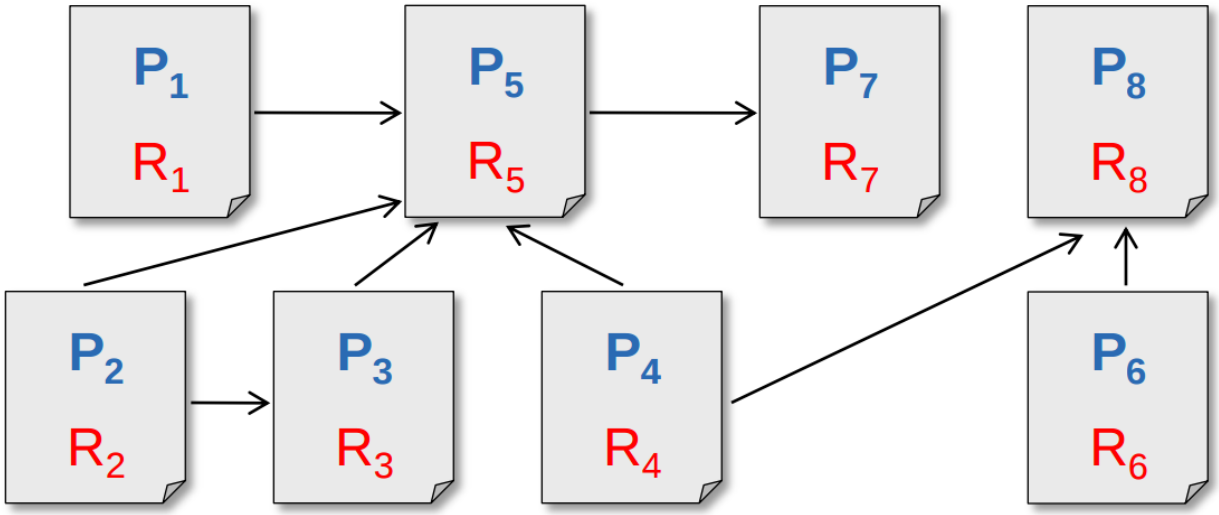
Author: Ward Schodts
Supervisor: Juan Soto

Datenbanksysteme und Informationsmanagement
Technische Universität Berlin

July 8, 2016

Agenda

Pagerank



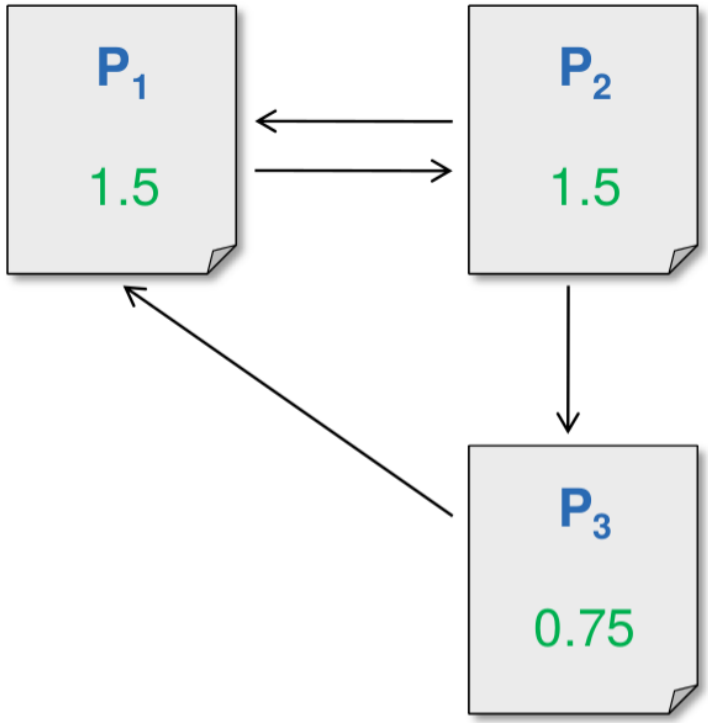
PageRank example 1 [6]

- A page has a high PageRank R if
  - there are many pages linking to it
  - or, if there are some pages with a high PageRank linking to it

PageRank

$$R(P_i) = \sum_{P_j \in B_i} \frac{R(P_j)}{L_j}$$

- where
  - B_i is the set of pages that link to page P_i
  - L_j is the number of outgoing links for page P_j linking to it



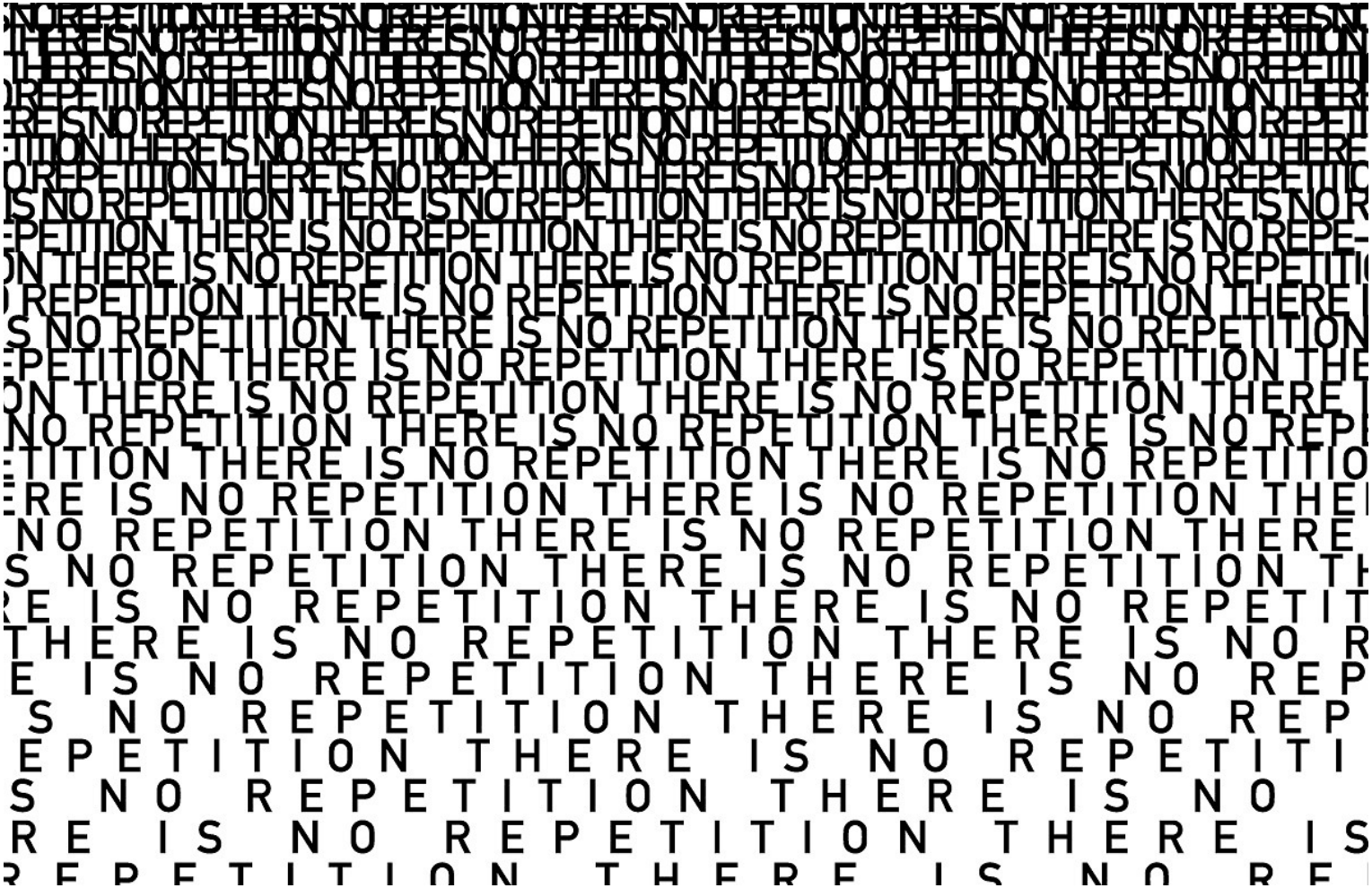
PageRank example 2 [6]

