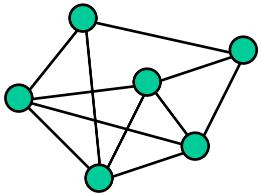


Low Diameter Graph Decompositions

N.Linial and M.Saks 1991

Seminar of Distributed Computing

ETH Zürich

16. December 2003

Josias Thöny <thoenyj@student.ethz.ch>

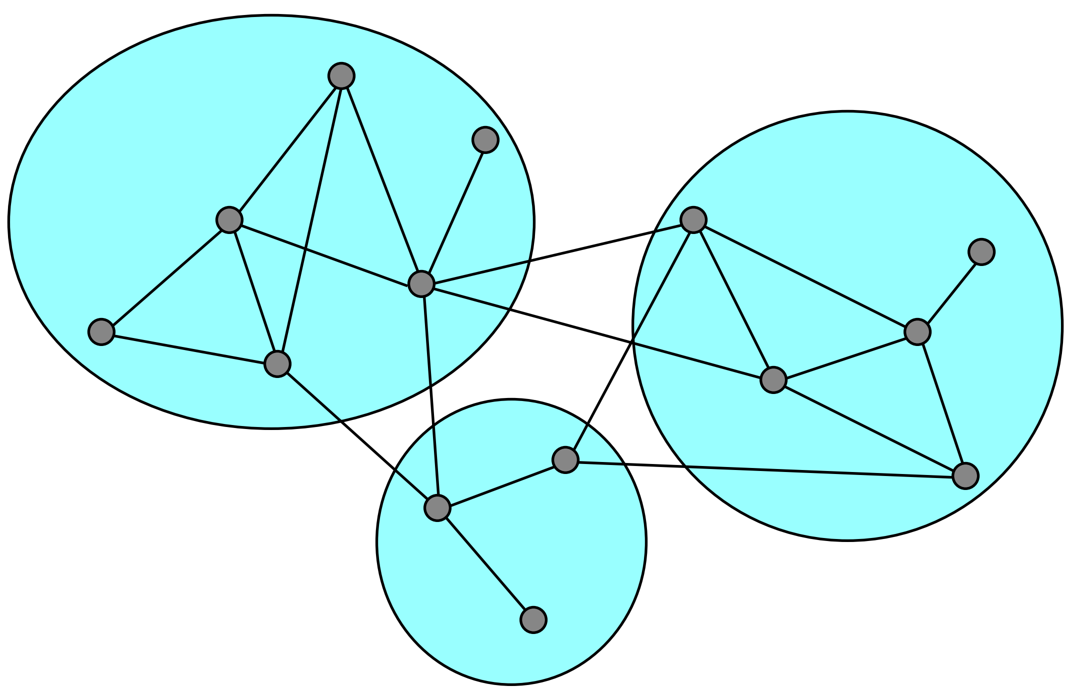
Contents

- Introduction
- Sequential algorithm
- Distributed randomized algorithm
- Summary
- Discussion

Introduction

- What is a Graph Decomposition?

  A decomposition of a graph G=(V,E) is a partition of the vertex set into subsets (“blocks”)

