Detecting quasi-cliques in massive sparse multi digraphs

Maurício G. C. Resende

Algorithms & Optimization Research Dept.

AT&T Labs Research

Florham Park, New Jersey

mgcr@research.att.com

http://www.research.att.com/~mgcr

Joint work with James Abello, Panos Pardalos, & Sandra Sudarsky

August 2000

Summary of talk

- Data explosion
- Massive graphs arising from telephone call detail database
- Structure of call detail graph
- Searching for large cliques and bicliques
- Some experimental results

Data explosion

(Abello, Pardalos, & R., Eds., "Handbook of Massive Data Sets," Kluwer, 2001)

- Proliferation of massive data sets brings with it computational challenges
- Data avalanche arises in a wide range of scientific and commercial applications
- Today's data sets are of high dimension and are made up of huge numbers of observations:
  - More often they overwhelm rather than enlighten
- Outstripped the capabilities of traditional data measurement, data analysis, and data visualization tools

Data explosion

- A variety of massive data sets can be modeled as a very large multi-digraph
  - Special set of edge attributes represent special characteristics of application
- WWW: nodes are pages, edges are links pointing from one page to another
- Telephone call graph is another example ...

Call detail

- Every phone call placed on AT&T network generates a record ($\sim$ 200 bytes) with:
  - Originating & terminating numbers
  - Start time & duration of call
  - Other billing information
- The collection of these records is known as the Call Detail Database

Call detail

- AT&T system (currently) generates:
  - 250 million records per day (on average)
  - 320 million records on busy day
  - 18 terabytes of data per year
- Data is accessed for:
  - Billing & customer inquiries
  - Marketing & traffic engineering

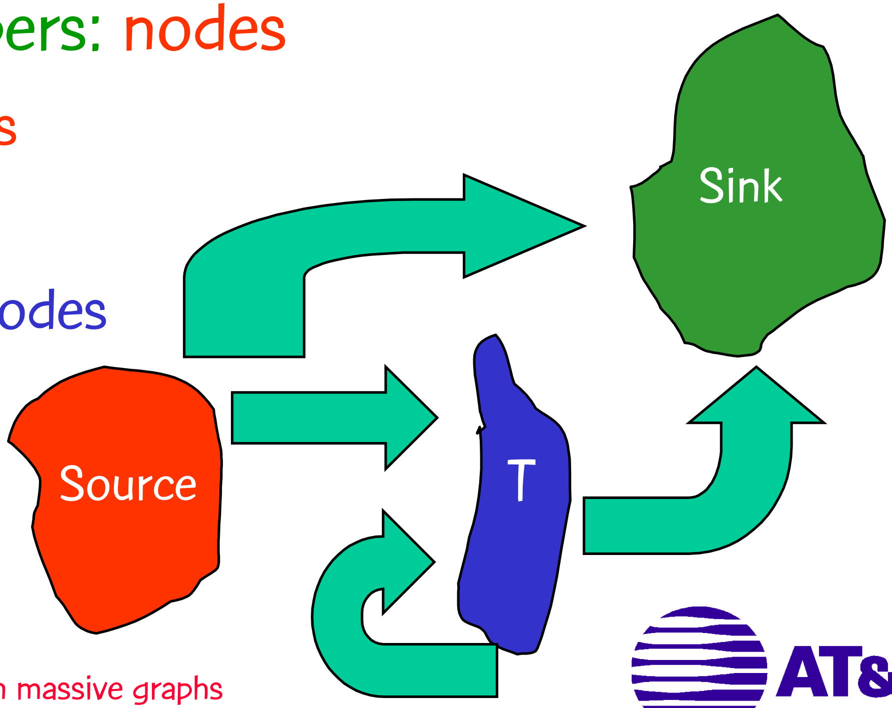
Call detail graph

- $G = (V, E)$ is a directed graph:
  - V is the set of phone numbers
  - E is the set of phone calls
    - $(u,v) \in E$ implies that phone u called phone v
- G quickly grows into a huge graph
  - Hundreds of millions of nodes and billions of edges
  - Our goal is to work with one year of data ($\sim$ 1 Tb)

AT&T

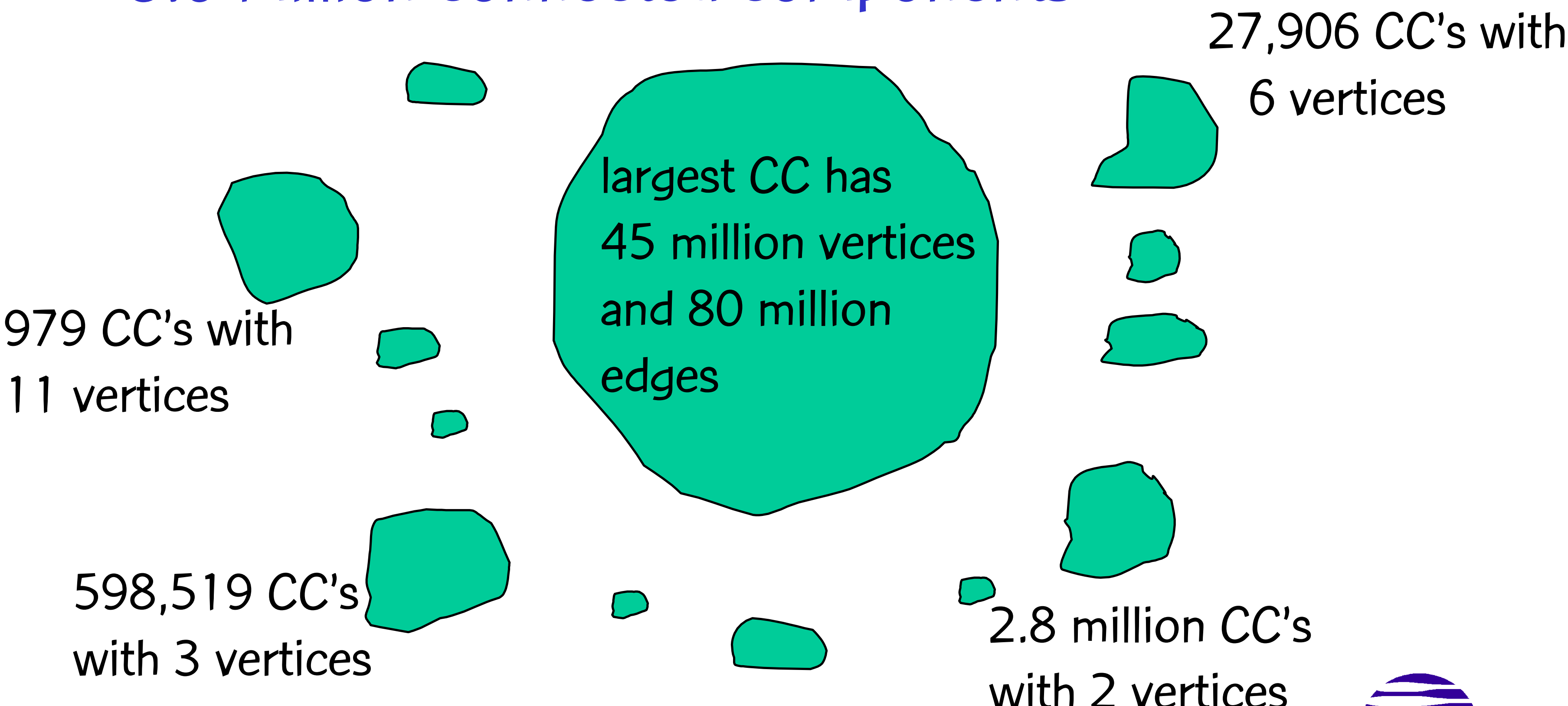
Structure of call detail graph

- Consider a 12-hour call detail graph
  - 123 million records: edges
  - 53 million phone numbers: nodes
    - 21 million source nodes
    - 22 million sink nodes
    - 10 million transmittal nodes

