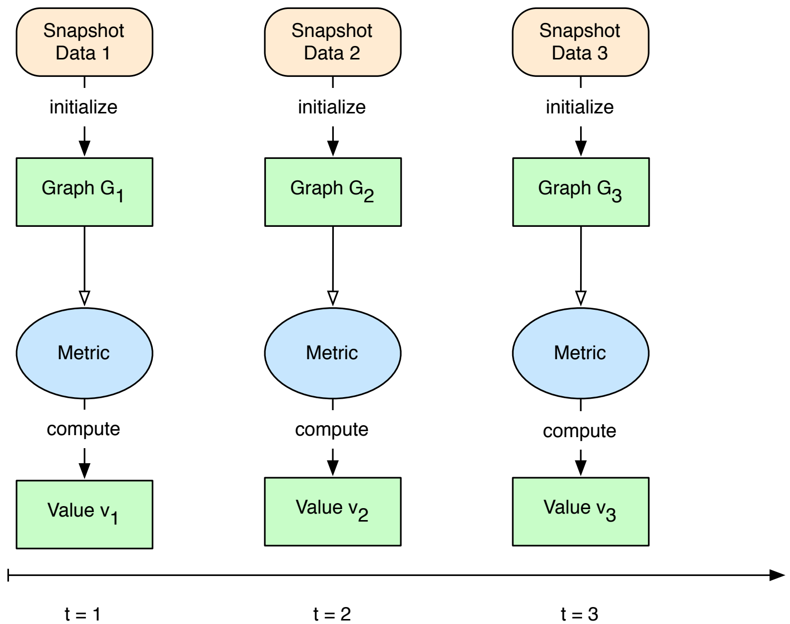
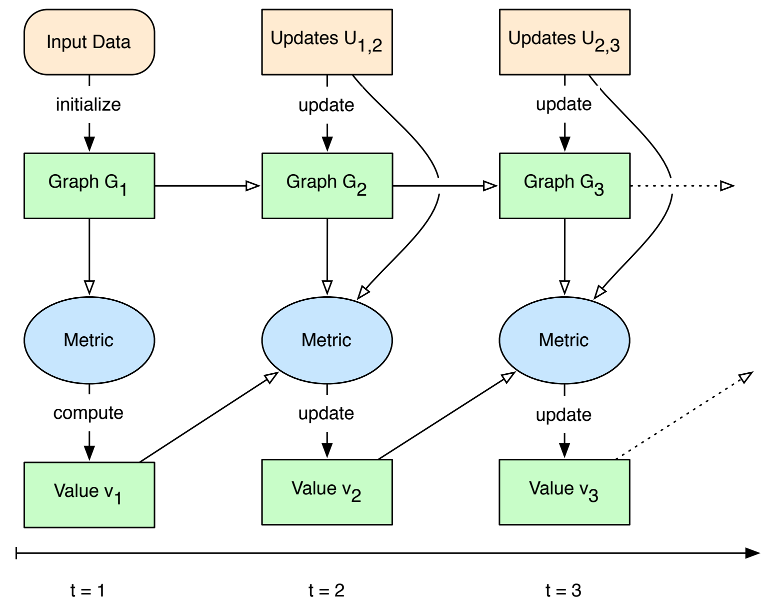
Dynamic Graph Algorithms
Creating a Classification

