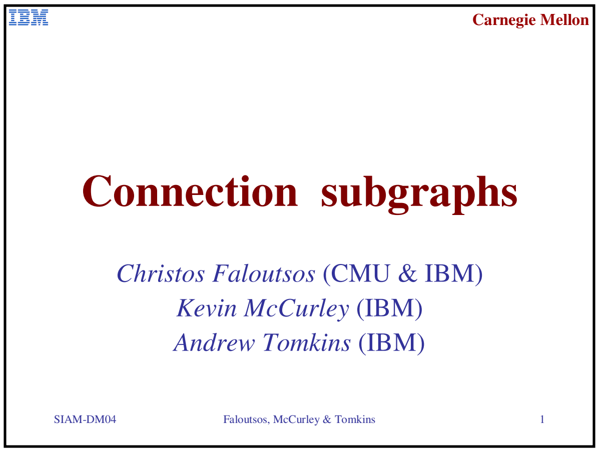
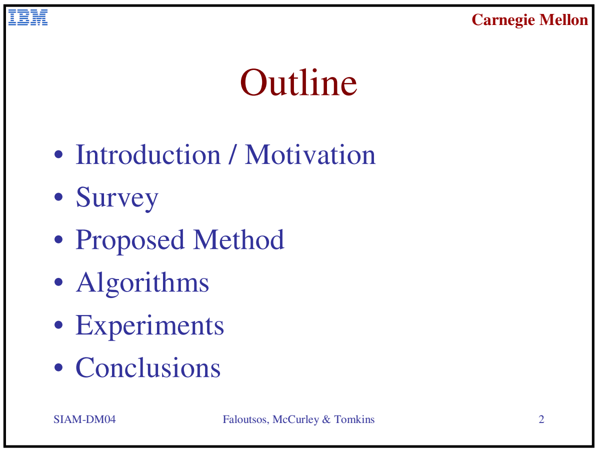


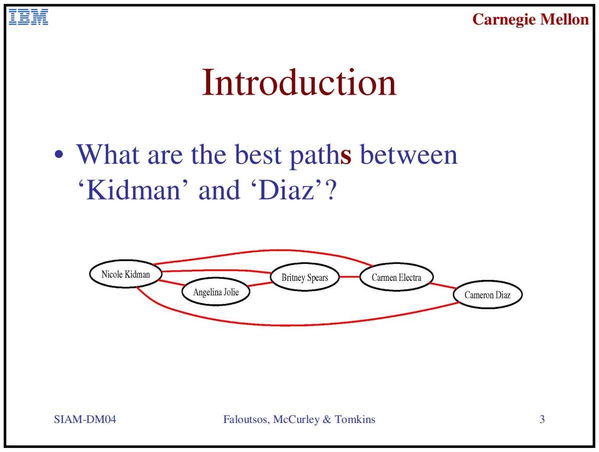
Connection subgraphs

Christos Faloutsos (CMU & IBM)
Kevin McCurley (IBM)
Andrew Tomkins (IBM)



Outline

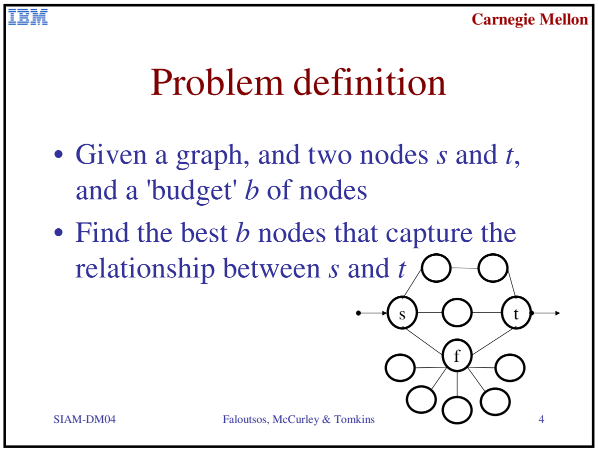
- Introduction / Motivation
- Survey
- Proposed Method
- Algorithms
- Experiments
- Conclusions



Introduction

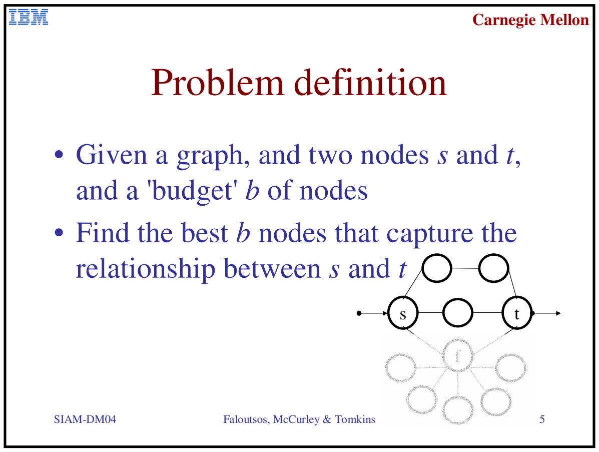
- What are the best path**s** between ‘Kidman’ and ‘Diaz’?