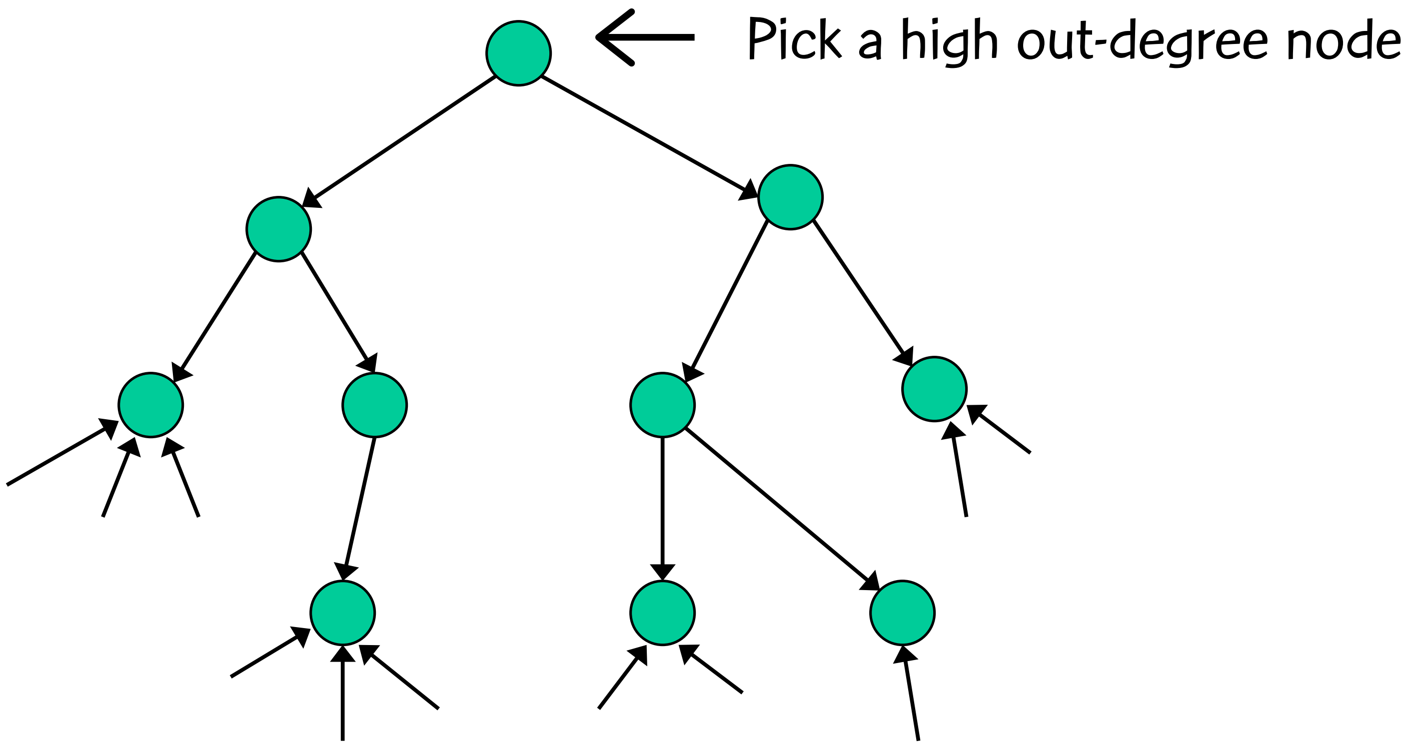


Connected components

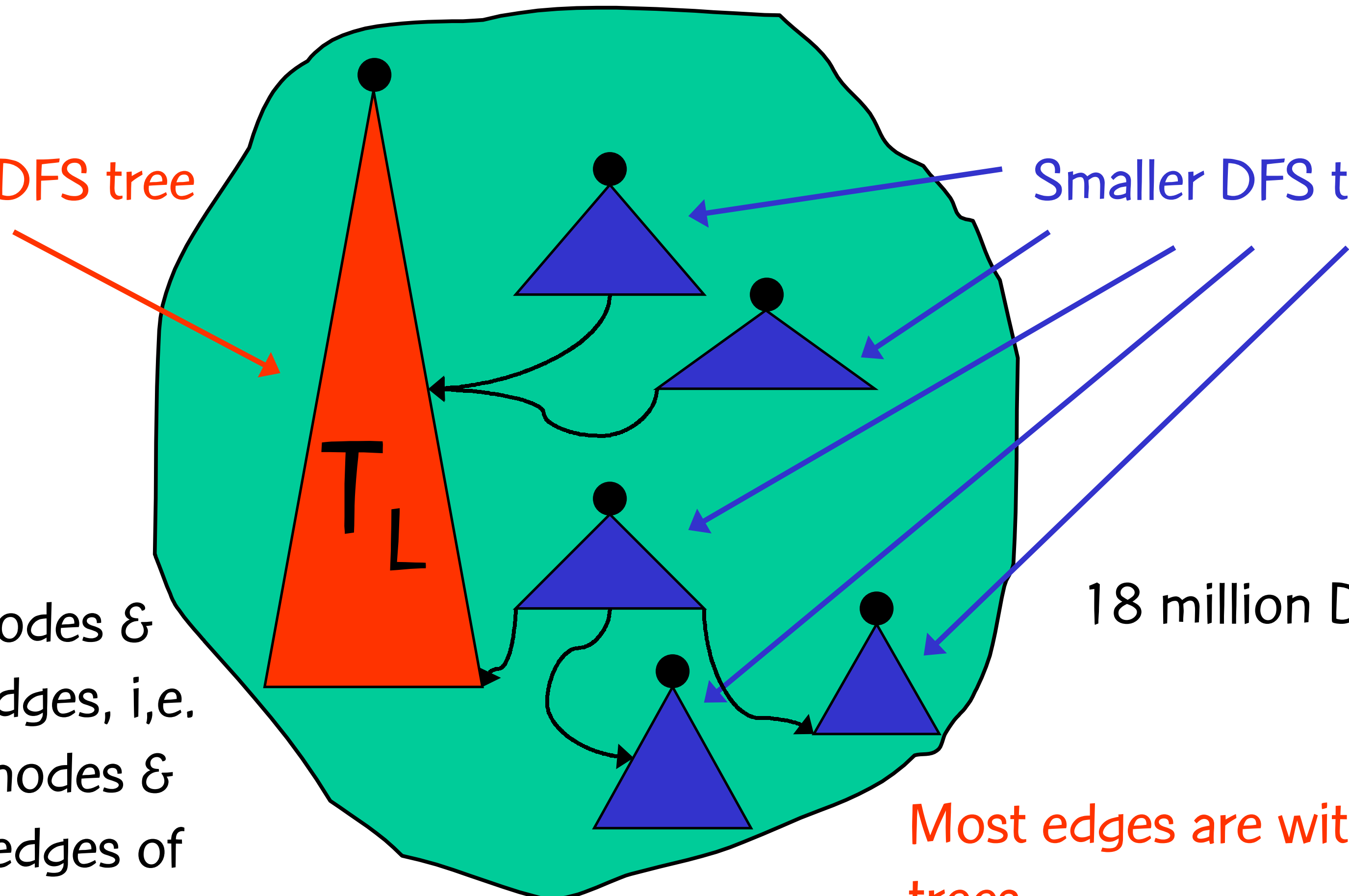
3.6 million connected components



Depth first search (DFS) tree



DFS trees in largest CC

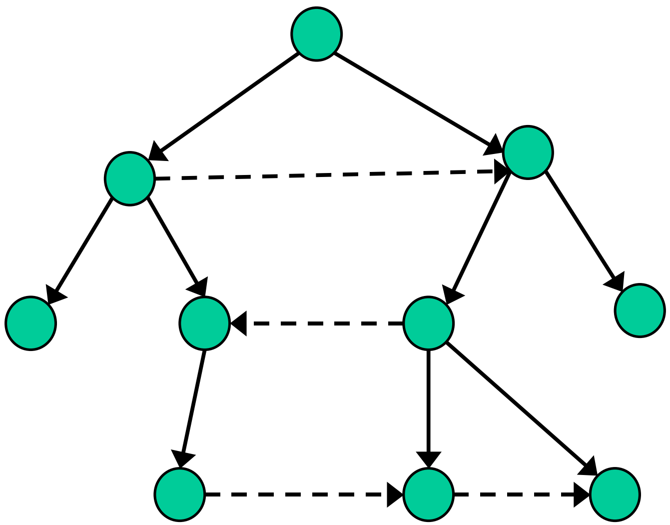


$G(T_L)$ has
10 million nodes &
19 million edges, i,e.
22% of the nodes &
24% of the edges of