Benjamin Schiller, P2P Networks, TU Darmstadt

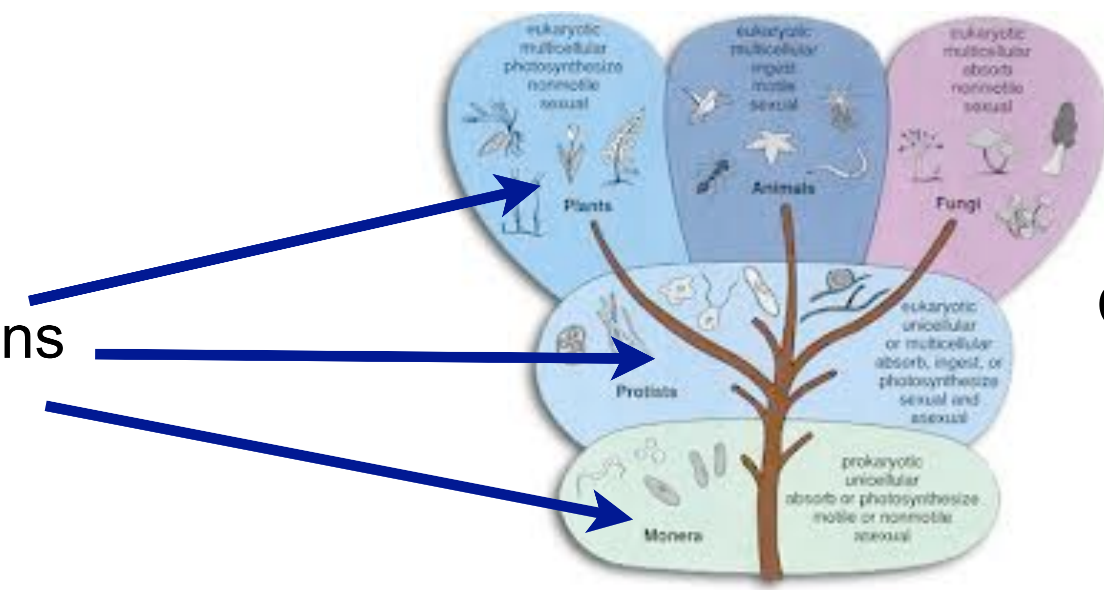
P2P Research Meeting / 07.08.2013

Motivation





Dimensions



Classification

Example

- Input
  - Static graph snapshots
  - Stream of data / changes to a system
  - Dynamic graph, 100 changes per second

- Graph Algorithm
  - Shortest Path Distribution
  - Shortest Path Trees
  - Shortest Path Query Structure

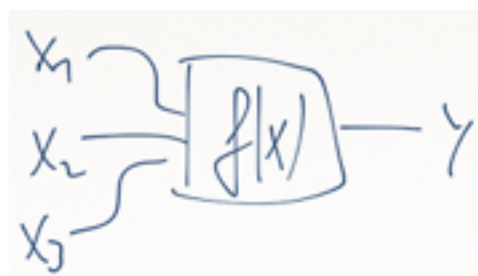
- Graph data structures
  - Arrays, HashSets, HashMaps
  - Bloom filters
  - Adjacency matrix

Algorithms on Data Streams

- Hinzinger et al. - “Computing on Data Streams”