Nicole Kidman
Angelina Jolie
Britney Spears
Carmen Electra
Cameron Diaz



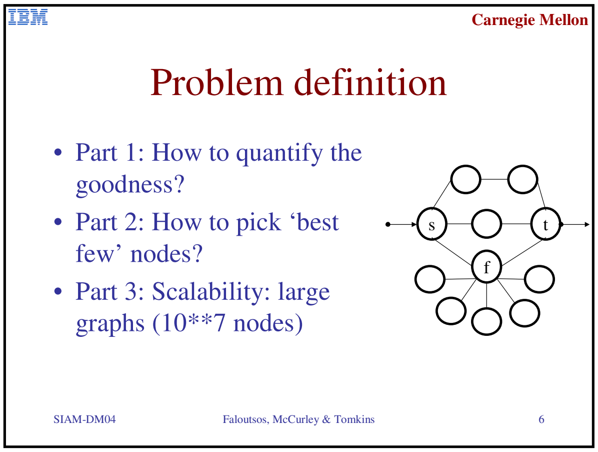
Problem definition

- Given a graph, and two nodes *s* and *t*, and a 'budget' *b* of nodes
- Find the best *b* nodes that capture the relationship between *s* and *t*



Problem definition

- Given a graph, and two nodes *s* and *t*, and a 'budget' *b* of nodes
- Find the best *b* nodes that capture the relationship between *s* and *t*



Problem definition

- Part 1: How to quantify the goodness?
- Part 2: How to pick ‘best few’ nodes?
- Part 3: Scalability: large graphs (10**7 nodes)

Survey

- Graph Partitioning
  - [Karypis+Kumar]; [Newman+];
  - [Virtanen]; …
- Communities
  - [Flake+]; [Tomkins, Kleinberg+]
- External distances [Palmer+]

Outline

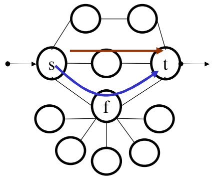
- Introduction / Motivation
- Survey
- ➡ Proposed Method
- Algorithms
- Experiments
- Conclusions

Proposed method

- part 1: measuring goodness:
  - electricity
- part 2: finding good paths
  - dynamic programming
- part 3: scalability
  - heuristics

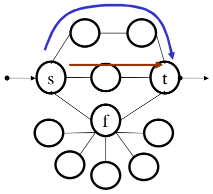
Electricity

- Why not shortest path?



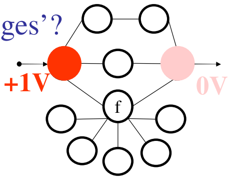
Electricity

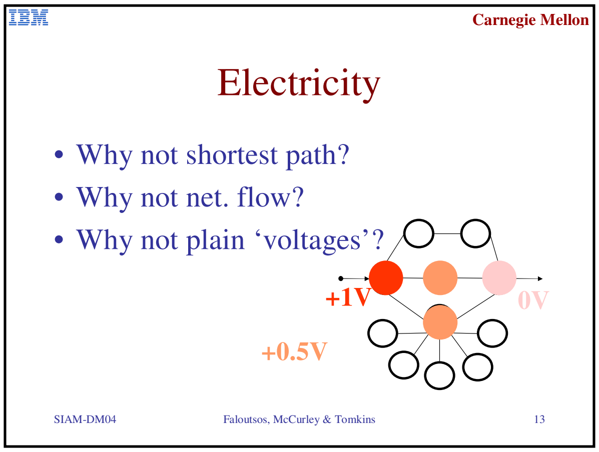
- Why not shortest path?
- Why not net. flow?