the CC

18 million DFS trees

Most edges are within trees.

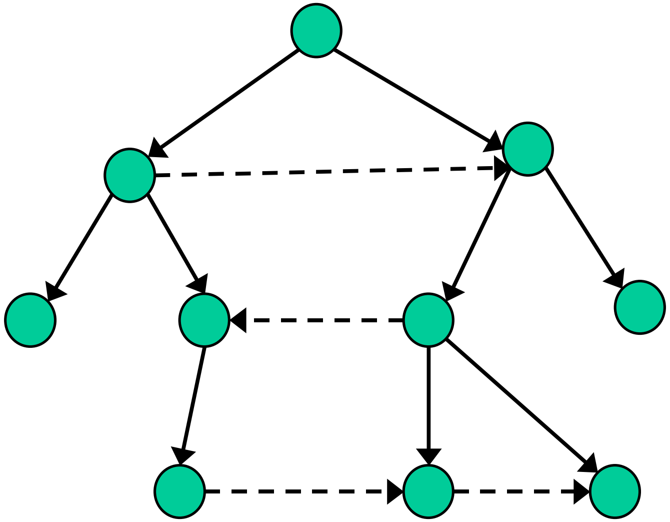
Subgraph induced by DFS tree nodes



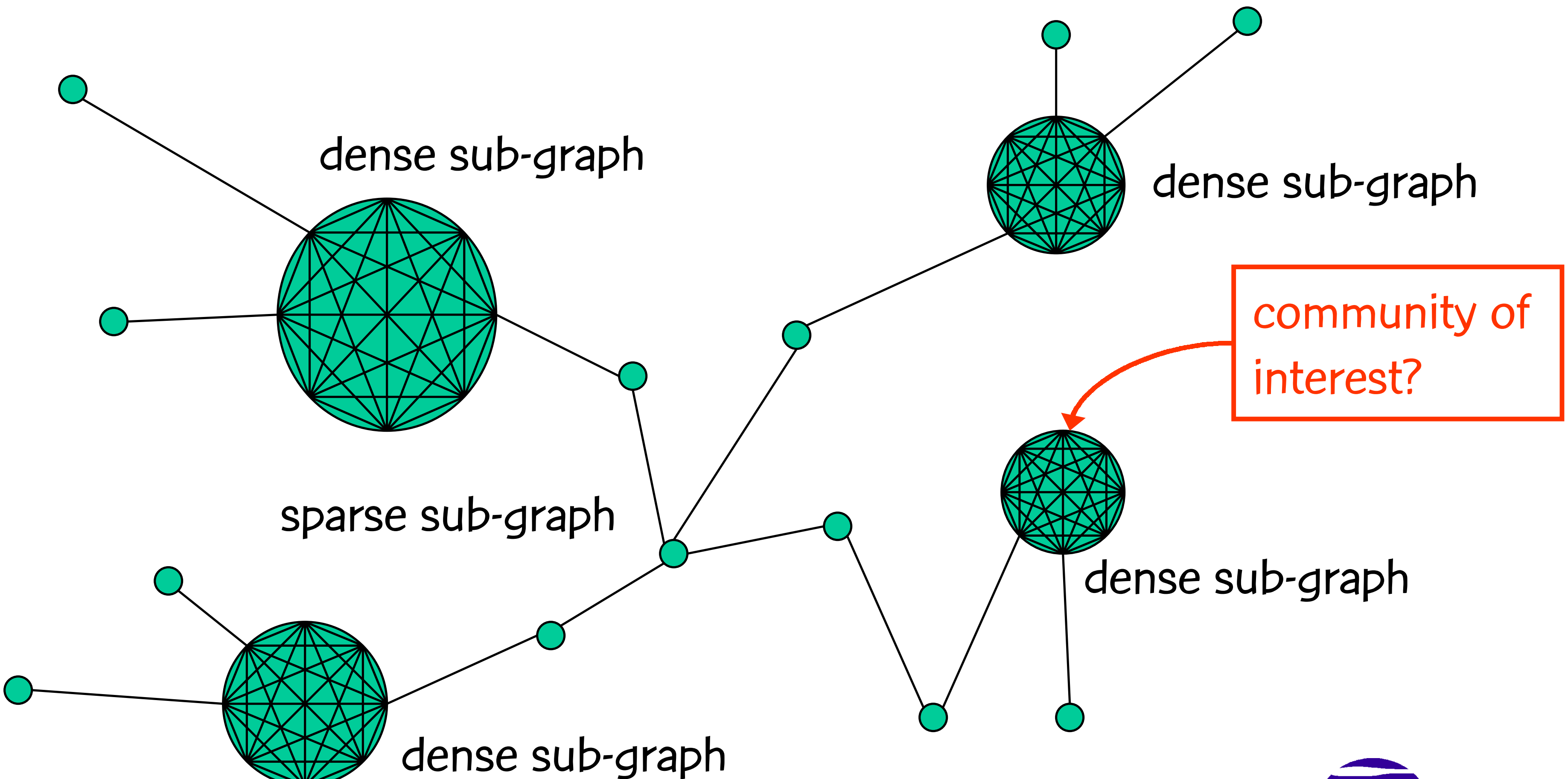
- Most subgraphs induced by DFS tree nodes are very sparse: $|E| < \log(|V|)$
- Few are dense: $|E| > \mathrm{sqrt}(|V|)$ with at most 32 nodes