[4]

Apache Flink

- Open source framework for distributed Big Data Analytics
- Exploits:
  - data streaming
  - in-memory processing
  - iteration operators

  to improve performance
- Formerly Stratosphere (Flink means agile)
- Developed here at TU Berlin

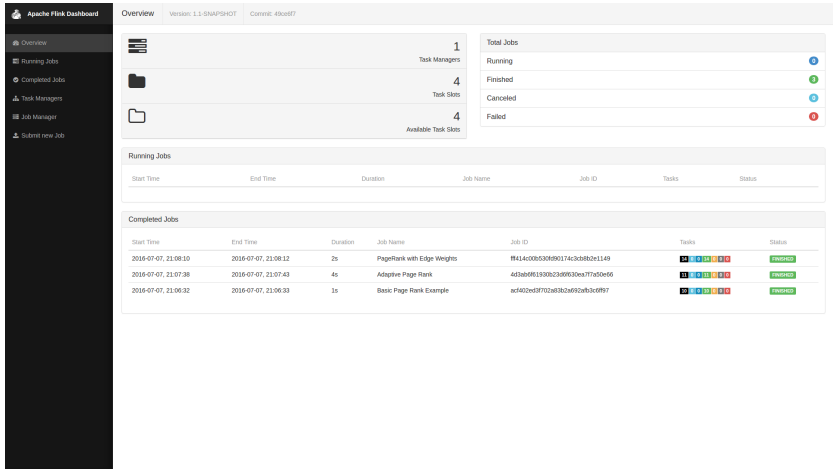


[7]

Apache Flink: 2 possible setups

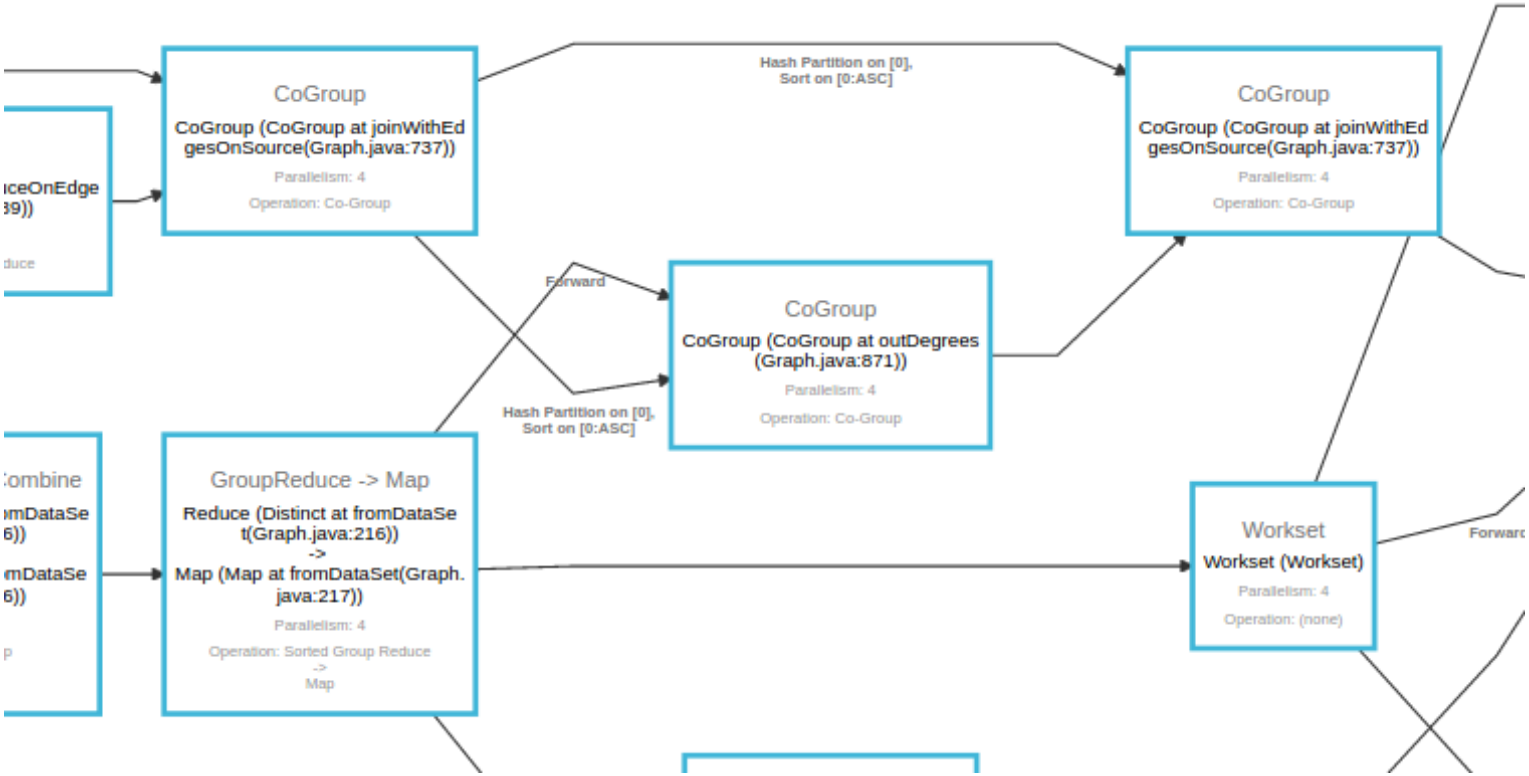
```xml
<dependencies>

    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-java</artifactId>
        <version>${flink.version}</version>
    </dependency>


    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-streaming-java_2.10</artifactId>
        <version>${flink.version}</version>
    </dependency>


    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-gelly_2.10</artifactId>
        <version>${flink.version}</version>
    </dependency>

    <dependency>
        <groupId>org.apache.flink</groupId>
        <artifactId>flink-table_2.10</artifactId>
        <version>${flink.version}</version>
    </dependency>
```