Electricity

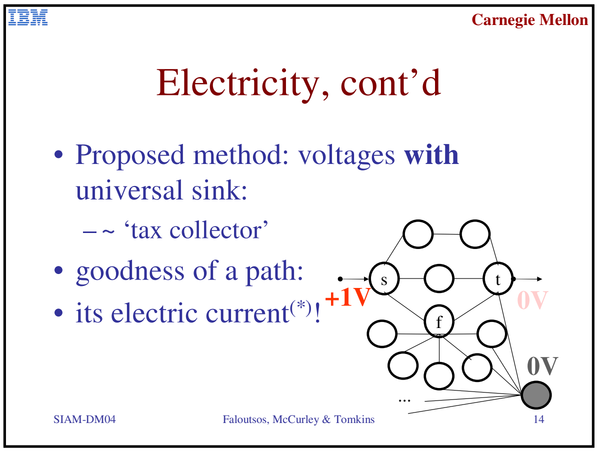
- Why not shortest path?
- Why not net. flow?
- Why not plain ‘voltages’?





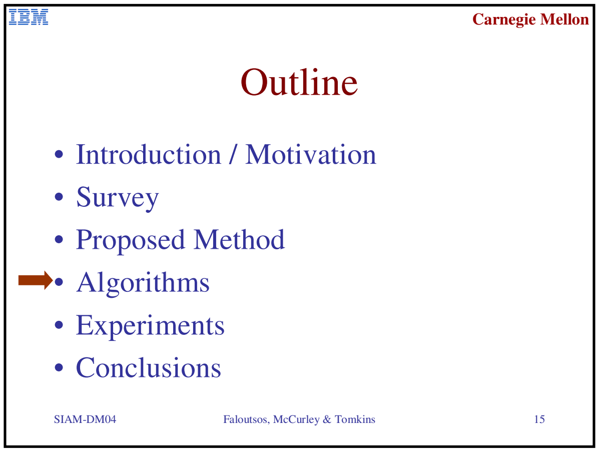
Electricity

- Why not shortest path?
- Why not net. flow?
- Why not plain ‘voltages’?



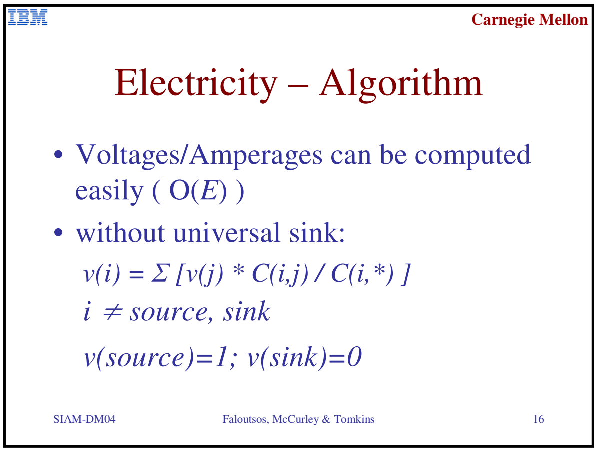
Electricity, cont’d

- Proposed method: voltages **with** universal sink:
  - – ~ ‘tax collector’
- goodness of a path:
- its electric current(*)!



Outline

- Introduction / Motivation
- Survey
- Proposed Method
- ➡ Algorithms
- Experiments
- Conclusions



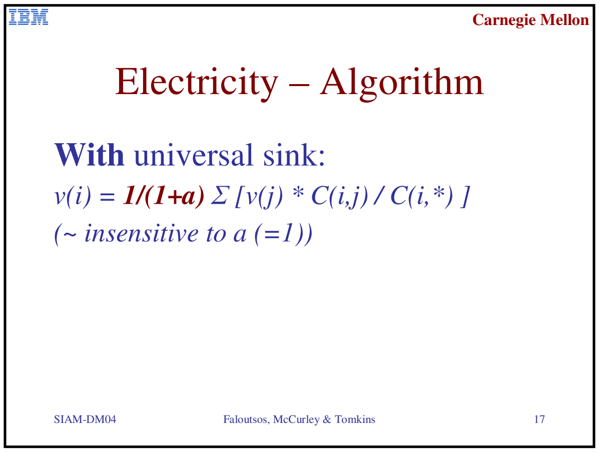
Electricity – Algorithm

- Voltages/Amperages can be computed easily (O(*E*))
- without universal sink:

$v(i) = \Sigma\, [v(j) * C(i,j) / C(i,*)\,]$

$i \neq source, sink$

$v(source)=1;\ v(sink)=0$

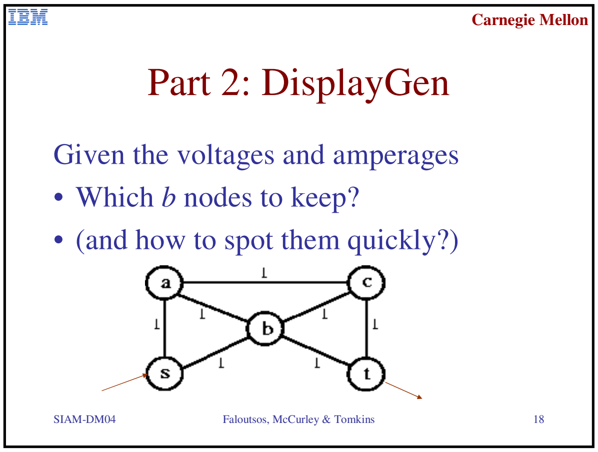


Electricity – Algorithm

With universal sink:

$v(i) = \mathbf{1/(1+a)}\ \Sigma\, [v(j) * C(i,j) / C(i,*)\,]$

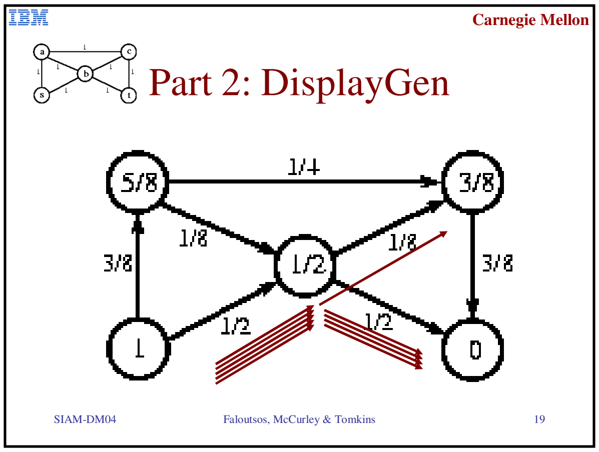
(~ *insensitive to a (=1)*)



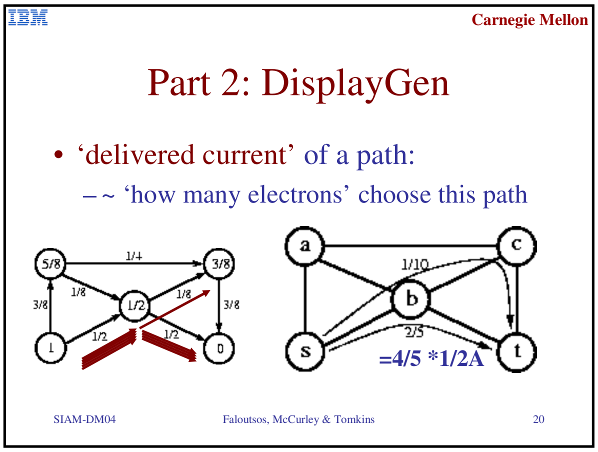
Part 2: DisplayGen

Given the voltages and amperages

- Which *b* nodes to keep?
- (and how to spot them quickly?)