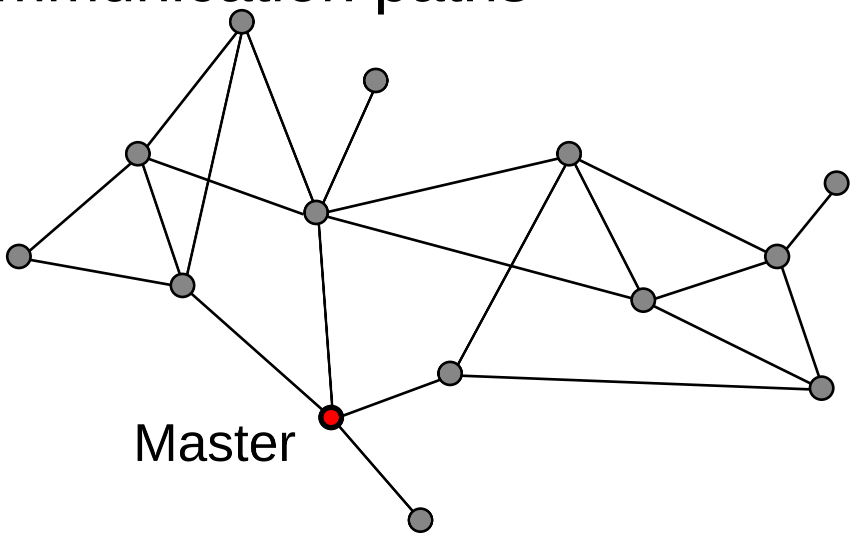


Why to decompose Graphs?

- Problem comes from **Distributed Computing** where networks are modeled as graphs
- Coordinate nodes for efficient distributed algorithms (symmetry breaking)
- How can we do this coordination?

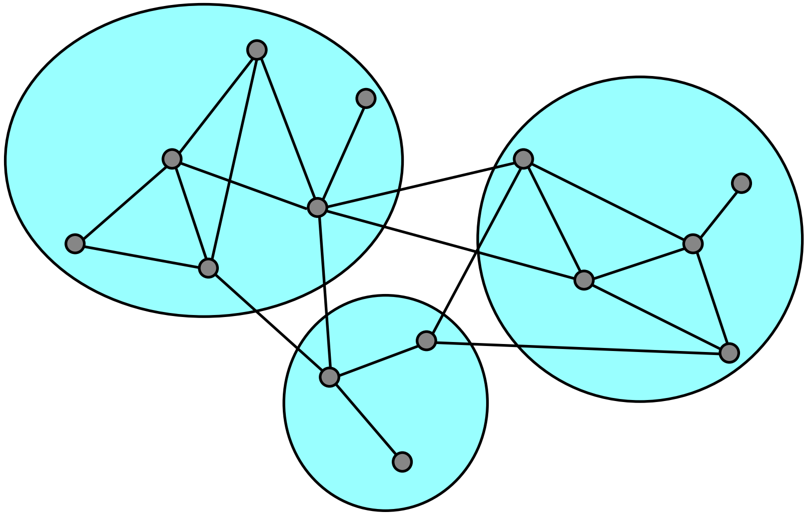
Centralization

- Choose one node to be the master and to manage all other nodes
- Drawbacks:
  - Master node is a bottleneck
  - Long communication paths



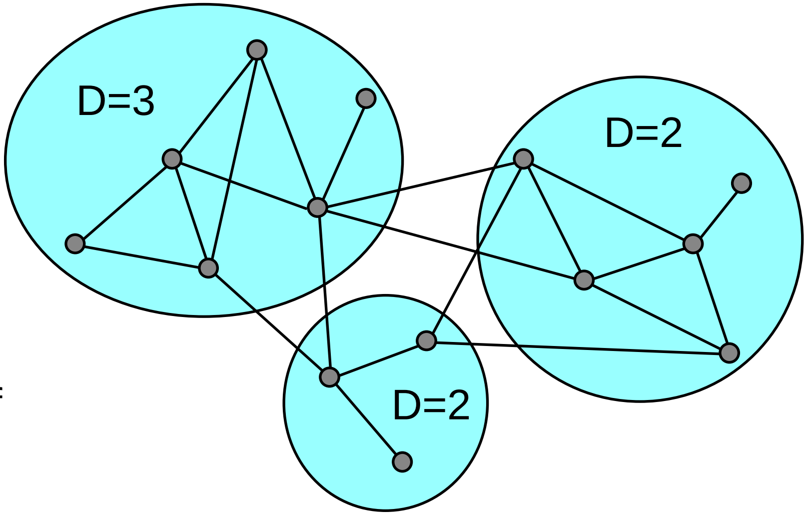
Decentralization

- Decompose the graph into regions of nearby nodes ("blocks")
- Take advantage of locality
- Efficient tool for some distributed algorithms like MIS, graph coloring, etc.



