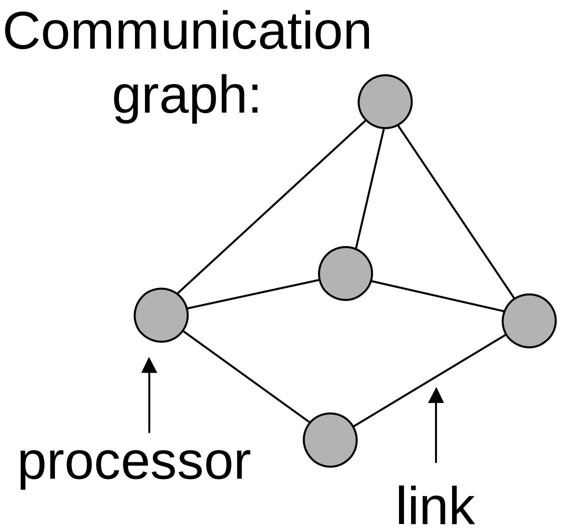
Distributed approximation in trees and planar graphs.

Andrzej Czygrinow,
Michał Hanćkowiak

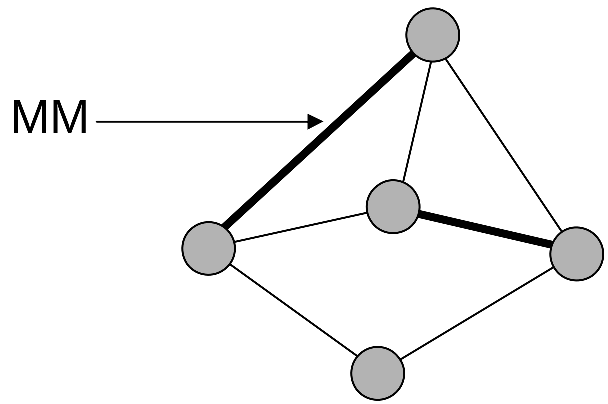
Model of computation



- Distributed (message passing).
- Synchronous:
  computation = a sequence of *rounds*;
  in each round, each processor does:
  - Sends messages to its neighbours.
  - Receives messages from its neighbours.
  - Does local computations.
- Randomisation not allowed.

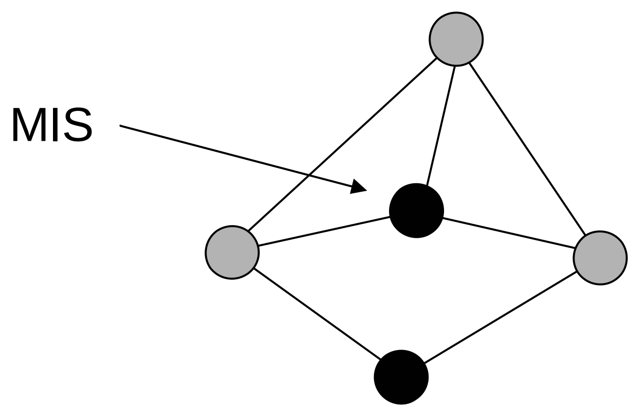
Definitions of problems

- **MM** = Maximum Matching
  - the largest set of disjoint edges
- **MWM** = Maximum Weighted Matching
  - weights are assigned to edges
  - MWM = the heaviest set of disjoint edges



Definitions of problems

- **MIS** = Maximum Independent Set
  - set S of vertices is *independent* iff $\underset{v,w\in S,v\neq w}{\forall} \quad d(v,w)\geq 2$
  - MIS = the largest independent set
- **MWIS** = Maximum Weighted Independent Set
  - vertices have weights
  - MWIS = independent set of the largest (total) weight



What we have done

- almost exact approximation for problems:
  - MWM – in **trees**
  - MWIS – in **planar graphs**
- what is "almost exact approximation" ?
  (for MM problem)

$$\frac{|M|}{m(G)} > \left(1 - \frac{1}{\log n}\right)$$

where

M – matching found by our algorithm in G
m(G) – size of Maximum Matching in G

(in other words: approximation ratio is going to 1)