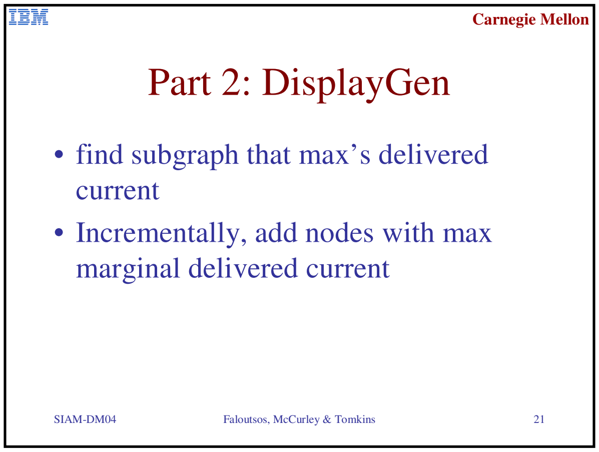


Part 2: DisplayGen



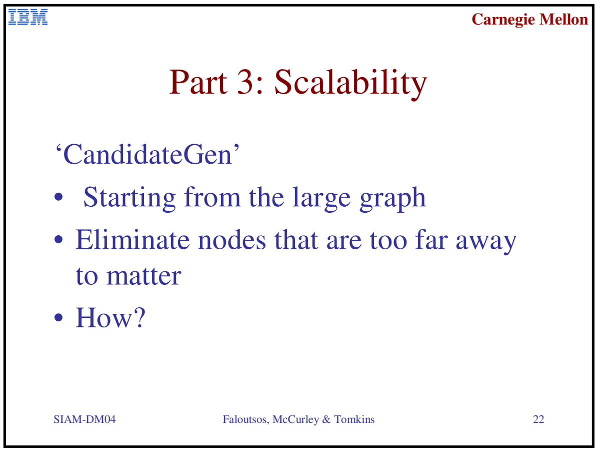
Part 2: DisplayGen

- ‘delivered current’ of a path:
  - ~ ‘how many electrons’ choose this path



Part 2: DisplayGen

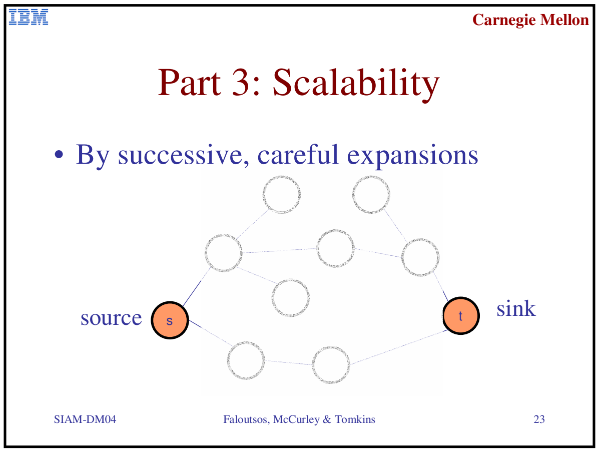
- find subgraph that max’s delivered current
- Incrementally, add nodes with max marginal delivered current



Part 3: Scalability

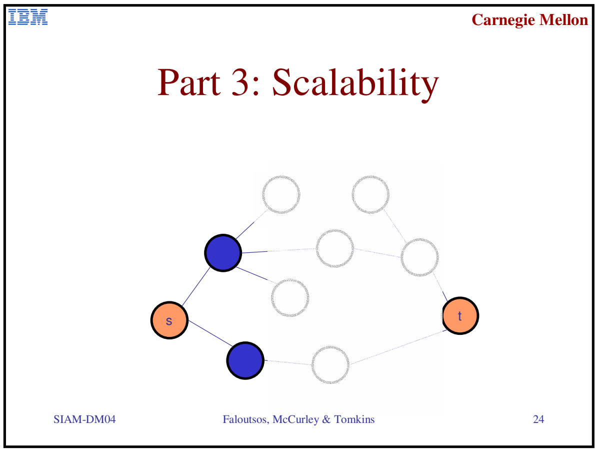
‘CandidateGen’

- Starting from the large graph
- Eliminate nodes that are too far away to matter
- How?



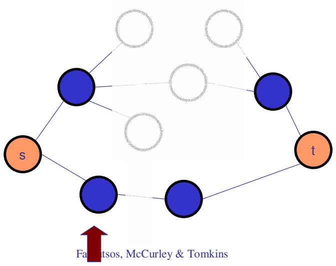
Part 3: Scalability

- By successive, careful expansions

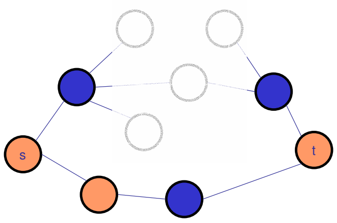


Part 3: Scalability

Part 3: Scalability



Part 3: Scalability



Pseudo-code

```
Until (stoppingCriterion)
  use pickHeuristic() to pick a node n
   expand node n
```

Pseudo-code

pickHeuristic() favors

- Nearby nodes with
- Strong connections to source or sink and with
- Small degree

Outline

- Introduction / Motivation
- Survey
- Proposed Method
- Algorithms
- ➡ Experiments
- Conclusions

Experiments

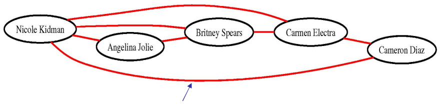
- on large real graph
  - ~15M nodes, ~100M edges, weighted
  - ‘who co-appears with whom’ (from 500M web pages)
- Q1: Quality of ‘voltage’ approach?
- Q2: Speed/accuracy trade-off?