Definitions

- Graph G=(V,E), |V|=n
- Number of blocks N:
  decomposition of the graph into N blocks.
- Diameter D of a block:
  maximum distance between any two vertices in a connected component of the block
- Diameter D of a decomposition:
  maximum diameter of any of its blocks



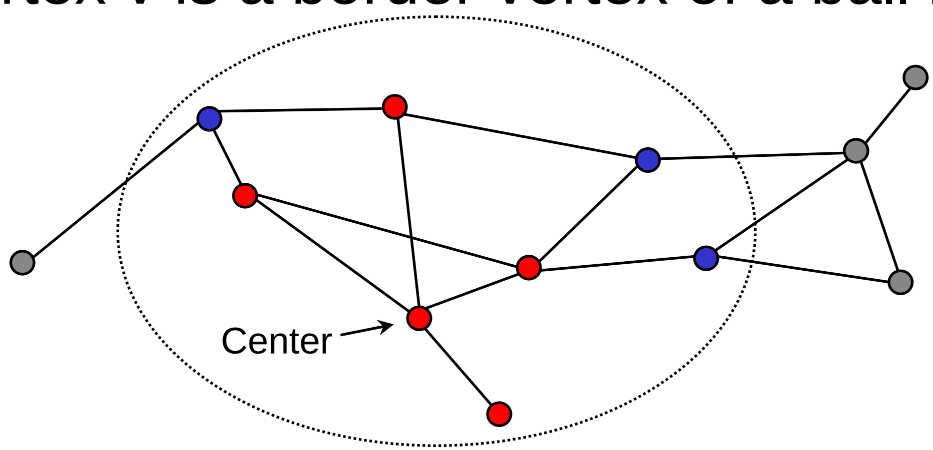
N = 3
D = max(3,2,2) = 3

Sequential Algorithm

- Goal: Find a decomposition of a graph with small number of blocks and small diameter
  (good trade-off between the two)
- Algorithm iteratively constructs one block at a time
- For each block, there is a growing phase where vertices are added to that block

Balls

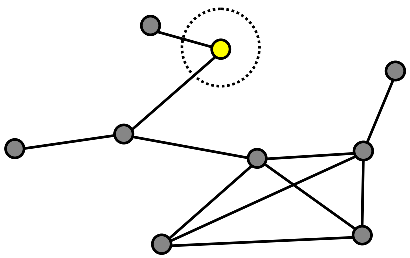
- The algorithm uses balls to decompose the graph
- Each ball has a center c and a radius r
- The ball contains all vertices of the graph which are within distance r of the center c (->volume of the ball)
- A vertex v is an inner vertex of a ball if d(v,c)<r
- A vertex v is a border vertex of a ball if d(v,c)=r



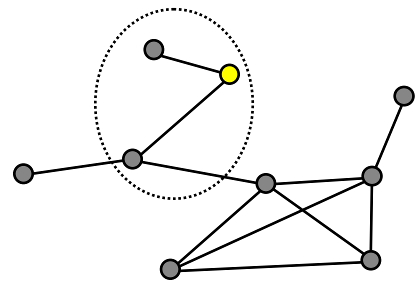
Ball of radius 2
Red: Inner vertices
Blue: Border vertices
Volume = 8

Sequential Algorithm: Example

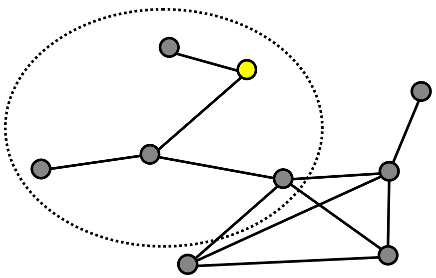
1) Choose a vertex and create a ball



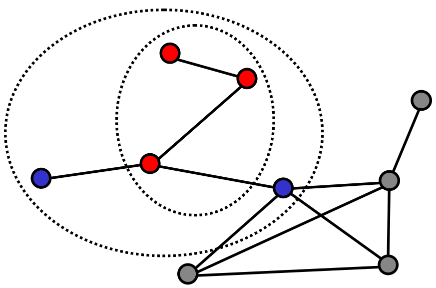
2) Increase the radius...