Basic method: clustering

- approximating MM in trees

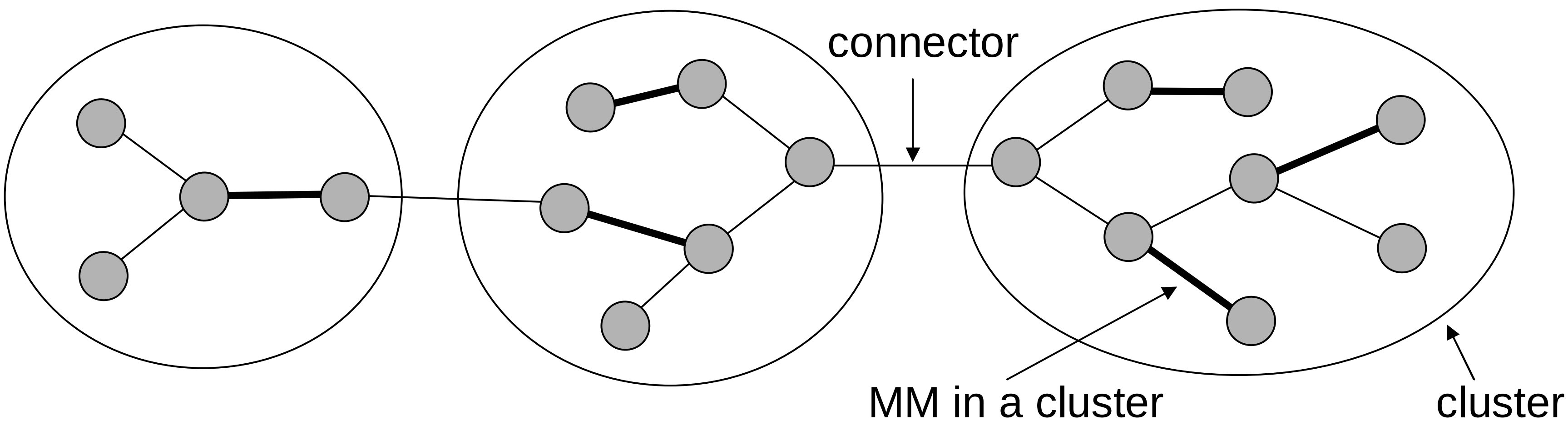
  this is only an example!,
  approximating MWM in trees is much harder ...

procedure MM_InTree

1. partition set of vertices into **clusters**
   with lower and upper bounds for diameter

$$\log n < diam(X) < \log^2 n$$

2. in each cluster, in parallel, compute (optimal) MM
3. (*in some problems, but not in MM*) modify solutions in clusters to get proper solution in the whole input graph



Basic method: clustering

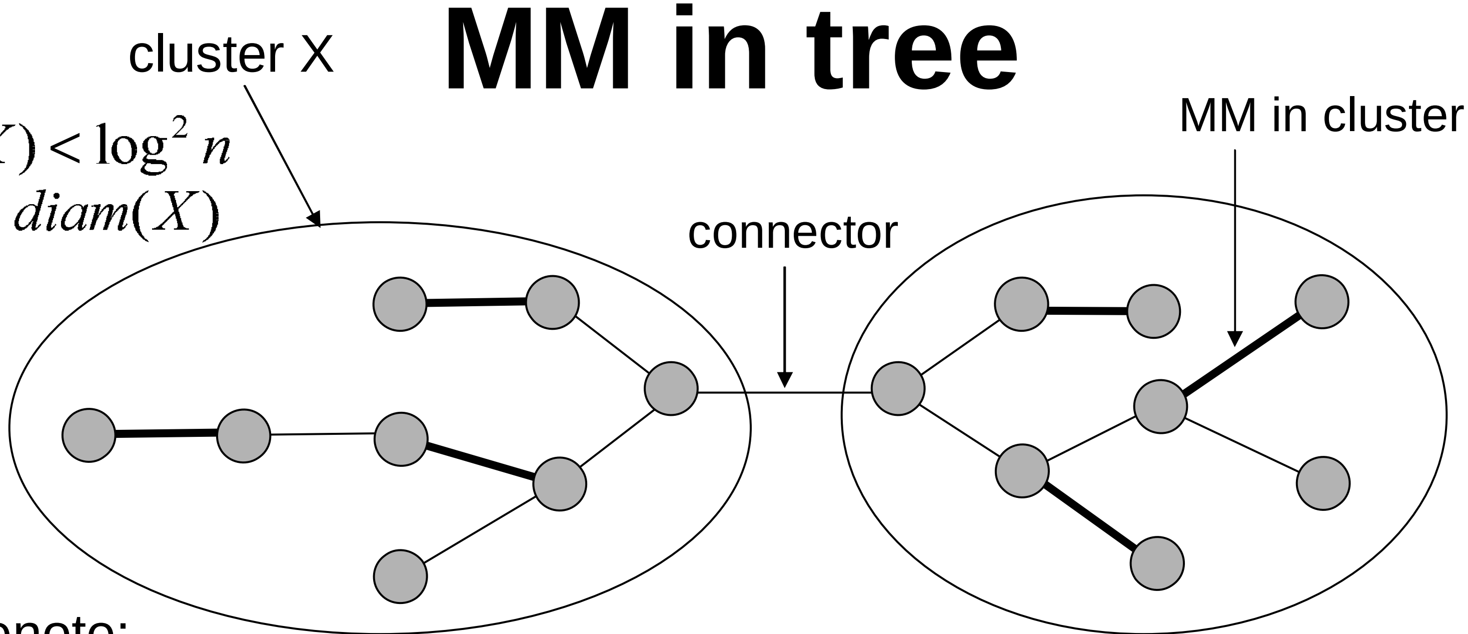
- what is a cluster?
  - subset of vertices X such that graph induced by X is connected
  - clusters have lower and upper bounds for diameter