Maven



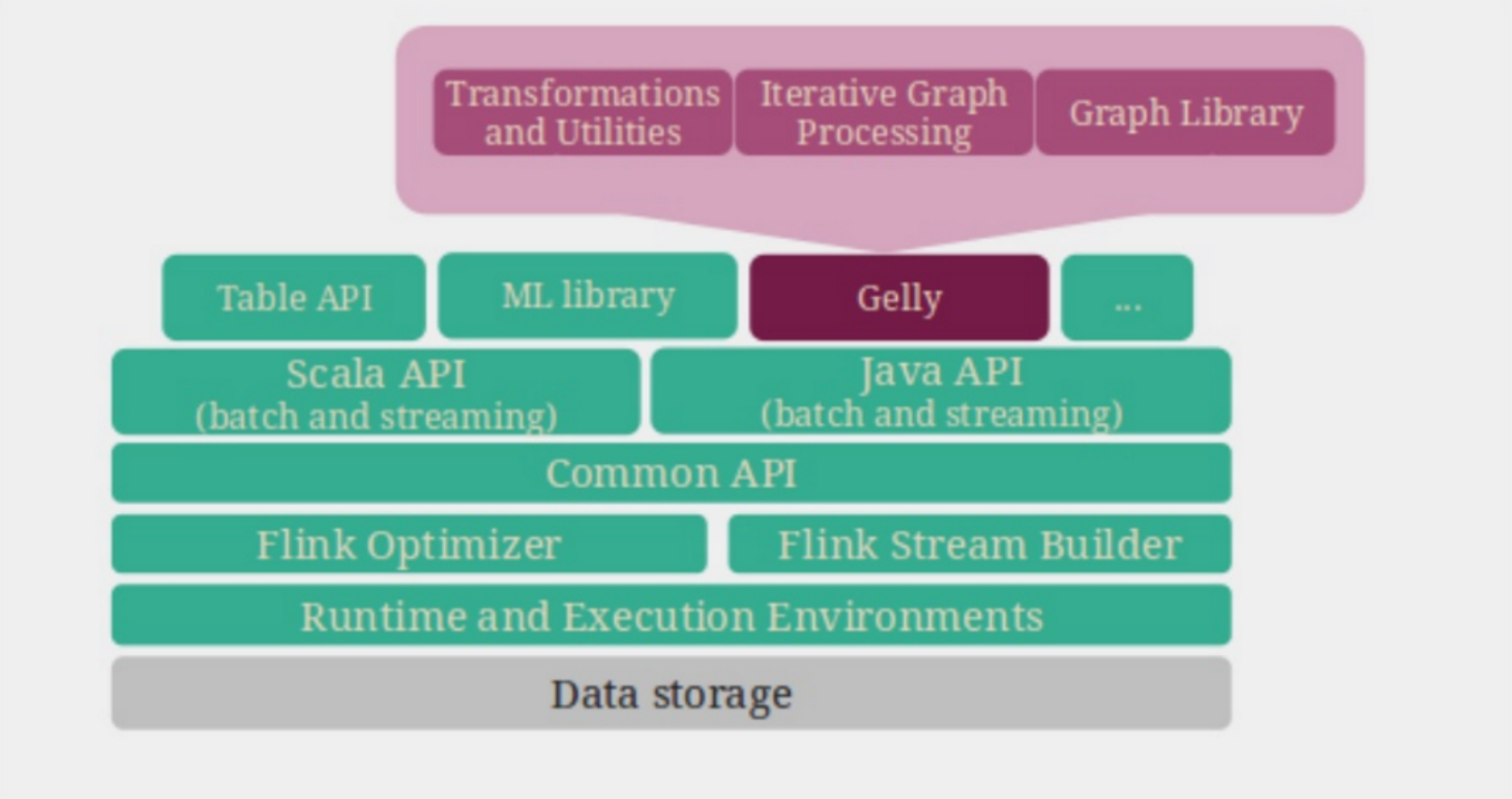
Binary version (self compiled)

Demo



Visualisation of a Flink job

Apache Flink: Gelly



Gelly

Apache Flink: Gelly

- Large-scale graph processing API
- On top of Flink's Java API
- Off-the shelf library methods (e.g. PageRank)
- Iterative algorithms

Agenda

General experiment setup

Experiment 1:

1. Data file with graph (and PageRank solution)

   ⇓

2. Use Flink and Graphlab implemention to compute PageRank

   ⇓

3. Compare with solution

General experiment setup

Experiment 1:

1. Data file with graph (and PageRank solution)

   ⇓

2. Use Flink and Graphlab implemention to compute PageRank

   ⇓

3. Compare with solution

Experiment 2:

1. Data file with huge graph (no solution yet)

   ⇓

2. Use Flink and Graphlab implemention to compute PageRank

   ⇓

3. Compare with each other

Experiment 1 data

Data from a former Hadoop toolkit (Cloud9, now Bespin):

Name	# vertices	# edges
Small	93	195
Medium	316	430
Large	1458	3545

Experiment 2 data

Webgraph from `snap.stanford.edu/data/`

Name	# vertices	# edges
web-Google	875713	5105039

Agenda

Flink algorithm 1

dataArtisans logo, [1]

- An exercise from dataArtisans
- Uses the standard Gelly implementation
- # input nodes = # output nodes

Flink algorithm 2

dataArtisans

- A case study implementation from dataArtisans
- A custom implementation
- # input nodes = # output nodes

Flink algorithm 3

- An example from the Apache Flink repository
- A custom implementation
- # input nodes != # output nodes $\rightarrow$ filters

Turi PageRank algorithm

Turi logo, [8]

- Used the standard implementation
- Builds a graph out of the edges dataset

Comparing the algorithms

As part of the experimental setup, I implemented a test harness to compare the two PageRank solutions.

