Graph theory

Victor Eijkhout

Fall 2022

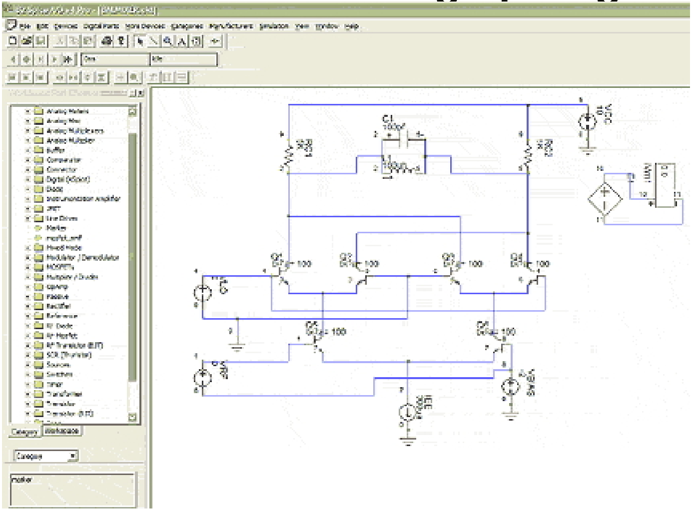
Justification

Graph theory has many applications in computational math. Here we focus on the equivalence with sparse matrices.

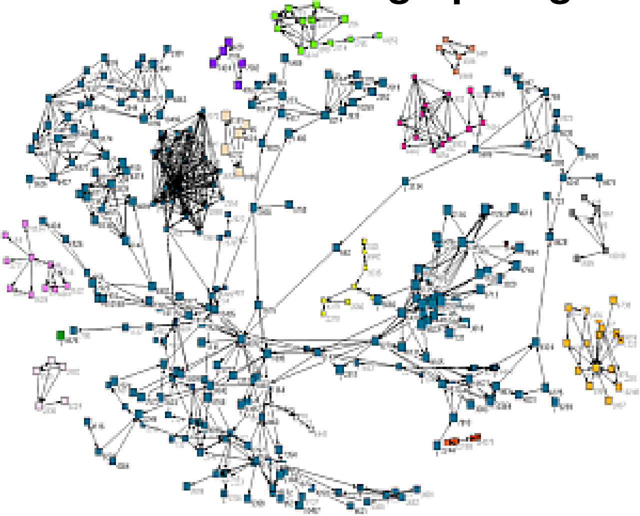
1 Graph algorithms

- Traditional: search, shortest path, connected components
- New: centrality

2 Traditional use of graph algorithms

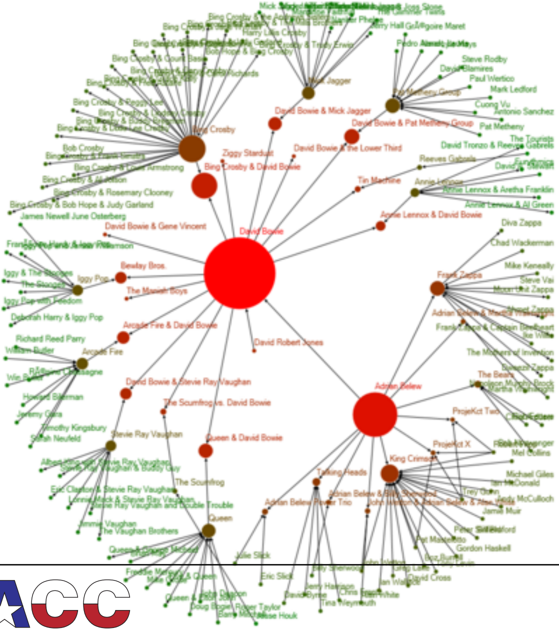


3 1990s use of graph algorithms



4 2010 use of graph algorithms

5 2010 use of graph algorithms



6 Shortest distance algorithm

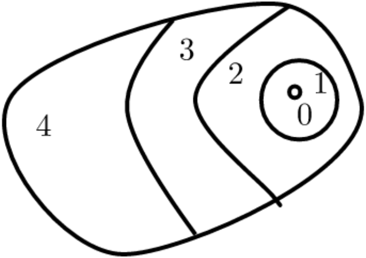
Given node s:

$$d\colon v \mapsto d(s,v)$$

Input : A graph, and a starting node s
Output: A function $d(v)$ that measures the distance from s to v
Let s be given, and set $d(s) = 0$
Initialize the finished set as $U = \{s\}$
Set $c = 1$
while *not finished* **do**
    Let V the neighbours of U that are not themselves in U
    if $V = \emptyset$ **then**
        We're done
    else
        Set $d(v) = c + 1$ for all $v \in V$.
        $U \leftarrow U \cup V$
        Increase $c \leftarrow c + 1$