$$\log n < diam(X) < \log^2 n$$

- how to compute clusters?
  - Rulling Forest, see Awerbuch
    (α,β)-Rulling Forest is a spanning forest such that:
    1. each tree is a rooted tree, and two roots are at distance > **α**
    2. each tree has diameter < **β**
  - it is possible to compute (k, k log n)-Rulling Forest in time O(k log n) in distributed, synchronous model of computation
    (we take k=log n)
  - lower bound for diameter of clusters also holds because:
    (α,β)-RullingForest = (α,β)-RullingSet + BFS

MM in tree

$diam(X) < \log^2 n$
$\log n < diam(X)$



lets denote:

T – input tree; m(G) – size of a MM in graph G

C - set of clusters; L – set of connectors;

M – matching found by our algorithm in T

facts:

F1 $m(T) \le |M| + |L|$

F2 $\forall_{x \in C} m(X) > \frac{1}{10} \log n \Rightarrow m(T) > |C| * \frac{1}{10} \log n$

cluster graph is also a tree !!!
|C|=|L|+1

therefore:

$$\frac{|M|}{m(T)} \ge \frac{m(T) - |L|}{m(T)} = 1 - \frac{|L|}{m(T)} \ge 1 - \frac{|C|}{m(T)} > 1 - \frac{10}{\log n}$$