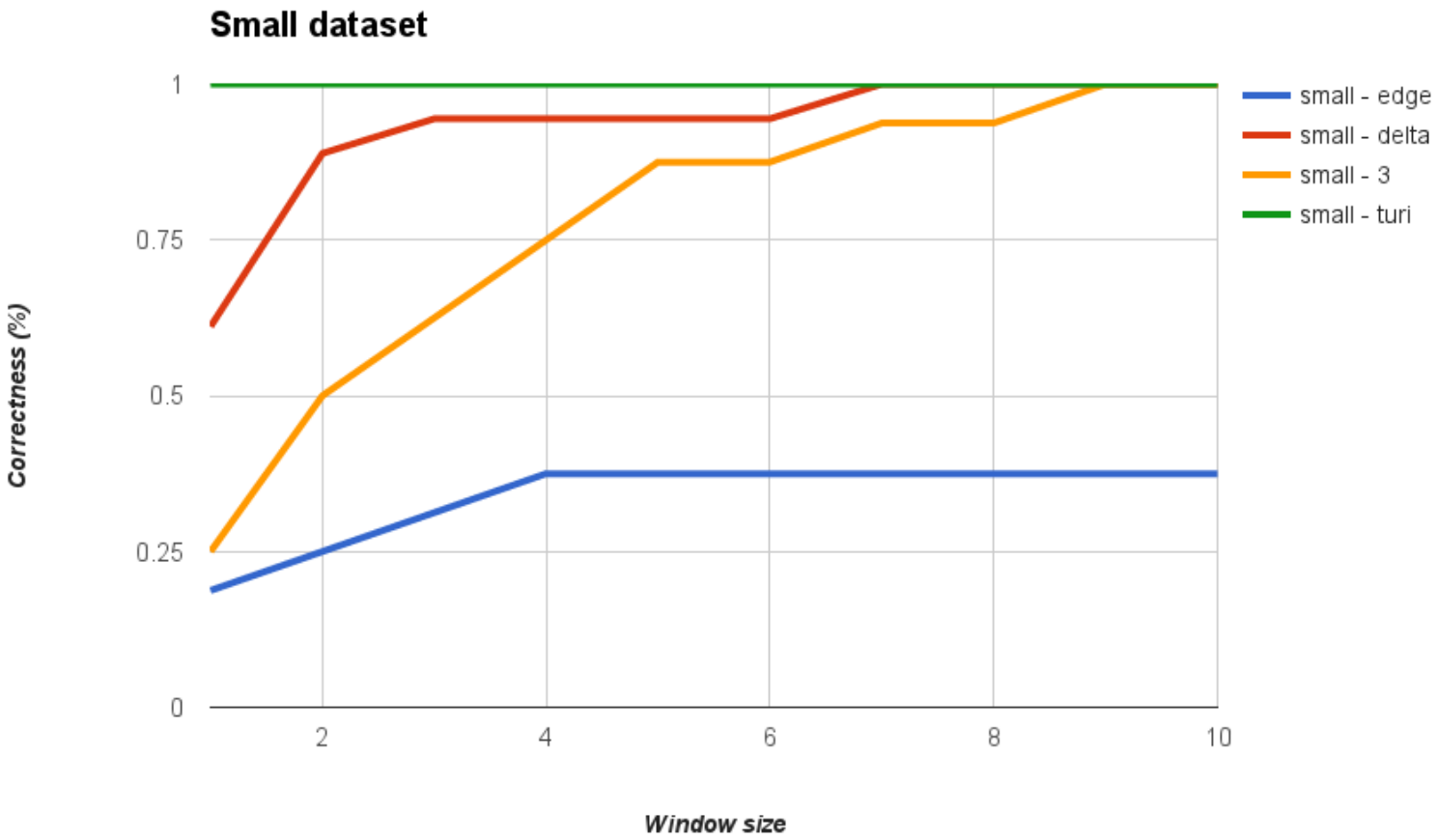
- It can handle list of diffrence sizes.
- It takes care of equal PageRank values (they maybe sorted in different way).
- Has a modifiable window to compare solutions.

Agenda

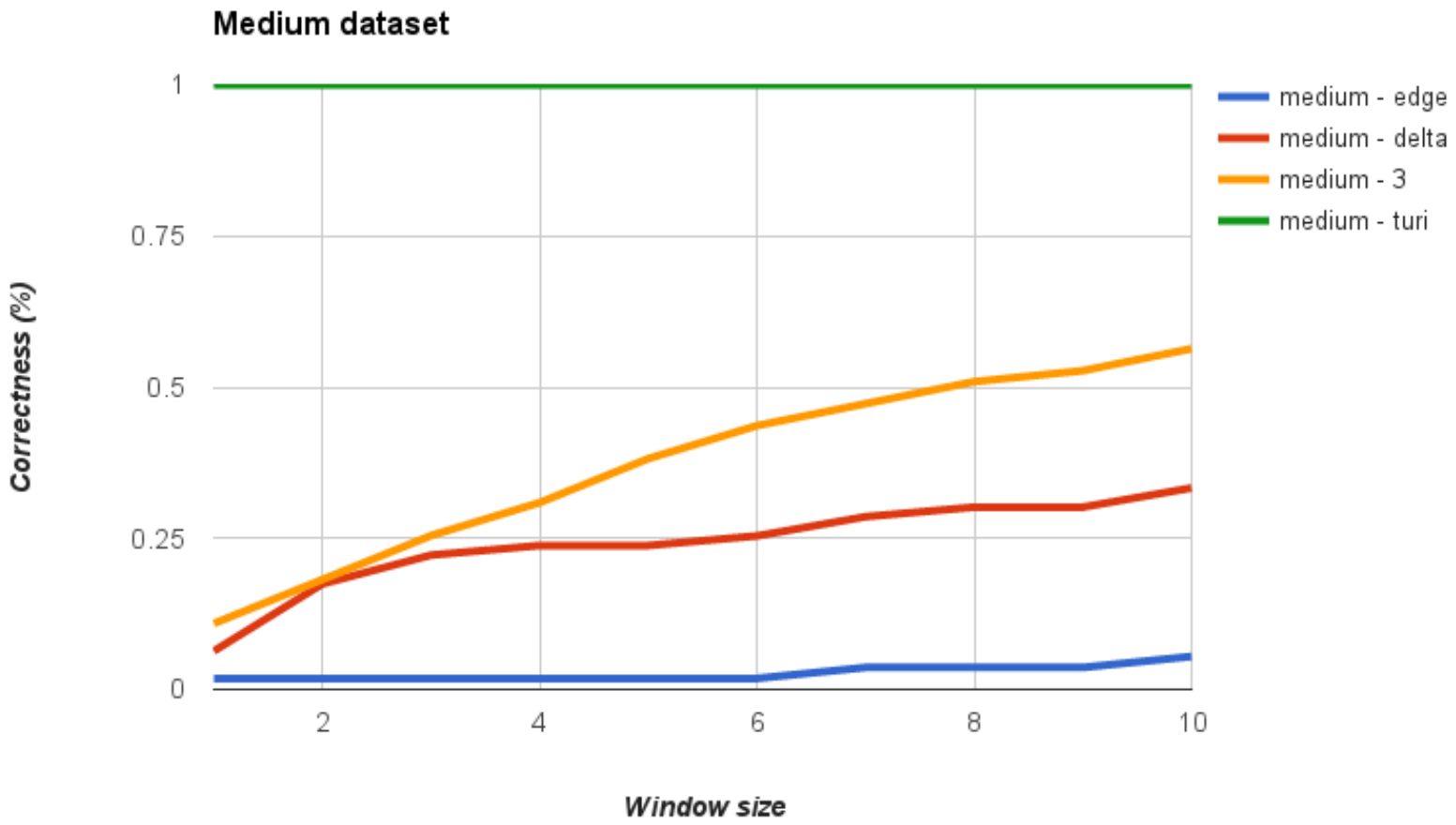
Results: experiment 1

Any expectations?