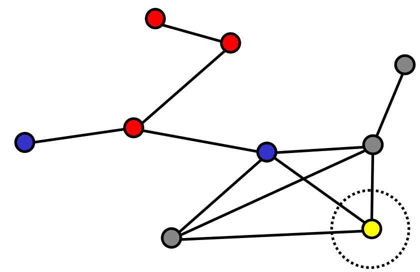
3) ...as long as the volume doubles



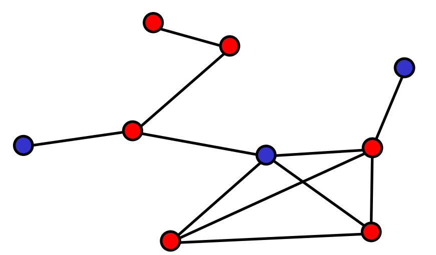
4) Color inner vertices red, border vertices blue



5) Start again with remaining vertices



...6) Red vertices form the first block

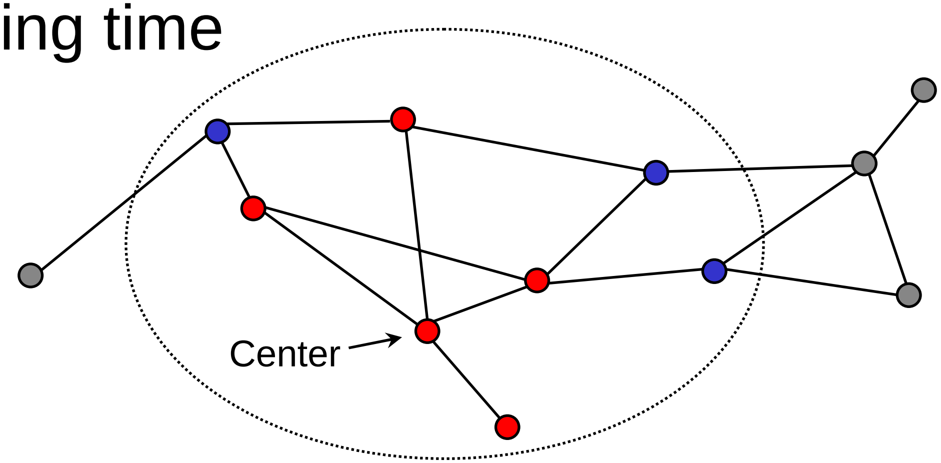


Sequential Algorithm: Analysis

- Diameter D ≤ 2 log n

  Follows from construction of balls (can double the volume only log(n) times)
- Number of blocks ≤ log n

  There are at least as many red vertices as blue ones -> each block takes at least half of all vertices
- Trade-off is optimal
- Polynomial running time



Distributed Algorithm

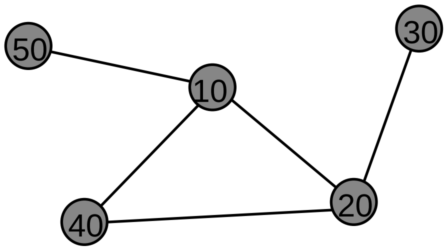
- Goal: same quality of decomposition: diameter and number of blocks are both O(log n)
- How can we use the idea of balls?
- Randomized algorithm
- Problem of overlapping balls if every vertex starts to create a ball at the same time
- Idea: Use IDs and join the ball with highest ID

Distributed Algorithm (2)