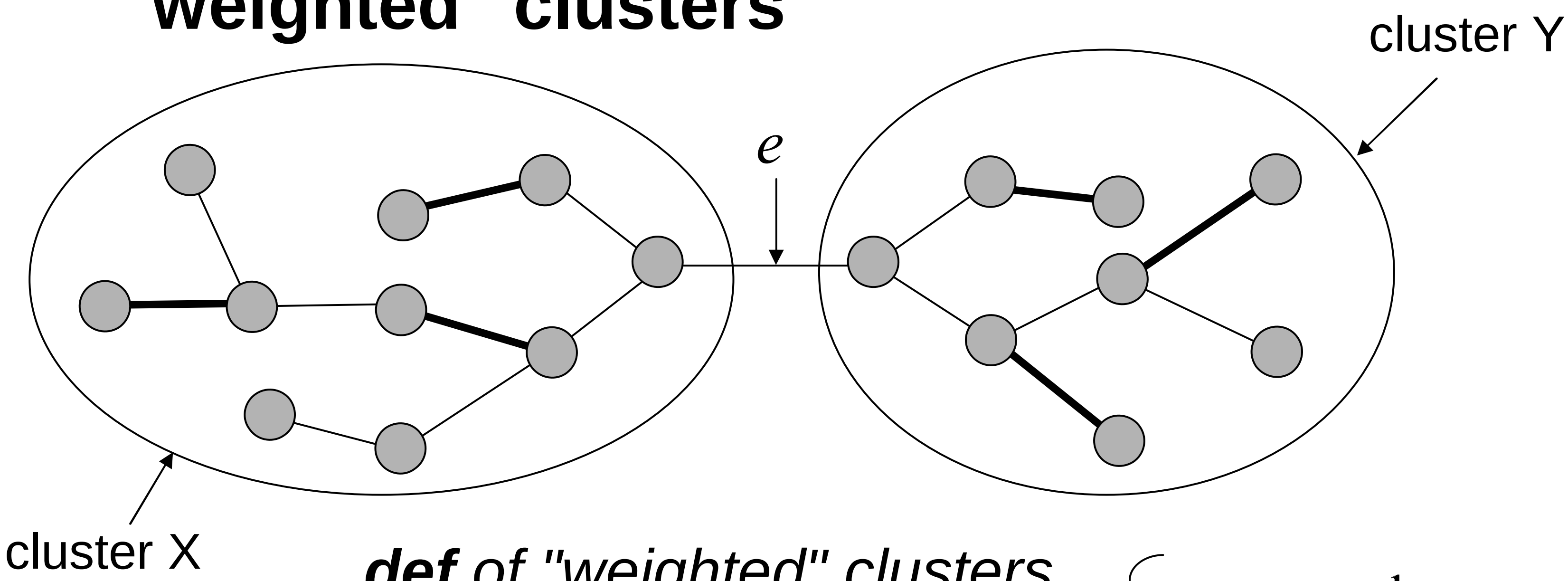
MM in tree

- this method works also for (unweighted) vertex problems ...
- doesn't work for Maximum Weighted Matching
  - connectors can have very big weights...
- doesn't work for planar graphs
  - number of connectors may by much grater than number of clusters...

MWM in tree

- special kind of clusters that depends on weights:

  "weighted" clusters



***def** of "weighted" clusters*

for each connector *e*,
connecting clusters X and Y:

$$\begin{cases} wm(X) > \dfrac{\log n}{20} w(e) \\ wm(Y) > \dfrac{\log n}{20} w(e) \end{cases}$$

wm(G) = weight of the heaviest matching in G

MWM in tree

- aproximating MWM in tree

  procedure MWM_InTree

  1. partition set of vertices into "weighted" clusters
  2. in each cluster, in parallel, compute (optimal) MWM

- facts:

  F1 we can choose a root R of T and orient all edges towards R; for cluster X lets denote its (one) outgoing edge by e_x

  F2 $$wm(T) \le w(M) + \sum_{x \in C} w(e_x) = \sum_{x \in C} \left[wm(X) + w(e_x) \right] \le$$
  $$\le \sum_{x \in C} \left(1 + \frac{20}{\log n} \right) wm(X) \le \left(1 + \frac{20}{\log n} \right) w(M)$$

- therefore:

$$\frac{w(M)}{wm(T)} \ge \left(1 - \frac{20}{\log n} \right)$$

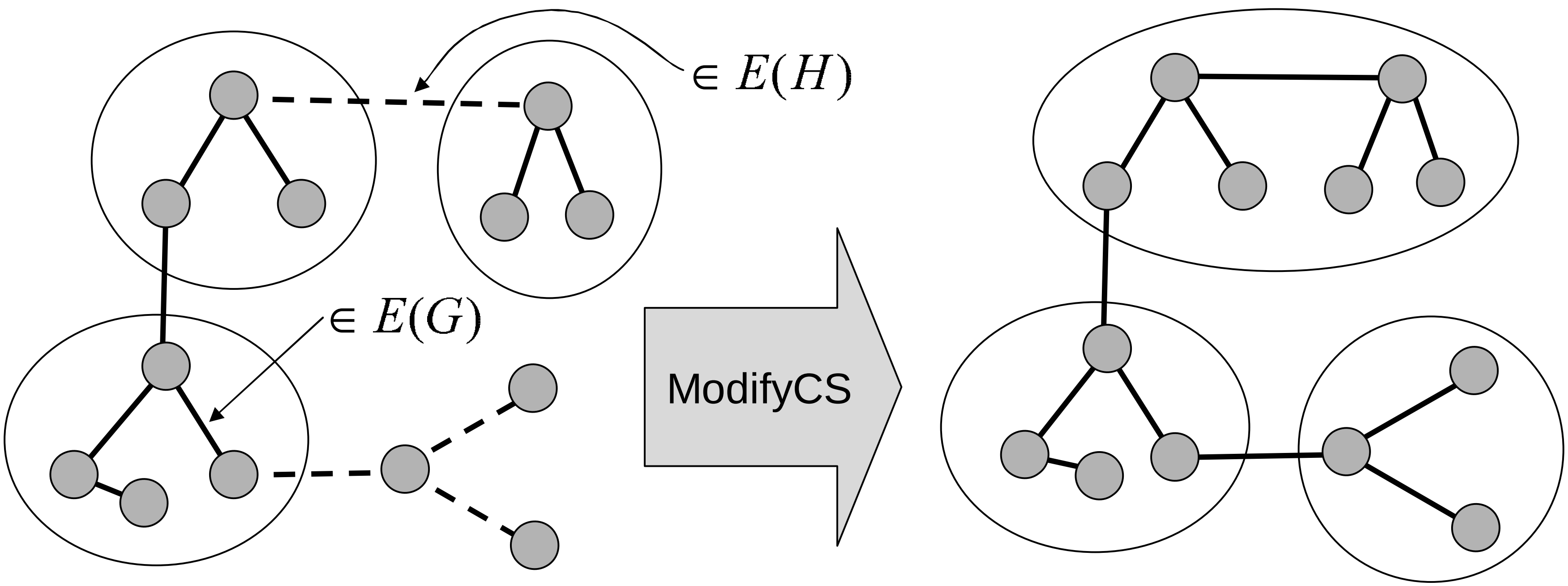
wm(G)	- weight of the heaviest matching in graph G
w(S)	- weight of a subgraph S
T	- input tree
C	- set of "weighted" clusters
M	- output of our procedure