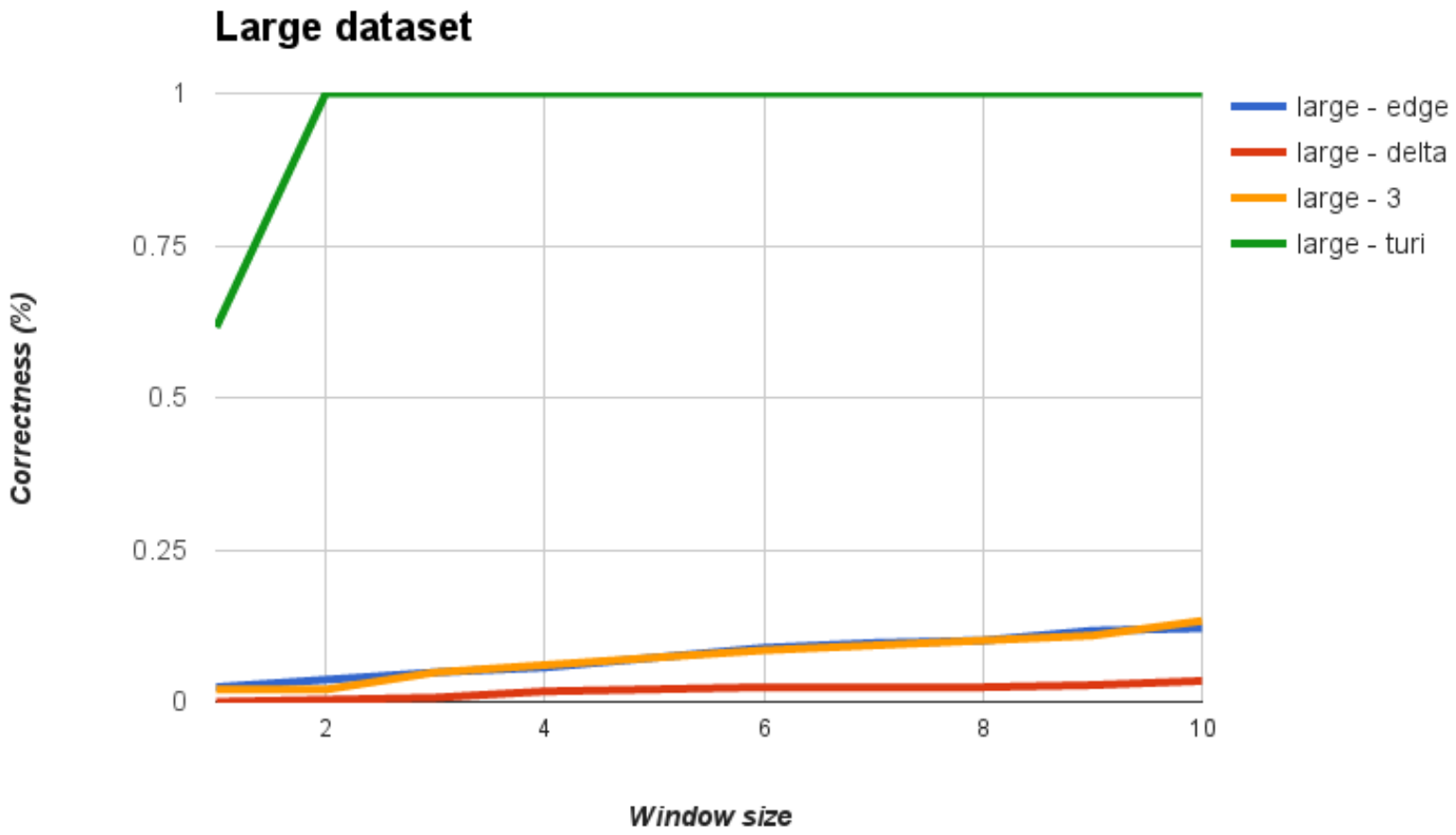
Results: experiment 1



Results: experiment 1



Results: experiment 1



These results are ...?

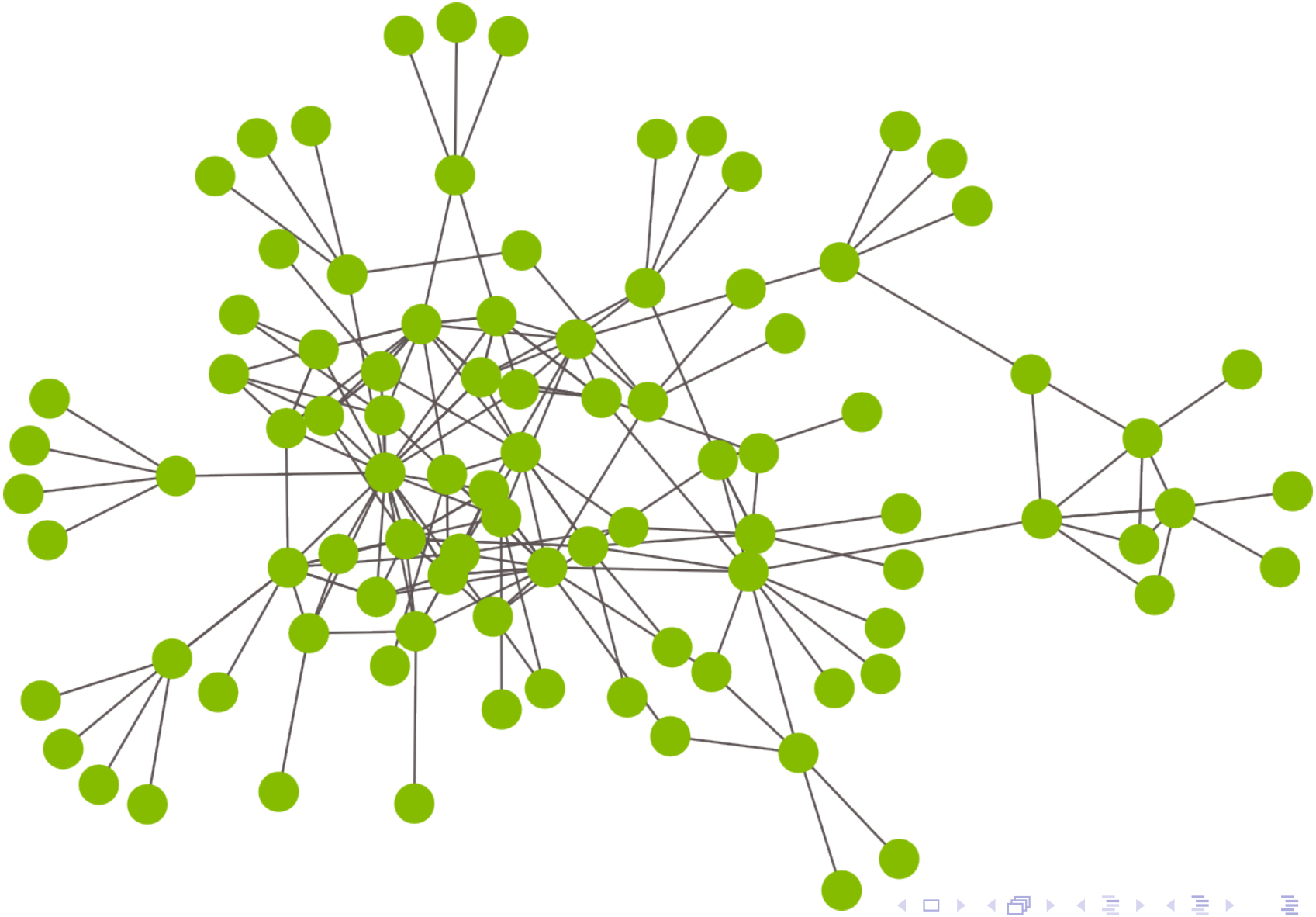
Why are the results so bad?

... Bad

Why are the results so bad?