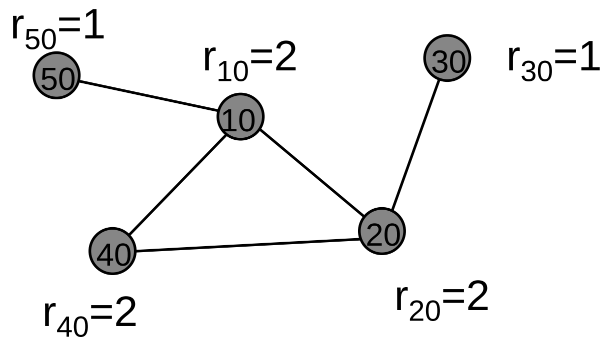
- Algorithm is a sequence of stages; in each stage every vertex decides whether to join the current block
- First every vertex v chooses a random radius r_v to create a ball and sends the pair (ID, r_v) to all neighbors within distance r_v
- Then each vertex v chooses the vertex C(v) of highest ID among all received pairs (ID, r) and joins the current block if $d(v, C(v)) < r_{C(v)}$
(v is an inner vertex of the ball around C(v))

Distributed Algorithm: Example

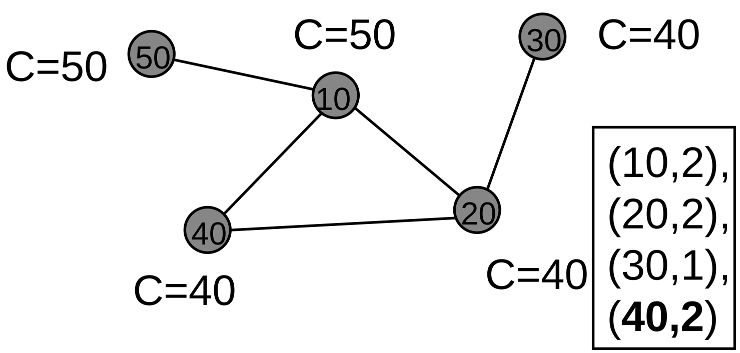
1) Graph with IDs



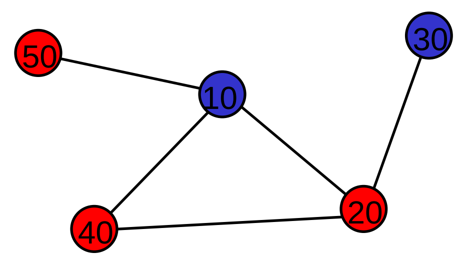
2) Choose random radius



3) Multicast, then select highest ID



4) Inner vertices join block



Analysis: Diameter

- Probability distribution of the radius:

$$r_v = \begin{cases} k \text{ with probability } p^k(1\text{-}p), \text{ for } k < \log n \\ \log\ n \text{ with probability } p^{\log\ n} \end{cases}$$

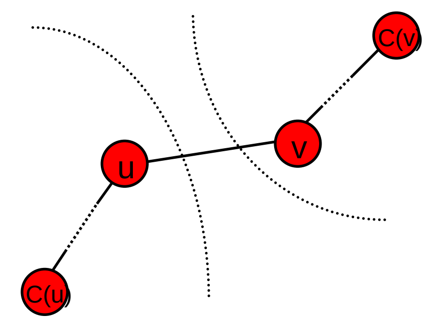
where $p = \frac{1}{2}$

- -> Diameter of one ball ≤ 2log n
- Diameter of a block ≤ 2log n, because:

Claim: Every vertex of a connected component in a block has the same center.

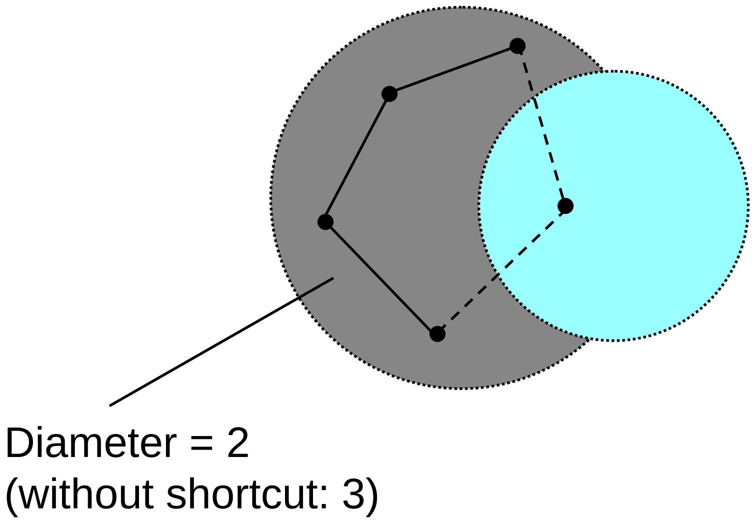
Proof by contradiction: Assume there are two neighbor vertices u, v with different centers C(u), C(v). Assume C(u) has a higher ID than C(v). u is an inner vertex of the ball around C(u) -> v must have received the message