Dense subgraphs

- Dense subgraphs could be
  - within G (DFS tree)
  - among different G (DFS tree)
- Counting edges:
  - most are within G (DFS tree)
  - leaves few edges between different G (DFS tree)

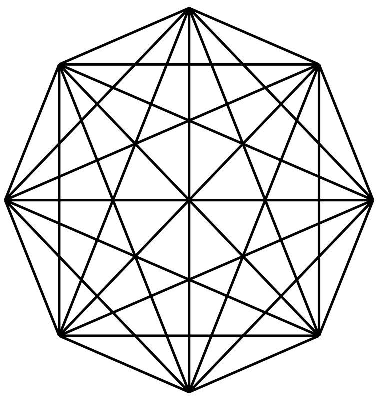


Macro structure of call detail graph

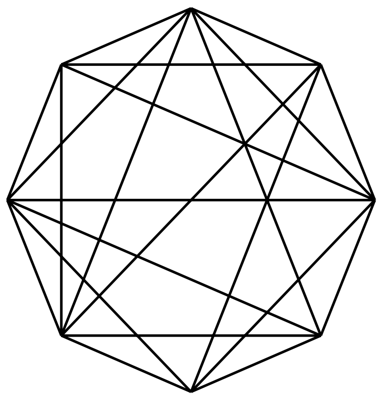


Searching for dense subgraphs

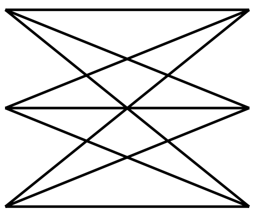
- We look for two types of subgraphs
  - cliques or quasi-cliques
  - bicliques or quasi-bicliques



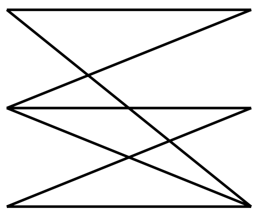
clique



quasi-clique



biclique



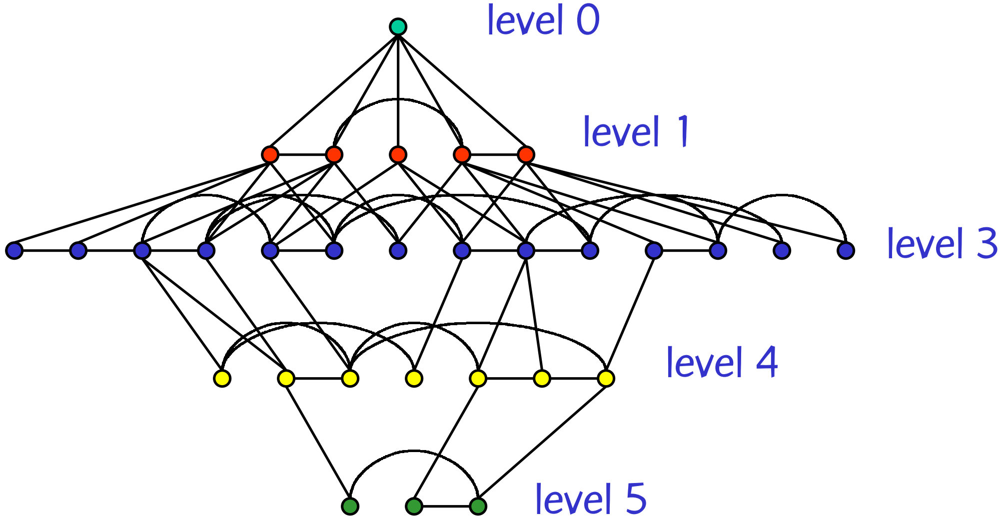
quasi-biclique

Clique case

- We illustrate the approach with the clique case.
  - We work on connected component of transmittal nodes (no cliques in sources or sinks)
  - Breadth first search decomposition
  - Peeling off vertices to focus in on large cliques
  - Finding cliques in a subgraph