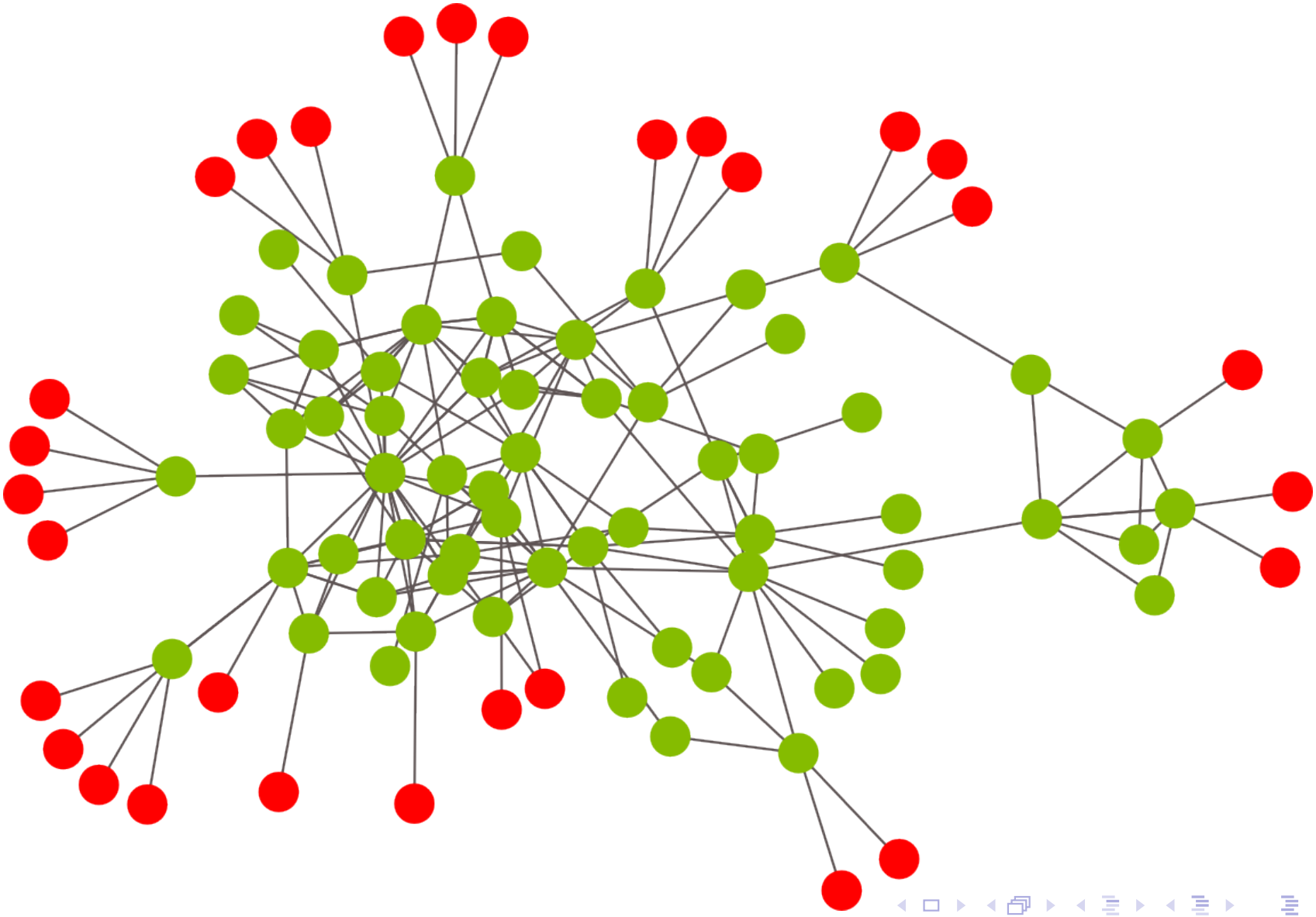
Any ideas?

Why are the results so bad?

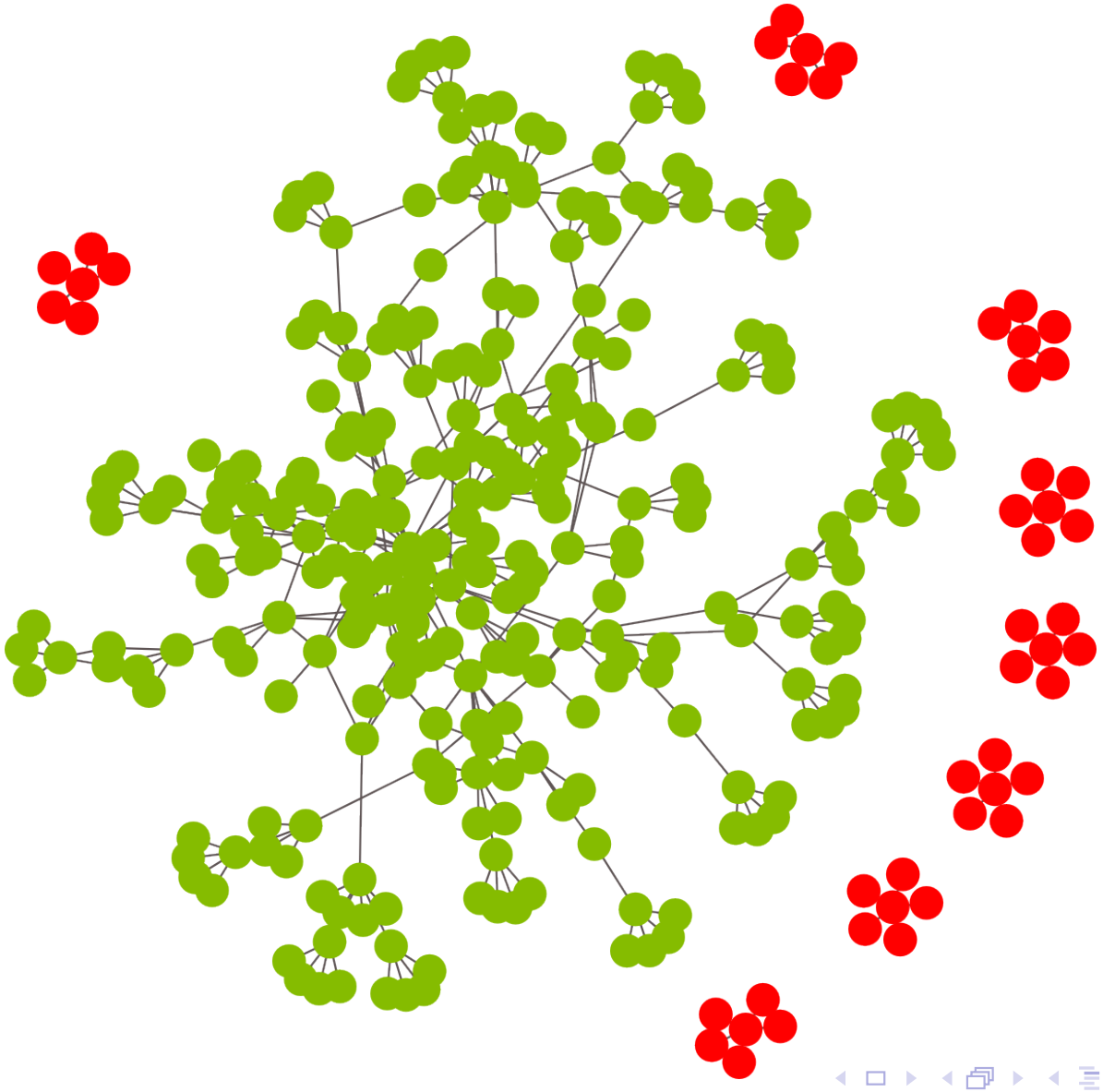


Why are the results so bad?