from C(u) and joined that ball -> contradicicition



Diameter

- This algorithm uses a slightly different definition of the diameter:
  It's allowed to have shortcuts through vertices from a different block. (Necessary because of overlapping balls)

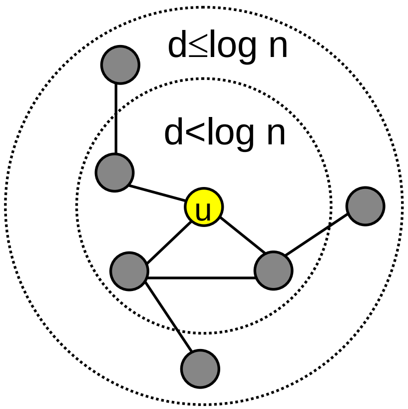


Number of Blocks (1)

- Overview:
- Estimate the probability that a vertex u is assigned to the current block B: Pr[u $\in$ B]
- Estimate the probability that all vertices are assigned to some block after i rounds (algorithm is over)
- Show that this probability is high after a logarithmic number of rounds
  (-> algorithm creates logarithmic number of blocks)

Number of Blocks (2)

Estimate the probability that vertex u is assigned to the current block B: $\Pr[u \in B]$
-> Consider neighborhood of u:



$$\Pr[u \in B] = \sum_{v|d(u,v)<\log n} \Pr[u \in B | C(u) = v] \Pr[C(u) = v]$$

$$\Pr[u \in B | C(u) = v] = \Pr[d < r_v | d \le r_v] \overset{d=d(u,v)}{=} \frac{\Pr[d < r_v]}{\Pr[d \le r_v]} = \frac{p^{d+1}}{p^d} = p = \frac{1}{2}$$

$$\Rightarrow \Pr[u \in B] = p \sum_{v|d(u,v)<\log n} \Pr[C(u) = v] = p \Pr[d(C(u), u) < \log n]$$

$$\ge p \Pr[r_v < \log n, \forall v] = p(1 - p^{\log n})^n = \tfrac{1}{2}(1 - \tfrac{1}{n})^n \approx \frac{1}{2e}$$

Number of Blocks (3)

- Pr[vertex u is assigned to a block B] = Pr[u $\in$ B] = q $\approx$ 1/2e
- Pr[vertex u is not assigned to some block after i rounds] = $(1\text{-}q)^i$
- Pr[all vertices assigned to some block after i rounds] $\geq$ $1\text{-}n(1\text{-}q)^i$ (algorithm terminates -> number of blocks=i)

- Choose the number of rounds (i) to be logarithmic in n: i= c $\log_Q$ n
- Pr[logarithmic number of blocks] $\geq$ $1 - n(1-q)^{c \log_Q n}$

$$\approx 1 - n(1 - \tfrac{1}{2e})^{c \log_Q n} = 1 - n(\tfrac{1}{2e(2e-1)})^{c \log_Q n} \overset{c=2, Q=2e(2e-1)}{=} 1 - n\frac{1}{n^2} = 1 - \frac{1}{n} \approx 1$$

Distributed Algorithm: Remarks

- Nodes have to know when they've received all messages from their neighbors (can be solved with synchronization techniques)
- All nodes have to start the algorithm at the same time
- Nodes must know n

Summary

- Sequential algorithm:
  - Number of blocks $\leq$ log n
  - Diameter of blocks $\leq$ 2log n
  - Running time: polynomial
- Distributed randomized algorithm:
  - Number of blocks O(log n)
  - Diameter of blocks O(log n)
  - Running time $O(\log^2 n)$

Discussion