MWM in tree

- how to compute "weighted" clusters ?
  - Rulling Forest is useless ...
  - we use helper procedure that *extend clusters to new edges*:

procedure ModifyClusterSet
input: graphs G and H,
set of clusters C in G
output: set of clusters C**'** in G $\cup$ H

for x$\in$C: diam(x) $\leq$ p
for y$\in$C**':**
diam(y) < p+log^{2}n
diam(y) > log n



MWM in tree

- to compute "weighted" clusters we use LightRF

  that iterates over disjoint sets of edges $H_0, H_1, \ldots, H_{\log n}$
  and at iteration i it extends clusters to new edges from H_i

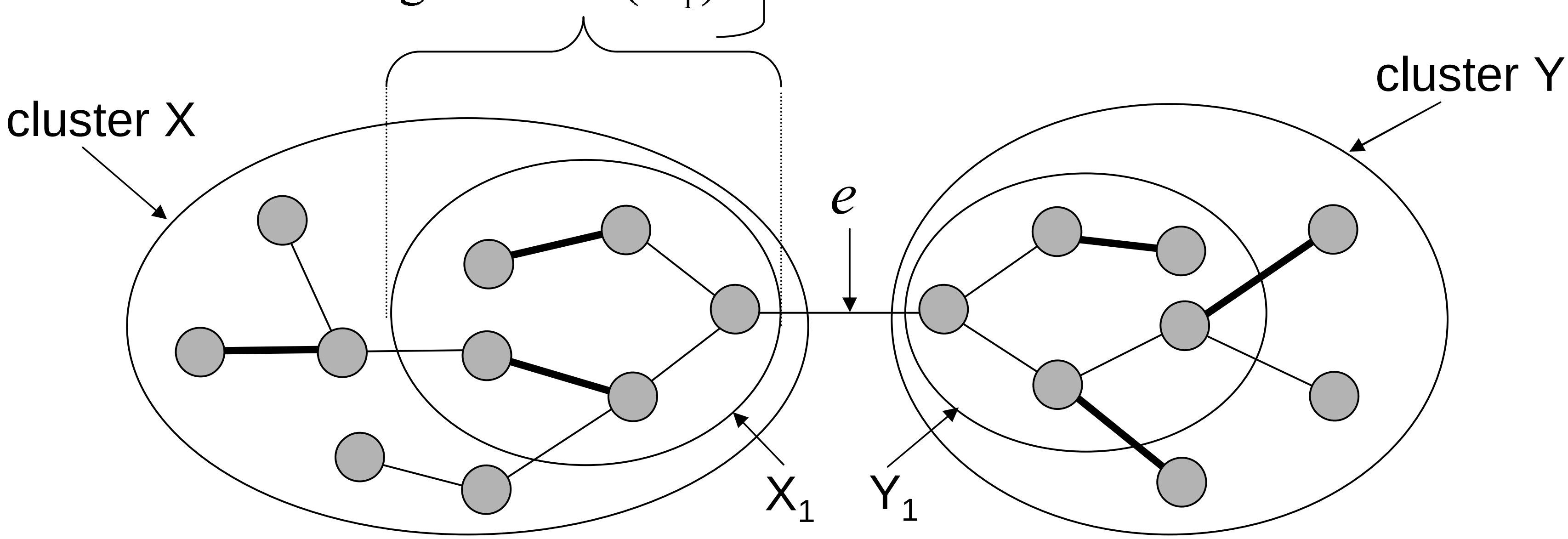
procedure LightRullingForest

1. $C := \{\{v\} : v \in V(T)\}$ — empty clusters (single vertex clusters)
2. $G := \emptyset$
3. For i:=0 to log n do
   - $H_i := \{e \in T : w(e) \in (\frac{M}{2^{i+1}}, \frac{M}{2^i}]\}$ — M: max weight of an edge
   - C' := ModifyClusterSet(G, H_i, C)
   - $G := G \cup H_i$; C := C'

MWM in tree

- why clusters produced by **LightRullingForest** are "weighted" clusters ?
  - let **e** be a connector in clusters returned by LightRF