Why are the results so bad?



Why are the results so bad?

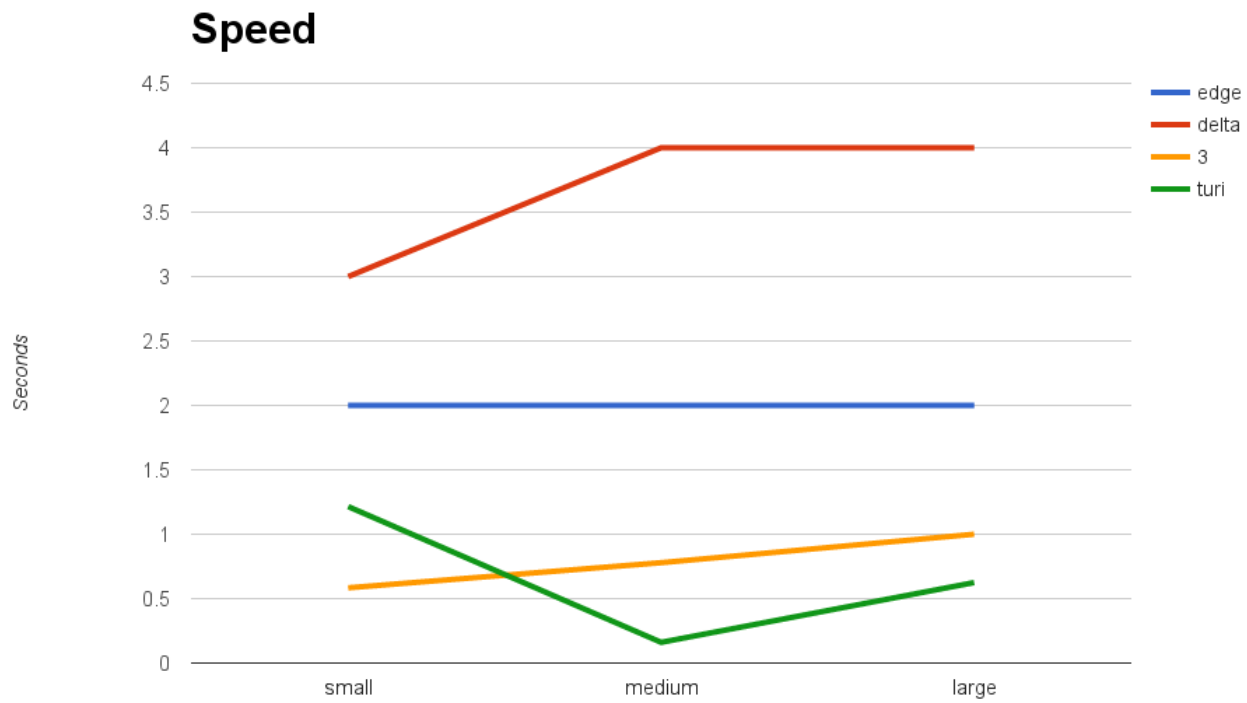


Why are the results so bad?

- Implementations expext an incoming and outging edge for every node
- Dangling nodes
- Spider traps

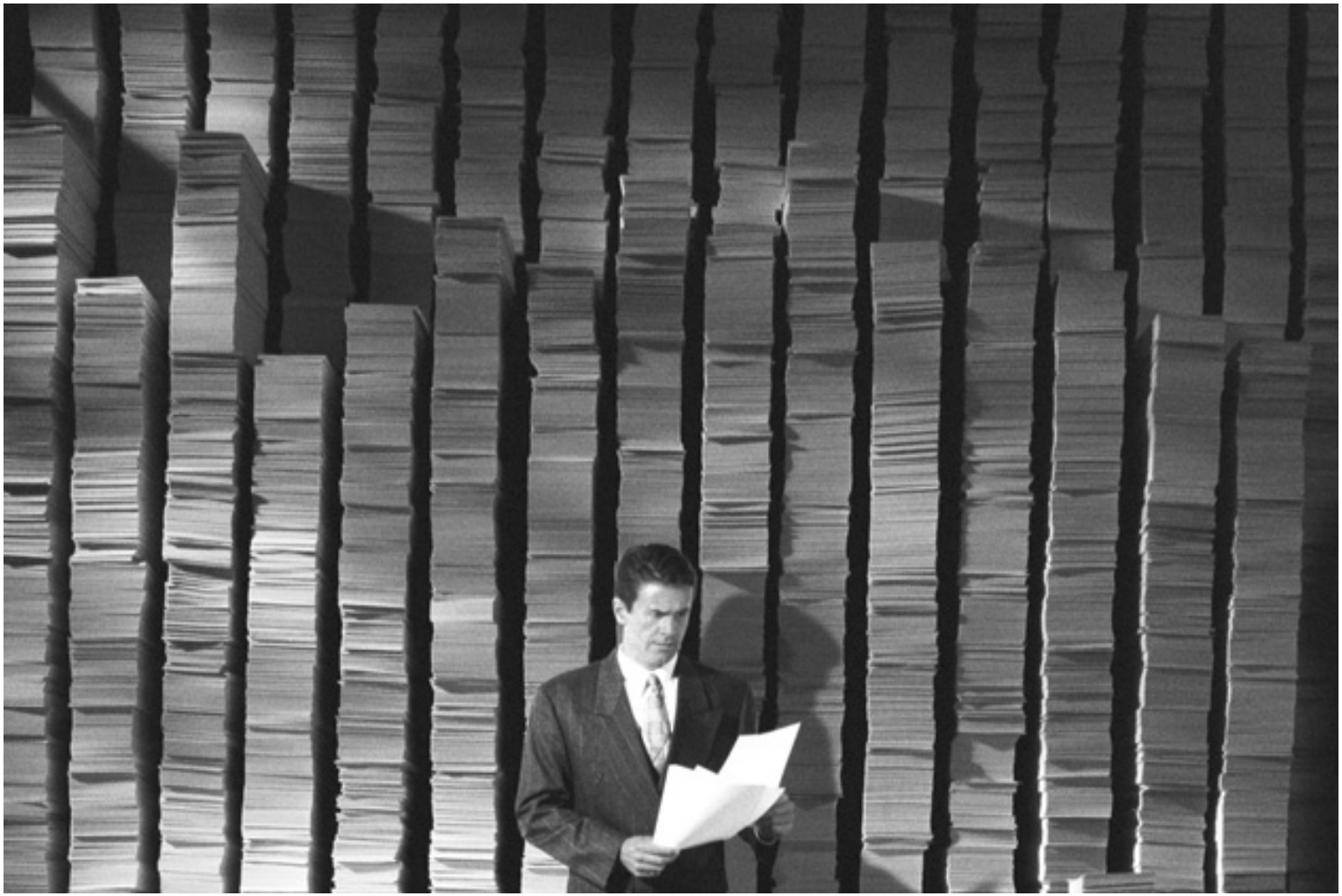
$\rightarrow$ they are all basic implementations
$\rightarrow$ Turi has a advanced implementation