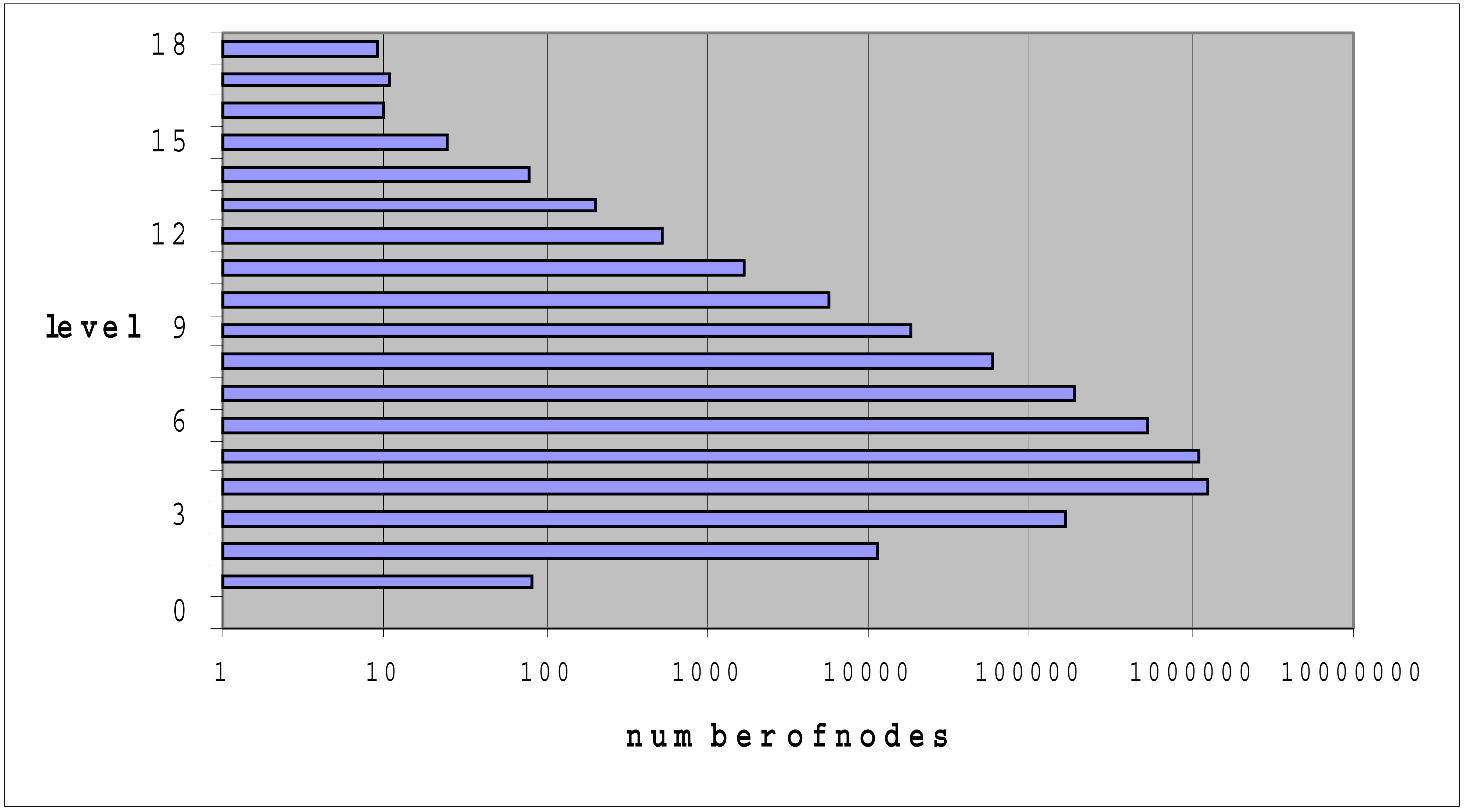
Breadth first search decomposition

- Given a graph G one can decompose its vertices into levels



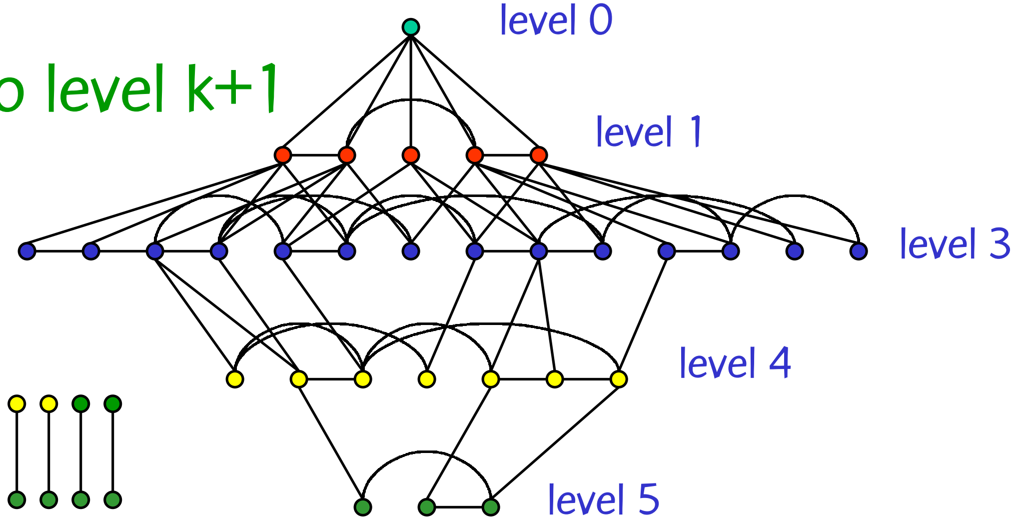
There are no cliques spanning three or more levels.

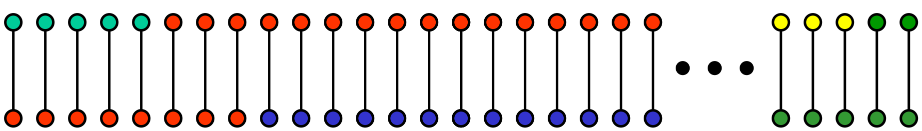
BFS: distribution of nodes per level



Edge ordering

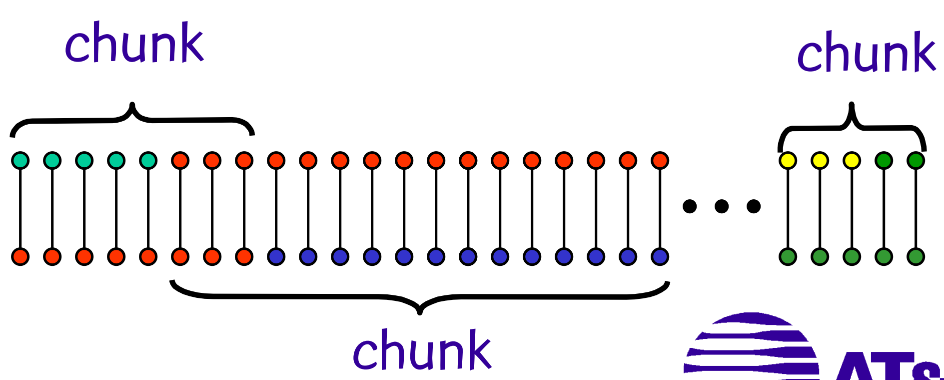
- Use levels to order edges ($k = 0,1,2,\ldots$)
  - Edges in level k
  - Edges from level k to level k+1





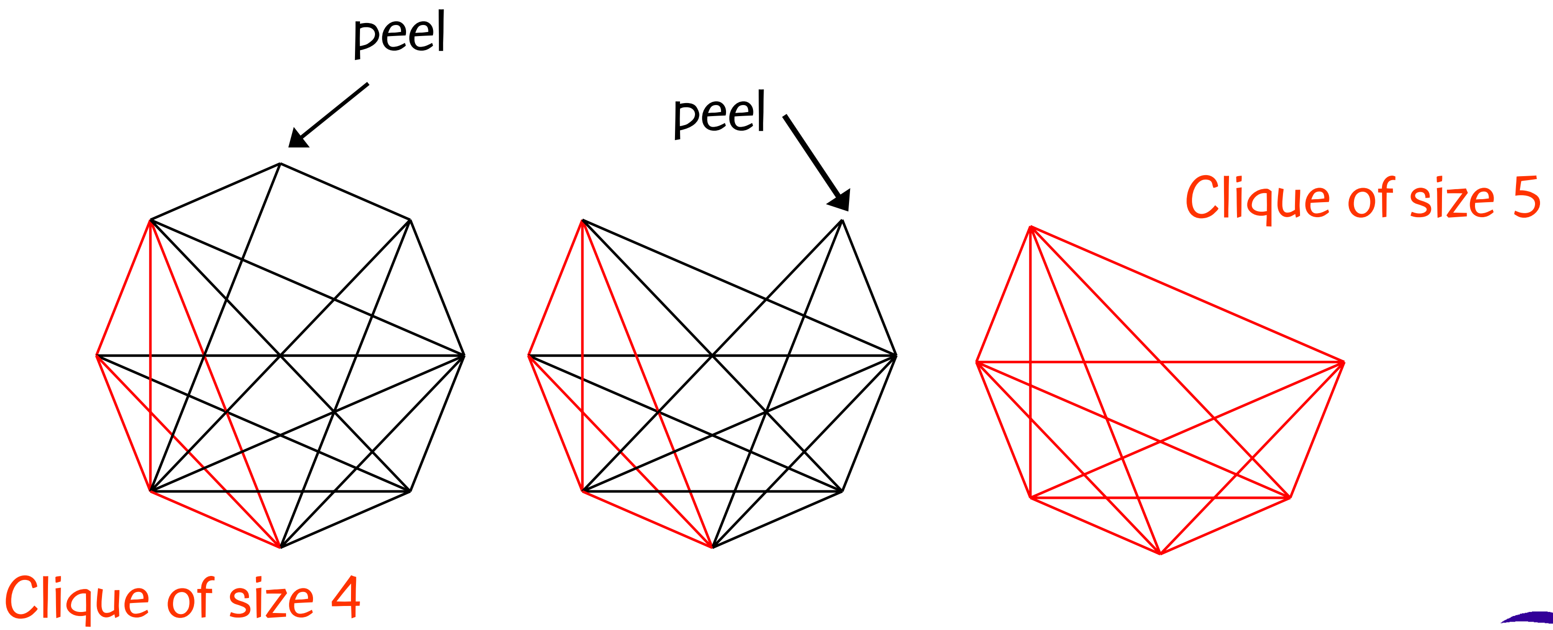
Chunking & peeling

- Start with all edges in E (set is massive)
- Repeat
  - Create a subgraph G' with one or more chunks
  - Find large clique (of size c') in G'
  - Peel from G all vertices v with $\deg(v) < c'$
  - $E = E(G)$