  - **e** belongs to some $\mathbf{H_i}$, therefore in i-th iteration of LightRF it connects two temporary clusters X_1 and Y_1 composed from edges with weights > 1/2w(e)

$$\left.\begin{array}{r}\forall_{r\in E(X_1)} w(r)\ge \frac{1}{2}w(e)\\ \log n < diam(X_1)\end{array}\right\}\Rightarrow wm(X) > \frac{\log n}{20}w(e)$$



MWM in tree

procedure ModifyClusterSet

input: G, C, H; output: C'

repeat O(log n) times:

1. compute large independent set of clusters "I" in a cluster graph T_C
2. each cluster $c \in I$ select one of its connectors e_c connecting him with a cluster d_c
3. for all $c \in I$ do in parallel:
   - if for all $v \in c$: $dist_{G \cup H}(v, d_c) \leq \log n$ then move vertices from c to d_c, and denote new cluster by d'_c
4. if there exist $c \in I$ such that $d_c \subseteq d'_c$ and for all $v \in d'_c$: $dist_{G \cup H}(v, c) \leq \log n$ then move vertices from d'_c to c
5. remove e_c that are not swallowed by clusters (that edges will be connectors in C')

let C' be a set of new clusters and unmodified clusters

so called "arrow edge"

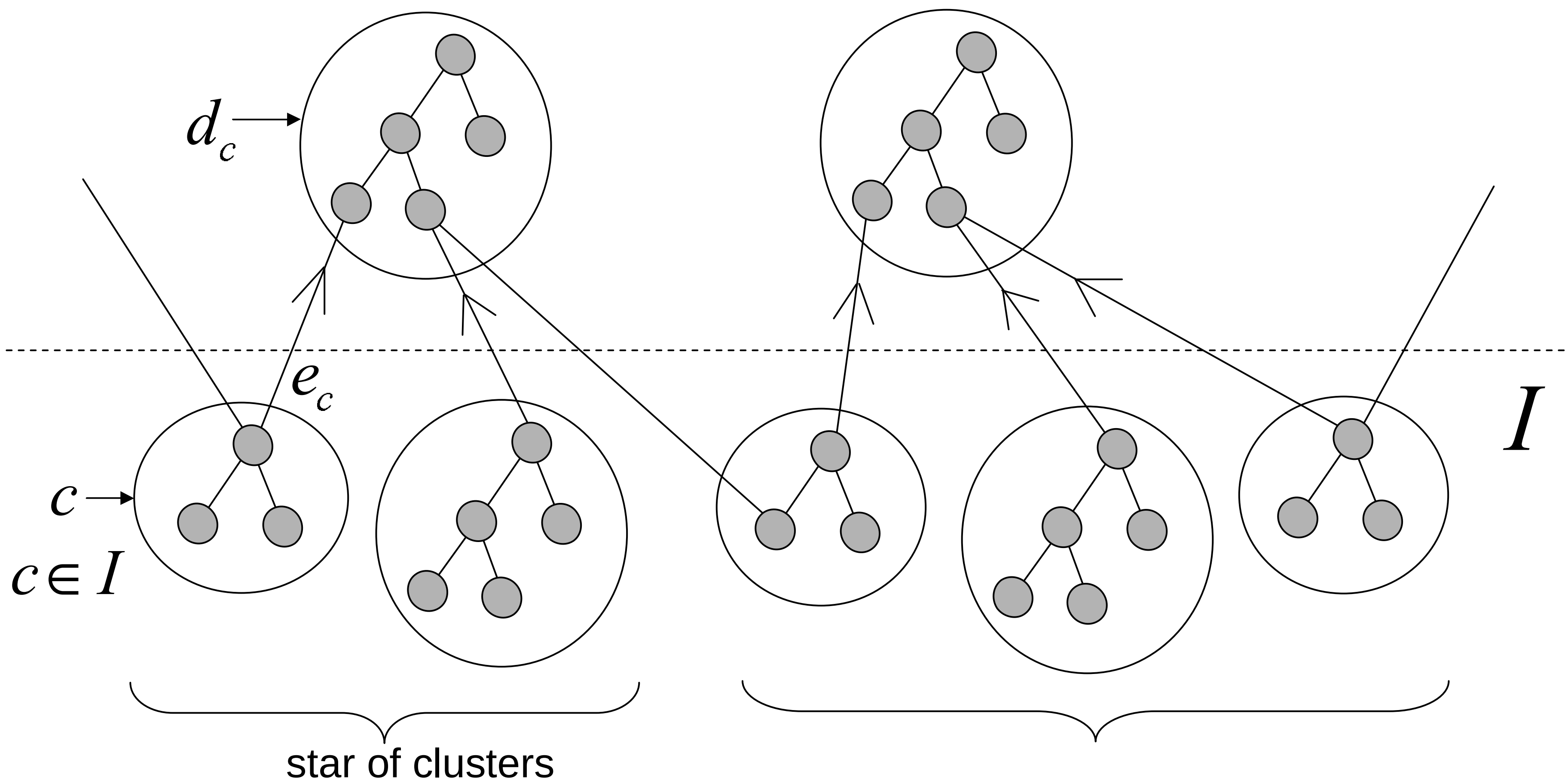
cluster c tries to join to the cluster d_c if it is possible ...