Speed of experiment 1



→ no huge differences with Turi

Experiment 2



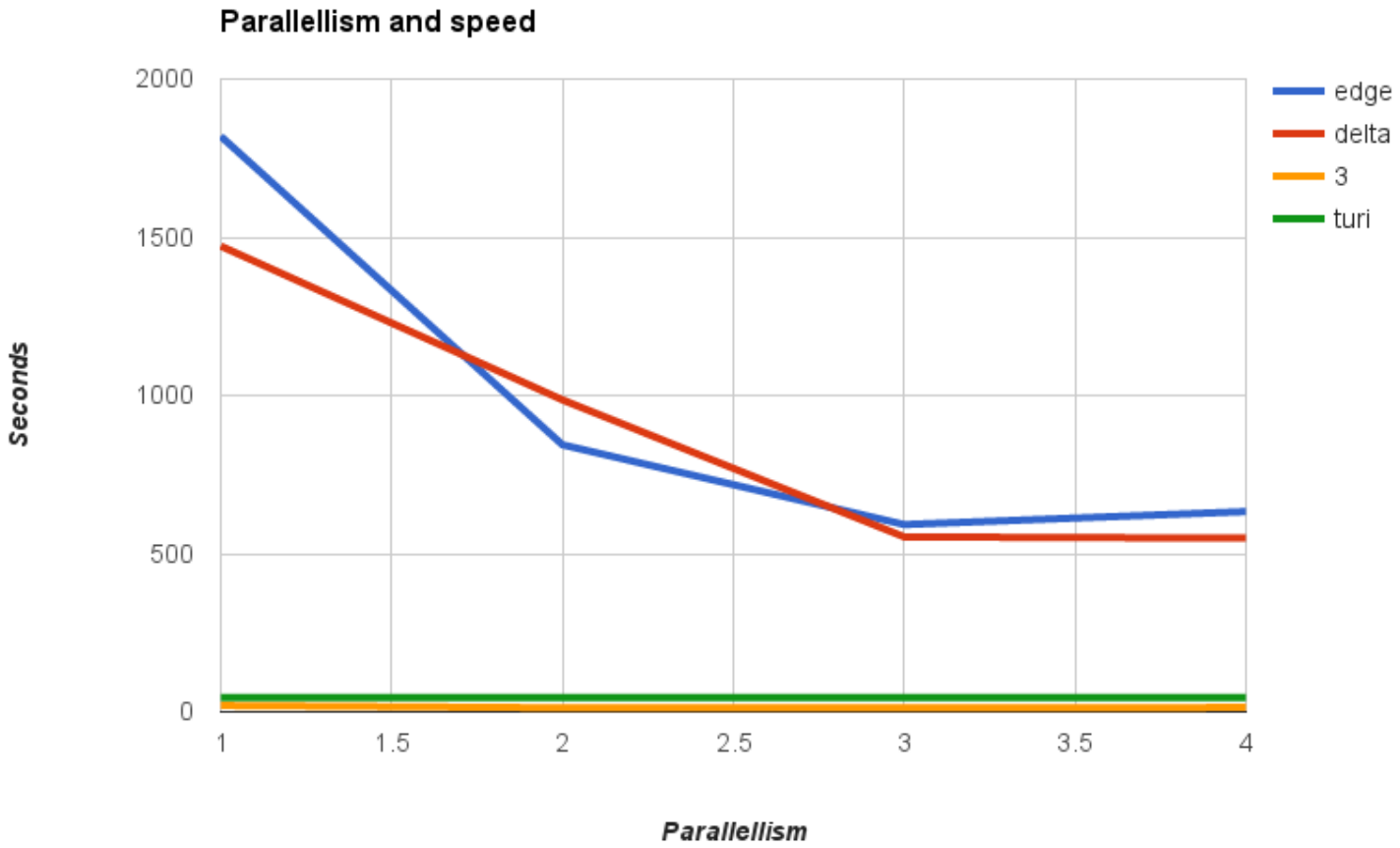
Speed of experiment 2

Algorithm	Edge	Delta	3	Turi
Time (s)	633	549	14	45

- First two algortihms are a lot slower $\backsim$ 10 times.
- Algorithm 3 cheats with the filtering


SLOW

Speed of experiment 2



Agenda

Conclusion

- Use Apache Flink when:
  - ✓ When you have an awfull amount of data
  - ✓ When you can run flink on a cluster

Conclusion

- Use Apache Flink when:
  - ✓ When you have an awfull amount of data
  - ✓ When you can run flink on a cluster
- Do not use Apache Flink when:
  - ✓ When there is ”only“ a lot of data
  - ✓ When you don’t want to lose time setting up Flink
  - ✓ You don’t understand the build-in algorithms well
  - ✓ You want extensive documentation

A `flink` Flink is `flink`

Thank you for your attention

Questions?

The code and data can be found on:
`http://flink.wardschodts.ws`

References I

Data Artisians. *Data Artisians logo*. URL: %5Curl%7Bhttps://www.mapr.com/sites/default/files/data_artisans_logo.png%7D.

Slim Baltagi. *Overview of Apache Flink: Next-Gen Big Data Analytics Framework*. 2015. URL: %5Curl%7Bhttp://www.slideshare.net/sbaltagi/overview-of-apacheflinkbyslimbaltagi?qid=5f0b5424-d187-4c79-a600-6cae794c686e&v=&b=&from_search=3%7D.

Bloomberg. *A lot of data*. URL: %5Curl%7Bhttp://assets.bwbx.io/images/users/iqjWHBFdfxIU/iWDYJ5TG_MhM/v1/-1x-1.jpg%20%7D.

References II


Apache Flink. *Apache Flink Squirrel*. URL: %5Curl%7Bhttps://flink.apache.org/img/logo/png/1000/flink_squirrel_1000.png%7D.


Lawrence Page et al. "The PageRank citation ranking: bringing order to the web." In: (1999).


Beat signer. *Google PageRank*. 2009. URL: %5Curl%7Bhttp://www.slideshare.net/signer/google-pagerank-presentation?qid=18af8836-30e7-41cd-9edb-956bd7ca324d&v=&b=&from_search=2%7D.

References III


Mathias Spahlinger. *There is no repetition*. URL: %5Curl%7Bhttps://www.google.com/search?q=repeat&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi4laH2tuLNAhVnB8AKHTPQCU4Q_AUICCgB&biw=1590&bih=765#tbm=isch&q=no+repetition&imgrc=h1qwLbEEezv8SM:%7D.


Inc Turi. *Turi*. URL: %5Curl%7Bhttps://www.google.com/search?q=repeat&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi4laH2tuLNAhVnB8AKHTPQCU4Q_AUICCgB&biw=1590&bih=765#tbm=isch&q=no+repetition&imgrc=h1qwLbEEezv8SM:%7D.