1. Passes
   - One-pass
   - Multi-pass


2. Computation
   - Deterministic
   - Randomized



3. Exactness
   - Exact
   - Approximation

Classification Dimensions

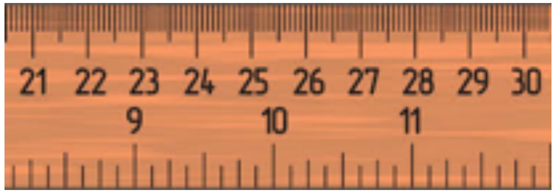
1. Workflow

2. Problem division

3. Data structures

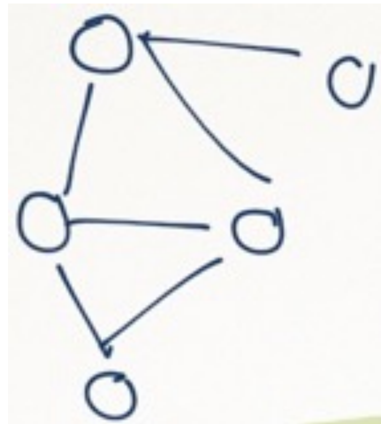
4. Results

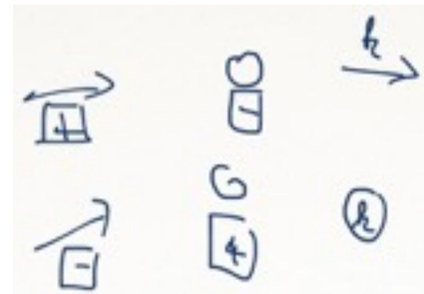
5. Exactness

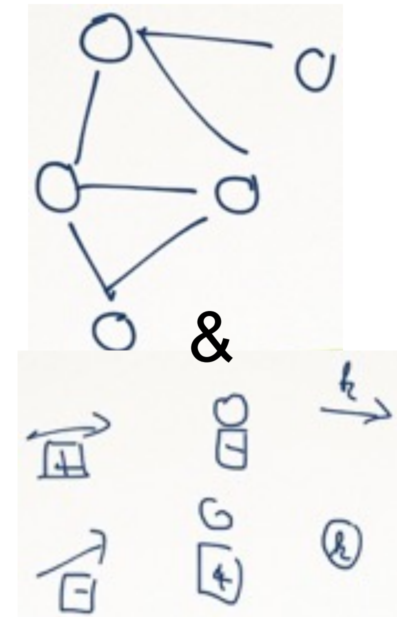


1. Workflow

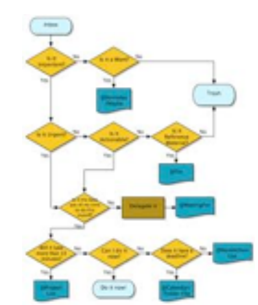
- Input
  - Graph
  - Updates
  - Graph & updates







- Processing
  - Single-pass
  - Multi-pass
  - Random access

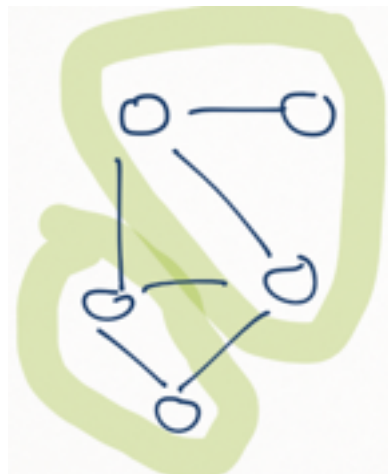


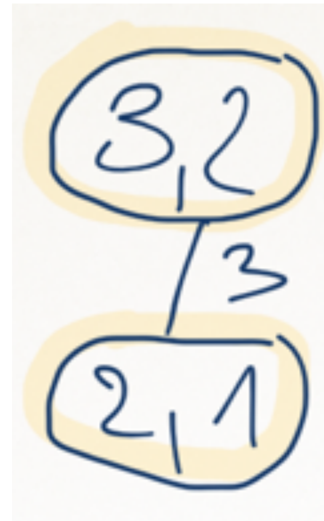
1. Workflow

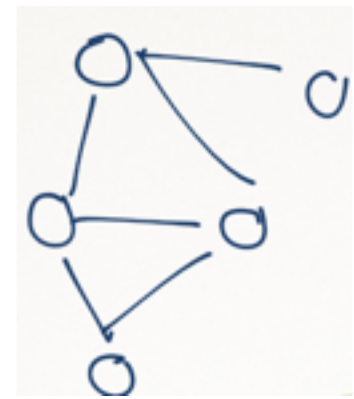
	Input			Processing			
	G	U	G&U	S-P	M-P	RA	
1	X			X			static
2	X				X		
3	X					X	
4		X		X			stream
5		X			X		
6		X				X	
7			X	X			single updates
8			X		X		batch updates
9			X			X	

2. Problem Division

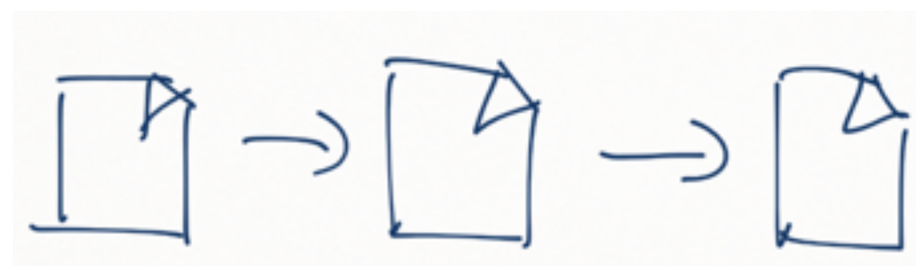
1. Spatial
   - Partitioning
   - Coarsening
   - None