cluster d'_c tries to join to the cluster c

MWM in tree

procedure ModifyClusterSet
input: G, C, H; output: C'

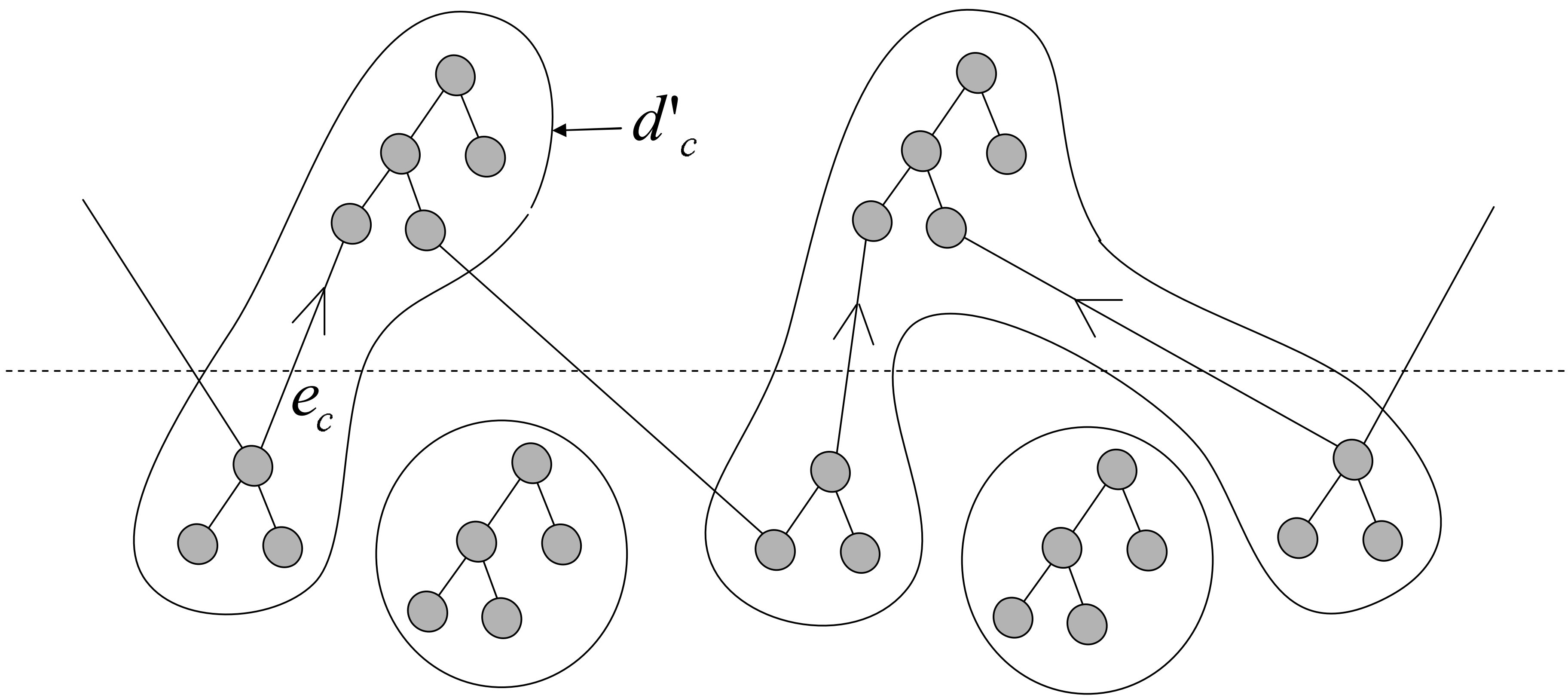
it is only a single iteration of the procedure !