Q1: Quality

- Actors (A); Computer-Scientists (CS)
- Kidman-Diaz (A-A)
- Negreponte-Palmisano (CS-CS)
- Turing-Stone (CS-A)

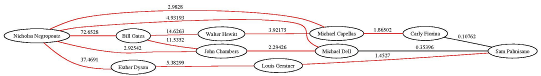
(A-A) Kidman-Diaz

- What are the best paths between 'Kidman' and 'Diaz'?



Strong, direct link

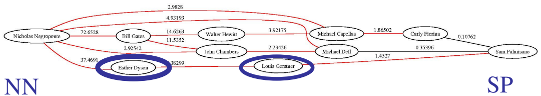
CS-CS: Negreponte - Palmisano



NN SP

- Mainly: CEOs of major Computer companies (Dell, Gates, Fiorina, ++)

CS-CS: Negreponte - Palmisano

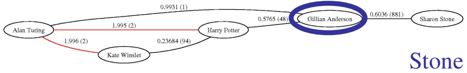


NN SP

Esther Dyson Louis Gerstner

CS-A: Turing - Stone

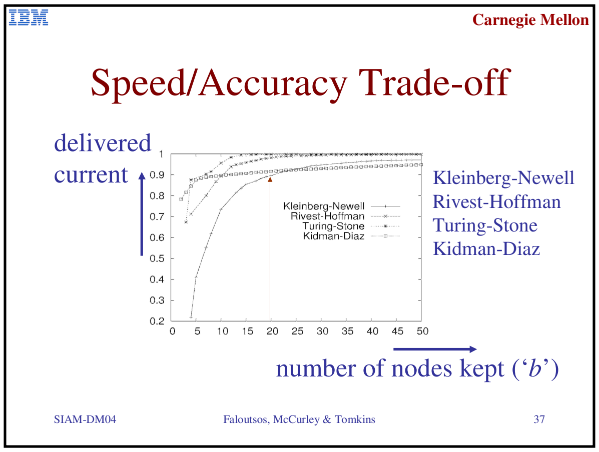
Turing Anderson



Stone

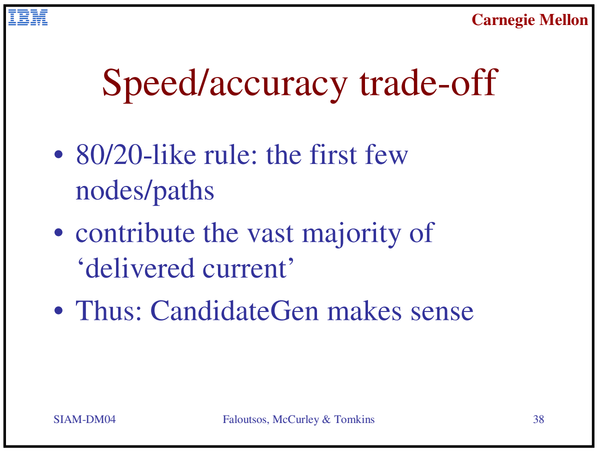
Outline

- Introduction / Motivation
- ...
- Experiments
  - Q1: quality
  - ➡ Q2: speed/accuracy trade-off
- Conclusions



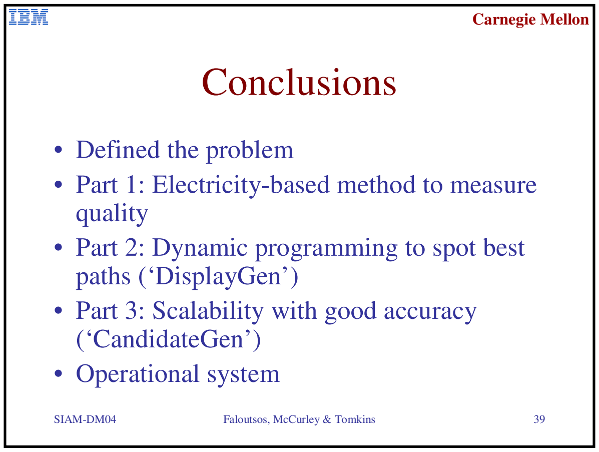
Speed/Accuracy Trade-off

Kleinberg-Newell
Rivest-Hoffman
Turing-Stone
Kidman-Diaz



Speed/accuracy trade-off

- 80/20-like rule: the first few nodes/paths
- contribute the vast majority of ‘delivered current’
- Thus: CandidateGen makes sense



Conclusions

- Defined the problem
- Part 1: Electricity-based method to measure quality
- Part 2: Dynamic programming to spot best paths (‘DisplayGen’)
- Part 3: Scalability with good accuracy (‘CandidateGen’)
- Operational system