2. Computational
   - Parallel
   - Iterative



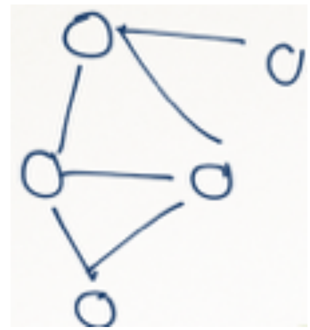
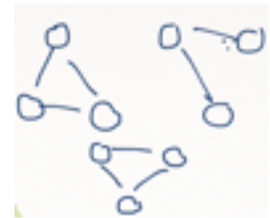
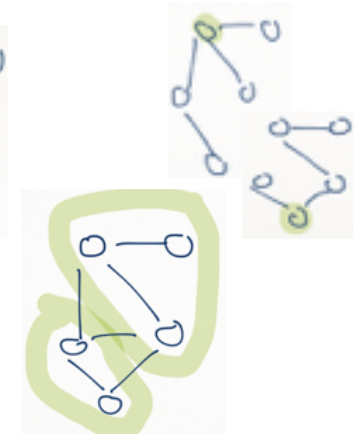
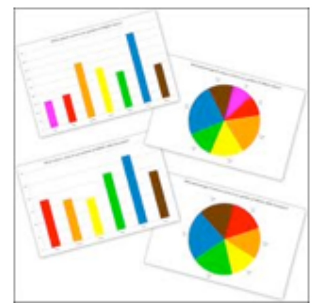
2. Problem Division

	Spat.			Comp.				
	Part	Coar	-	Para	Iter	*mem*	*time*	*compl*
1	X			X		-	-	
2	X				X	-		
3		X		X		-	+/-	-
4		X			X	-	+	-
5			X	X				
6			X		X			

3. Data Structures

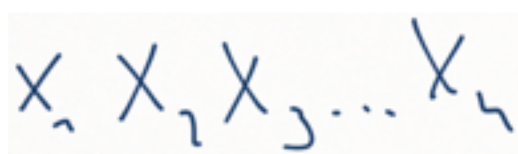
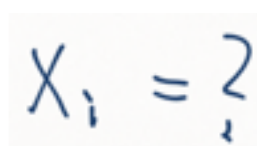
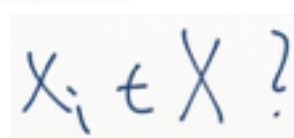
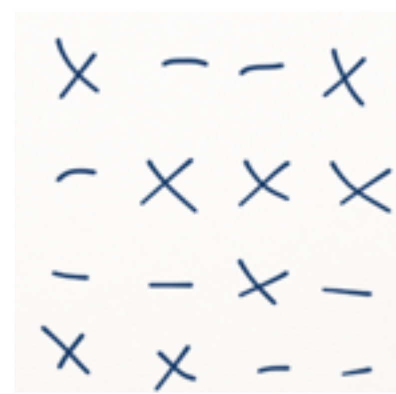
- Type

1. Internal Model
   - Graph representation
   - Auxiliary data (subgraph, tree, ...)

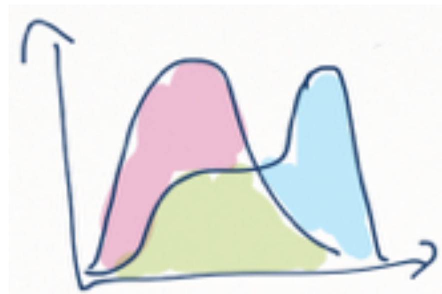
2. Results

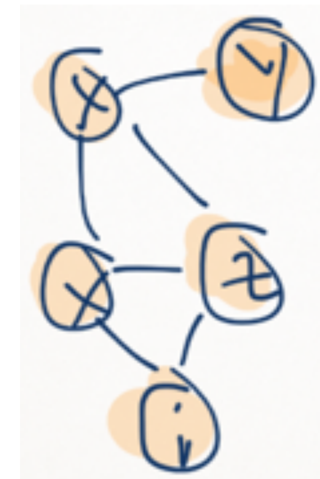
- Access type
  - Iterators
  - Random access
  - Contains
  - Adjacency matrix