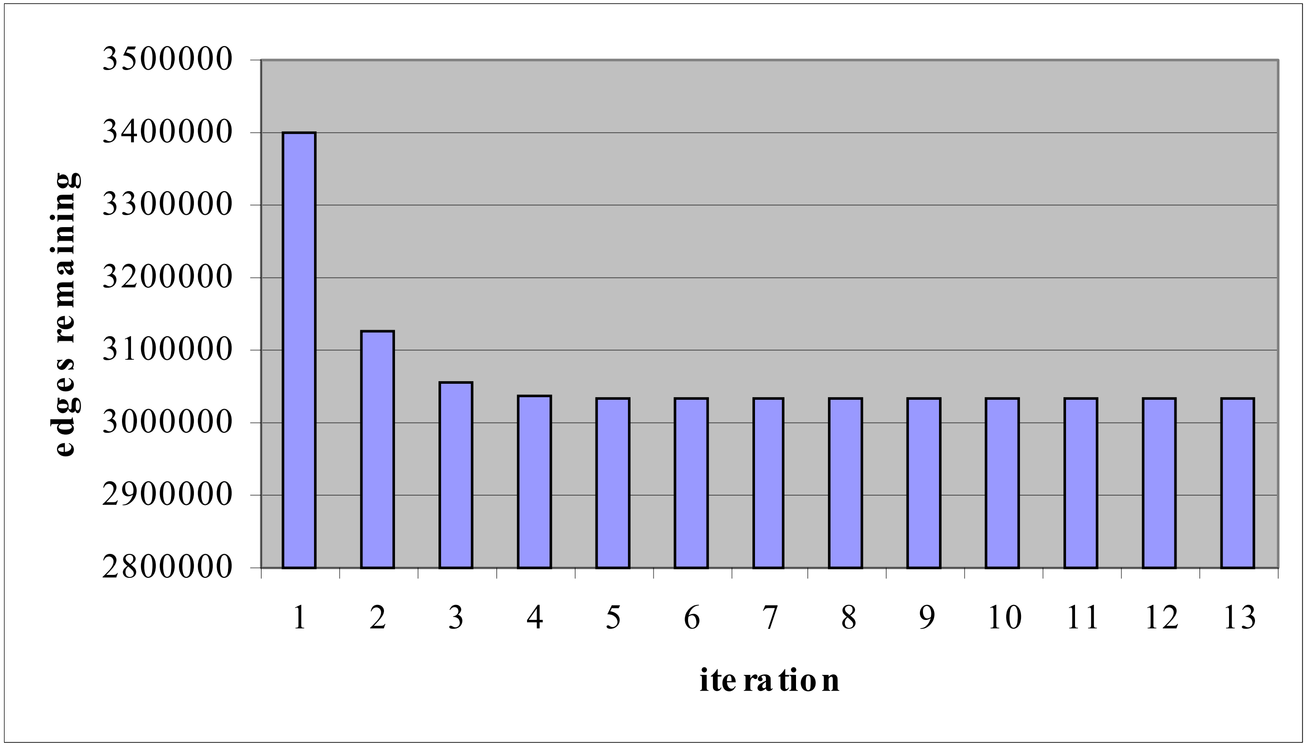
Peeling

- Peeling is applied recursively



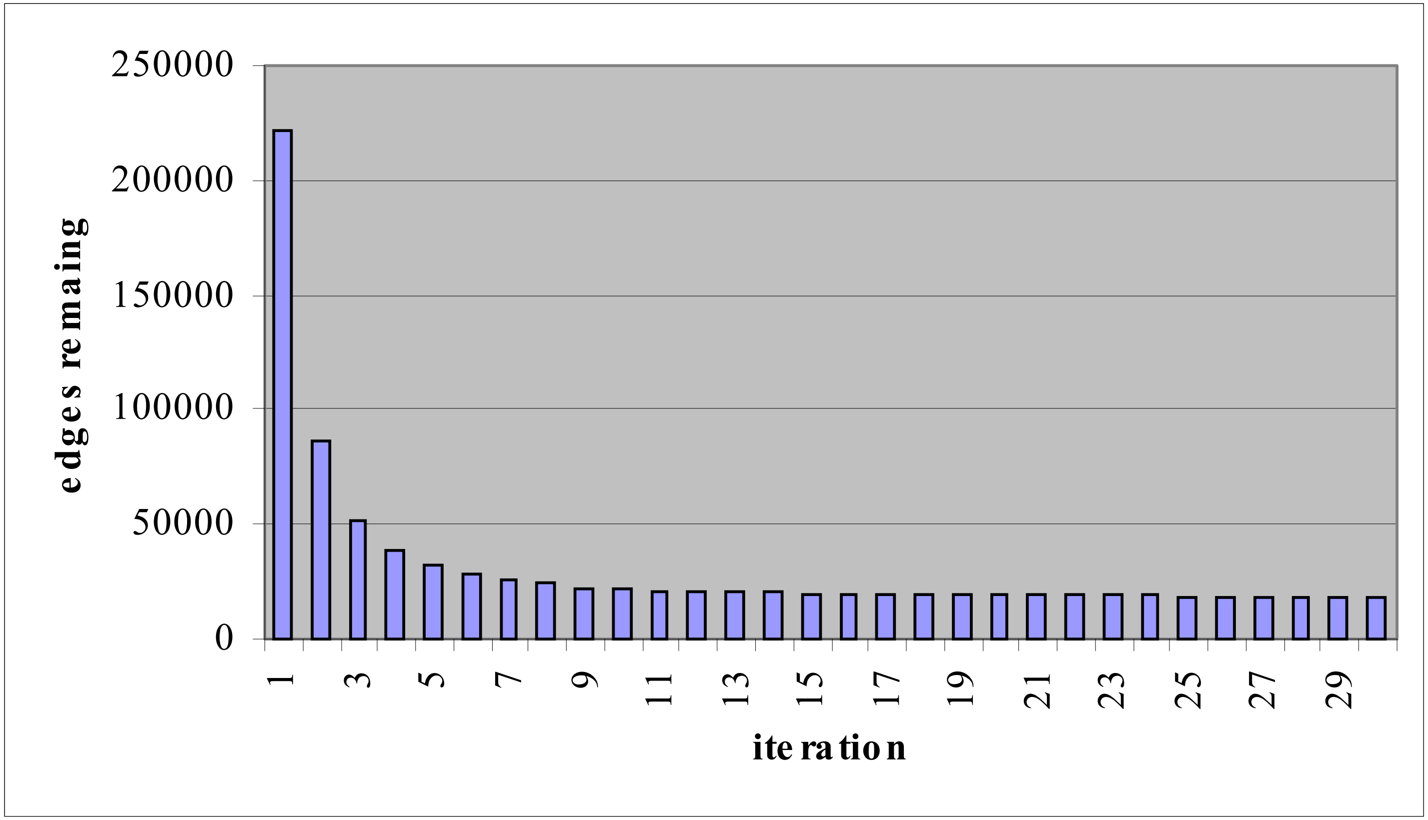
Peeling with degree = 2

reduction from 3.4 M edges to 3.0 M edges



Peeling with degree = 14

reduction from 3.0 M edges to 18.3 K edges

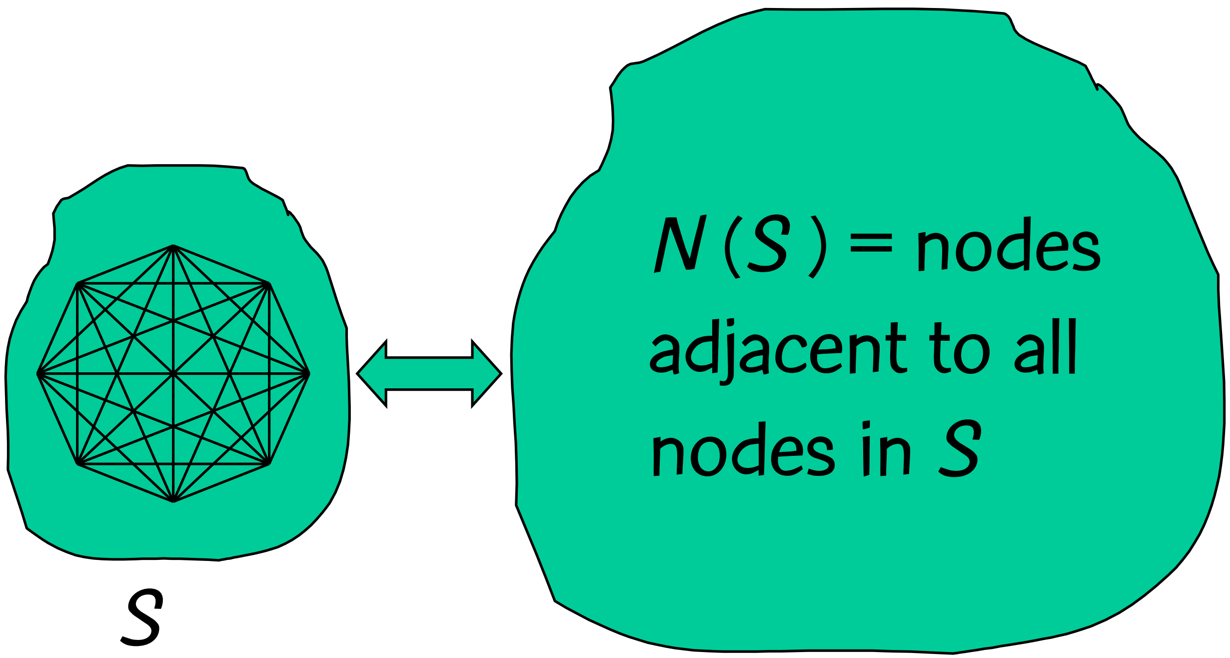


Finding cliques

- GRASP for max clique
  - multi-start
    - construct clique using randomized greedy algorithm
    - attempt to improve clique using 2-exchange local search