7 Level sets

The steps in the algorithm are ‘level sets’:

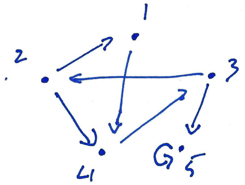


8 Computational characteristics

- Uses a queue: central storage
- Parallelism not self-evident
- Flexible assignment of work to processors, so no locality

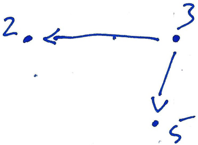
9 Example

Random graph:



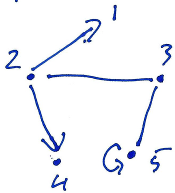
10 Level 1

Distance from 3
step 1



11 Level 2

step 2

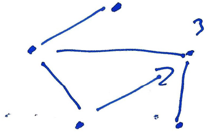


cycle?

5 reached in
1 step
2 steps

12 Level 3

step 3



cycle!
3 reached in
0 steps
3 steps

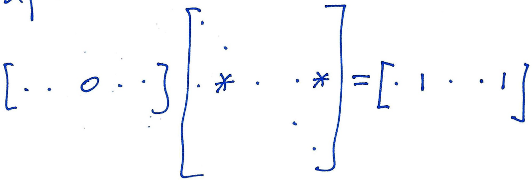
13 Matrix view

Adjacency matrix

$$\begin{bmatrix} \cdot & \cdot & \cdot & * & \cdot \\ * & \cdot & \cdot & * & \cdot \\ \cdot & * & \cdot & \cdot & * \\ \cdot & \cdot & * & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & * \end{bmatrix}$$

14 Level 1

step 1



15 Level 2

step 2

$$[\cdot\ 1\ \cdot\ \cdot\ 1]\begin{bmatrix} \cdot & & & \cdot & \\ * & \cdot & \cdot & * & \cdot \\ \cdot & & & & \\ \cdot & & & & \\ \cdot & \cdot & \cdot & \cdot & * \end{bmatrix} = [2\ \cdot\ \cdot\ 2\ \cdot] + [\cdot\ \cdot\ \cdot\ \cdot\ 2]$$

16 summing up

Summing:

$$x^t + x^t A + x^2 A^2$$

$$\begin{bmatrix} \cdot & \cdot & 0 & \cdot & \cdot \end{bmatrix}$$
$$+ \begin{bmatrix} \cdot & 1 & \cdot & \cdot & 1 \end{bmatrix}$$
$$+ \begin{bmatrix} 2 & \cdot & \cdot & 2 & 2 \end{bmatrix}$$
$$= \begin{bmatrix} 2 & 1 & 0 & 2 & 1 \end{bmatrix}$$

Gen MatVec product:

mult, reduction

$$x^t \circledast . \oplus A$$

outer-product

$$x \circledast a = \begin{cases} \bot & \text{if } a = 0 \\ x+1 & \text{if } a = 1 \end{cases}$$

$$x \oplus y = \begin{cases} x & \text{if } y = \bot \\ y & \text{if } x = \bot \\ \min(x,y) & \text{otherwise} \end{cases}$$

17 All-pairs shortest path

$$\Delta_{k+1}(u,v) = \min\left\{\Delta_k(u,v), \Delta_k(u,k) + \Delta_k(k,v)\right\}. \qquad (1)$$

Algebraically:

for *k from zero to* $|V|$ **do**
  $D \leftarrow D._{\min}\left[D(:,k) \min \cdot_+ D(k,:)\right]$

Similarity to Gaussian elimination

18 Pagerank

T stochastic: all rowsums are 1.

Prove $x^t e = 1 \Rightarrow x^t T = 1$

Pagerank is essentially a power method: $x^t, x^t T, x^t T^2, \ldots$ modeling page transitions.

Prevent getting stuck with random jump:

$$x^t \leftarrow s x^t T + (1-s)e^t$$

Solution of linear system:

$$x^t(I - sT) = (1-s)e^t$$

Observe

$$(I - sT)^{-1} = I + sT + s^2T^2 + \cdots$$

19 ‘Real world’ graphs

- Graphs imply sparse matrix vector product
- . . . but the graphs are unlike PDE graphs
- differences:
  - low diameter
  - high degree
  - power law
- treat as random sparse: use dense techniques
- 2D matrix partitioning: each block non-null, but sparse

20 Parallel treatment

- Intuitive approach: partitioning of nodes
- equivalent to 1D matrix distribution
- not scalable $\Rightarrow$ 2D distribution
- equivalent to distribution of edges
- unlike with PDE graphs, random placement may actually be good