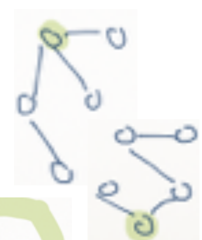
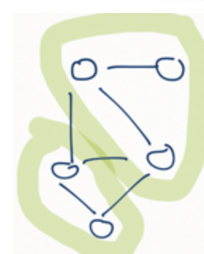
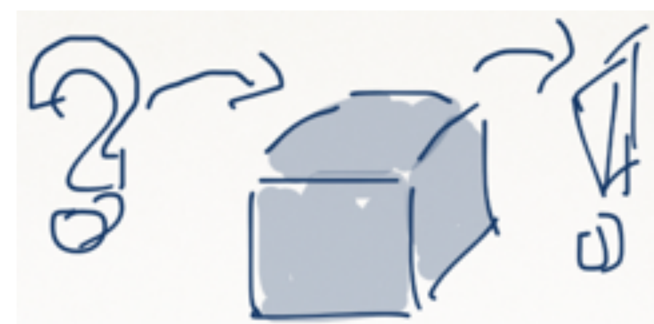
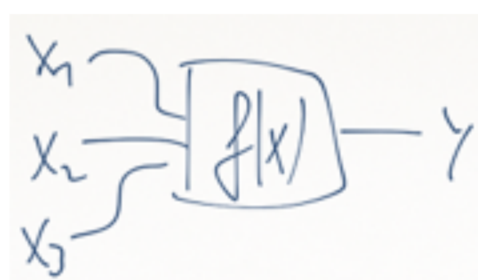






4. Results

- Type
  - Metric
    - Distribution
    - Value
    - Node Value List
    - Value List
  - Graph structure
  - Query structure
- Computation
  - Deterministic
  - Randomized

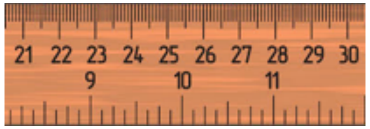




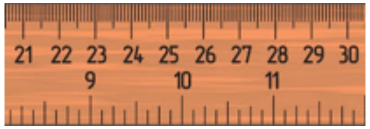
5. Exactness

- Results
  - Exact — X
  - Within bounds — $X \pm \varepsilon$
  - Approximated — $\approx X$
- Data structures
  - Edge / node existence
    - Correct — $x \in \{x\},\ x \notin \{y\}$
    - False-positive — $x \in \{a, b\}$
    - False-negative — $x \notin \{x, y\}$
  - Degree / weights
    - Correct — X
    - Within bounds — $X \pm \varepsilon$
    - Incorrect — $\approx X$



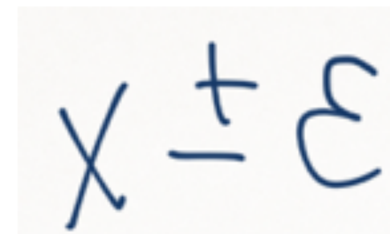
5. Exactness

	Results			Data Structures					
	=	+/- ε	~	=	f-pos	f-neg	=	+/- ε	~
1	X			X			X		
2		X		X			X	X	
3			X	X	X	X	X	X	X