    - store all cliques found in construction & local search

Greedy vertex choice

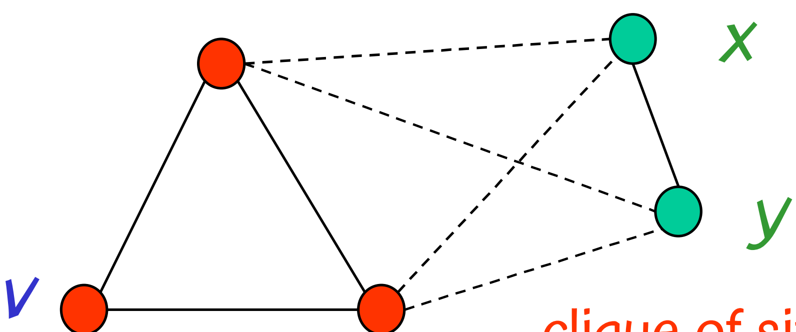
- Choose $v \in N(S)\}$ with max $\deg_{N(S)} \{v \in N(S)\}$.



(2,1) exchange local search

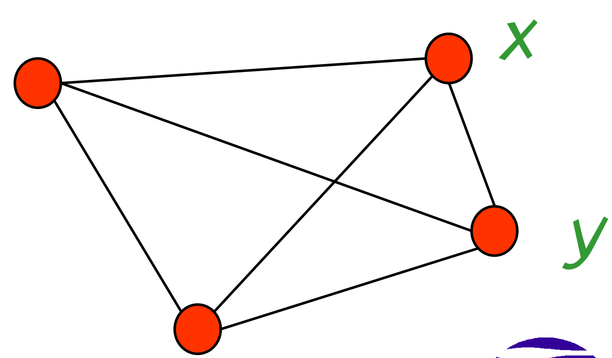
- for each vertex v in clique S
  - while $\exists$ an edge $(x, y) \in E$ with x and y adjacent to all vertices in $S \setminus \{v\}$
    - remove v from S and add x and y to S:

$$S = S \setminus \{v\} \cup \{x\} \cup \{y\}$$



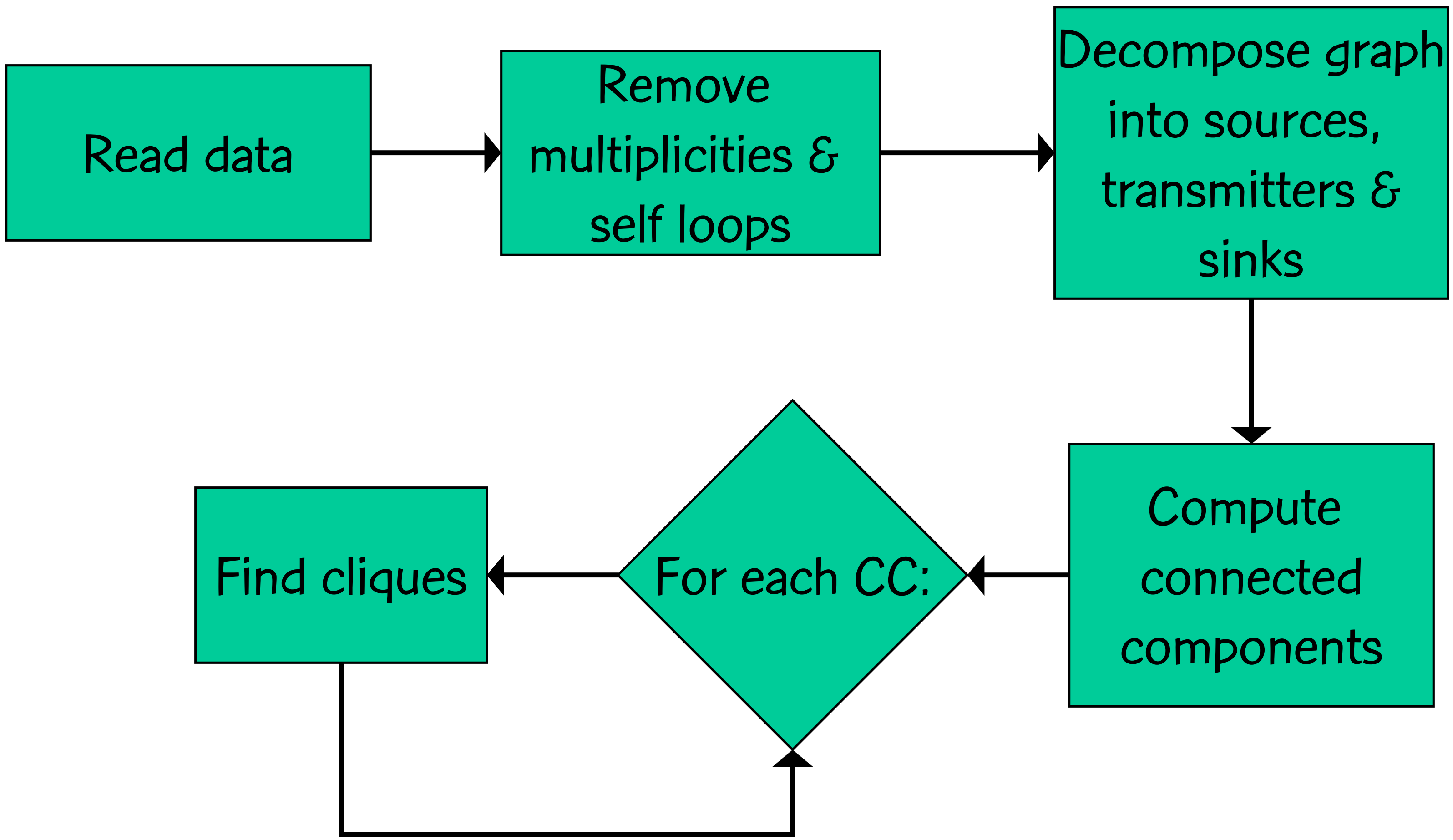
clique of size 3

clique of size 4



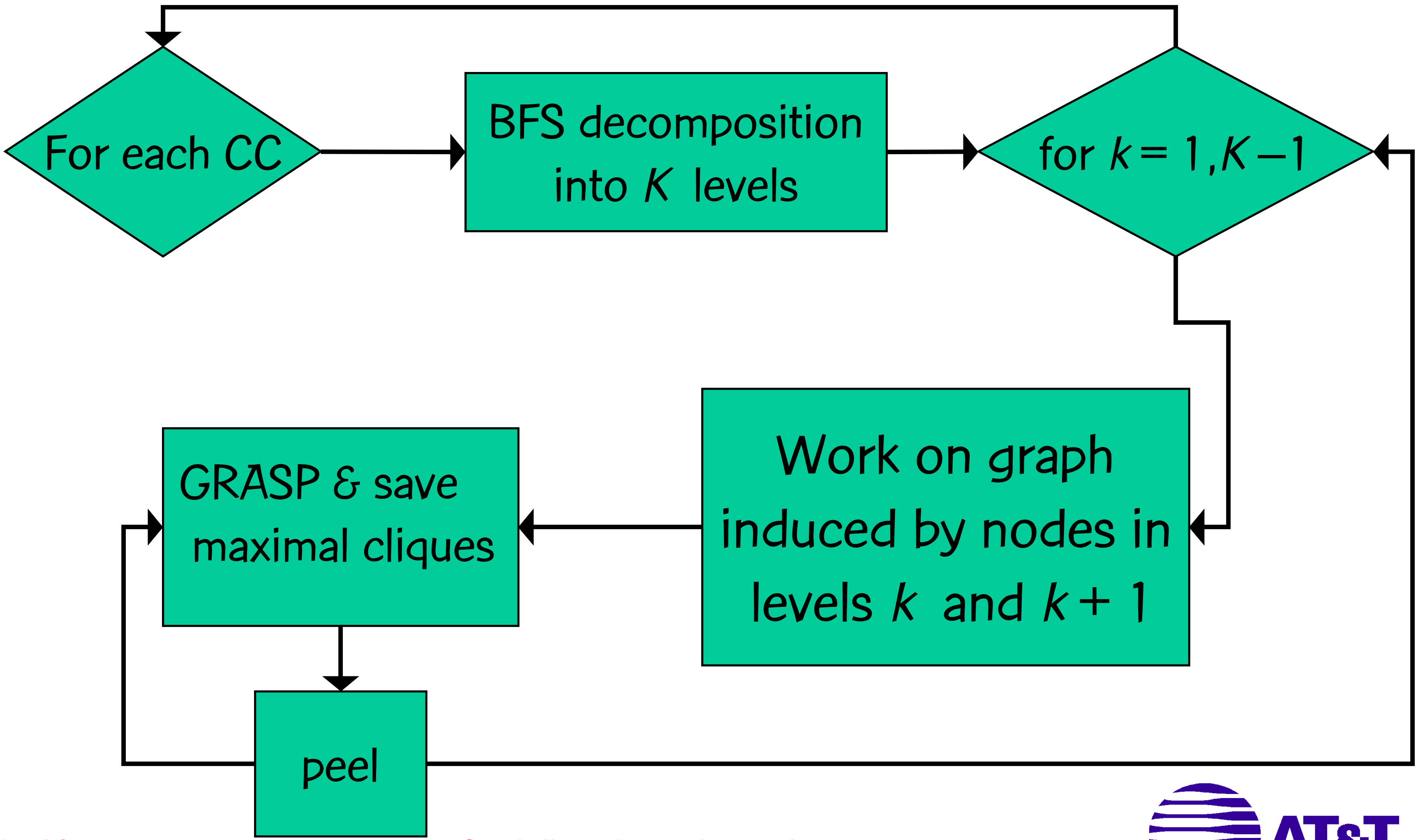
Software platform

external & semi-external memory algorithms



Software platform

computing cliques



Mining for cliques

examples

- 12 hours of calls
  - 53M nodes, 170M edges
  - 3.6M connected components (only 302K had more than three nodes)
    - 255 self loops, 2.7M pairs, and 598K triplets
  - Giant CC has 45M nodes
  - Found cliques of size up to 30 nodes in giant CC.
  - Found quasi-cliques of size 44 (90% density), 57 (80%), 65 (70%), and 98 (50%) in giant CC.

Concluding remarks

- We developed algorithms and systems for mining dense subgraphs is massive graphs.
- Subgraphs currently handled:
  - Cliques and quasi-cliques
  - Bicliques and quasi-bicliques
- We have explored data sets up to one week of calls, but aim to handle one year.
- Parallelization under way to speed up computations.