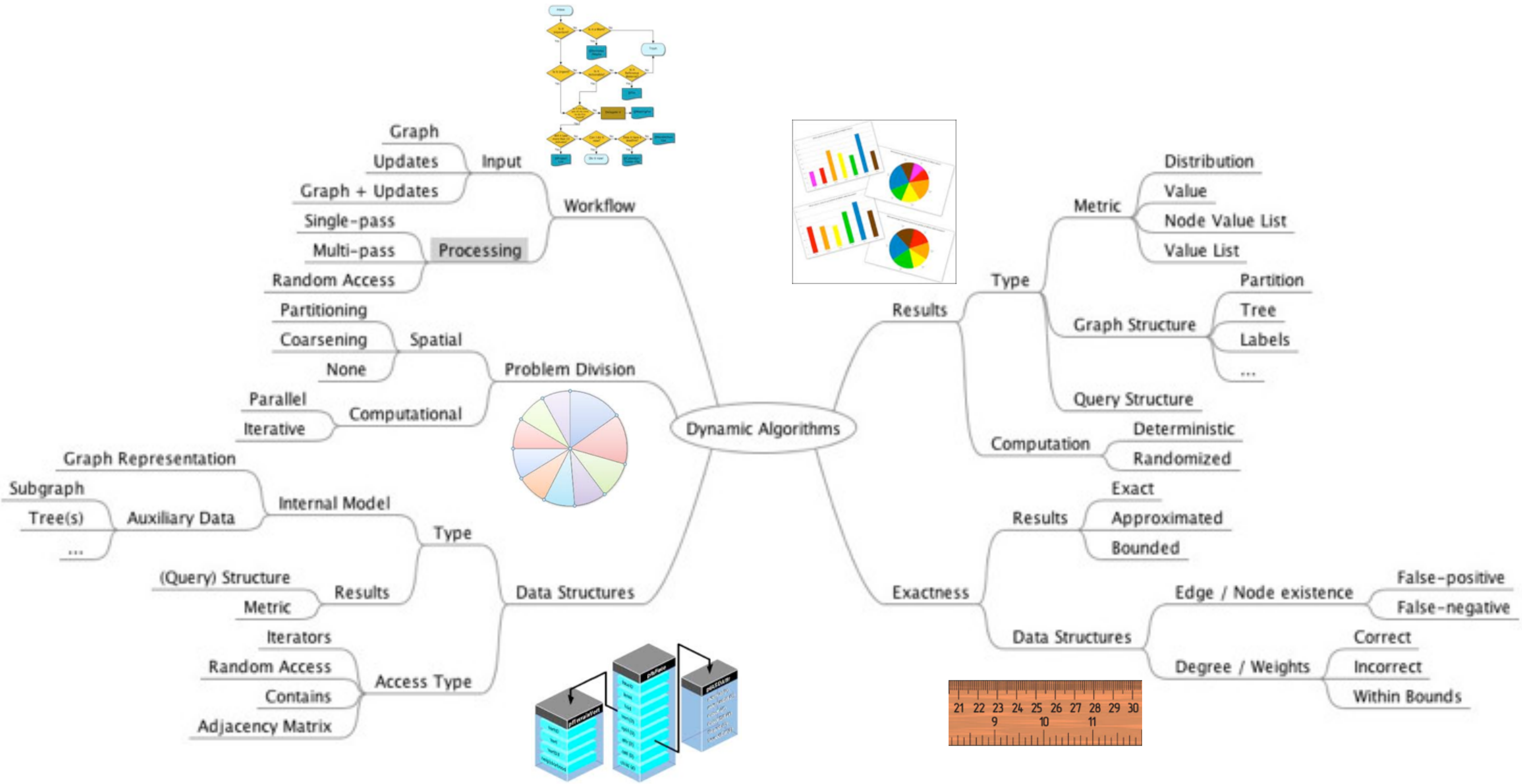
Properties of Dynamic Algorithms

- Exactness
  - Value: % of diff to exact solution
  - Node values: % of diff to exact solution (plotting?)
  - Distribution: ???
  - (=> René)



- Time / space complexity w.r.t.
  - Input size, input type
  - Data structures
  - (=> Nico)

Summary


Dynamic Algorithms
Workflow
Input
Graph
Updates
Graph + Updates
Processing
Single-pass
Multi-pass
Random Access
Problem Division
Spatial
Partitioning
Coarsening
None
Computational
Parallel
Iterative
Data Structures
Type
Internal Model
Graph Representation
Auxiliary Data
Subgraph
Tree(s)
...
Results
(Query) Structure
Metric
Access Type
Iterators
Random Access
Contains
Adjacency Matrix
Results
Type
Metric
Distribution
Value
Node Value List
Value List
Graph Structure
Partition
Tree
Labels
...
Query Structure
Computation
Deterministic
Randomized
Exactness
Results
Exact
Approximated
Bounded
Data Structures
Edge / Node existence
False-positive
False-negative
Degree / Weights
Correct
Incorrect
Within Bounds
21 22 23 24 25 26 27 28 29 30
9 10 11

Open Problems / Next Steps

- Problems reg. classification
  - General terminology: subgraph(s) vs. partition(s)
  - Exactness terms: results vs. degree / weights

- Next steps
  - Incorporate your suggestions / changes / clarifications
  - Determine and compare properties for many algorithms
    - Large table, potentially identify classes