MWM in tree

procedure ModifyClusterSet
input: G, C, H; output: C'

it is only a single iteration of the procedure !



MWM in tree

procedure ModifyClusterSet

input: G, C, H; output: C'

lemma

clusters of C**'** have diameter:

$\leq p + \log^2 n$, where $p := \max_{x \in C} \mathrm{diam}(x)$

$\geq \log n$ *(... simplified)*

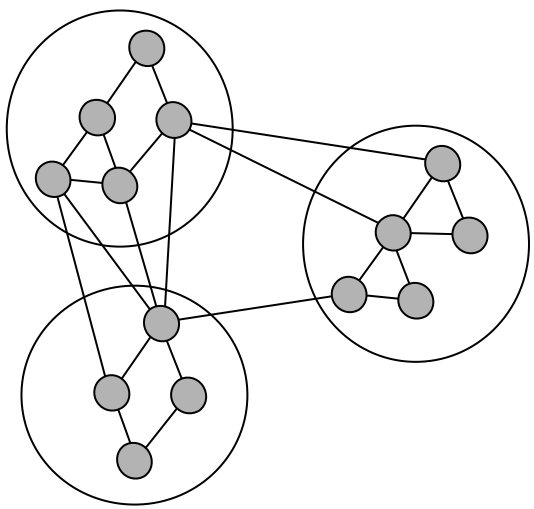
proof *(... sketch)*

- there are O(log n) iterations during which pairs of clusters join
- clusters never disjoin
- each edge of E(H) is considered as an "arrow edge"
  and swallowed by clusters or removed,
  because set I is "large" : $|I| > \Omega(n)$
  and there are O(log n) iterations

approximation
in planar graphs

approx. in planar graphs

- again, we use clusters ...
  - in planar graphs the number of connectors may be much greater then the number of clusters !!! *(we can not use the argument from tree clusters; see pages 8 and 11)*
- ”planar” clusters must have a special property:
  - total weight of connectors must be small in comparison to the weight of the input graph G



3 clusters
7 connectors

approx. in planar graphs

- the difference between clusters: in trees, and in planar graphs

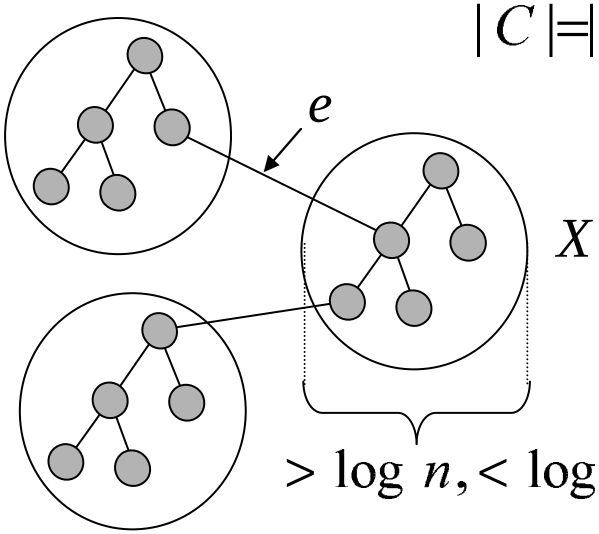
clusters in a tree

(weighted clusters)

"local" property

$$wm(X) > \frac{\log n}{20} w(e)$$

$$|C| = |L| + 1$$

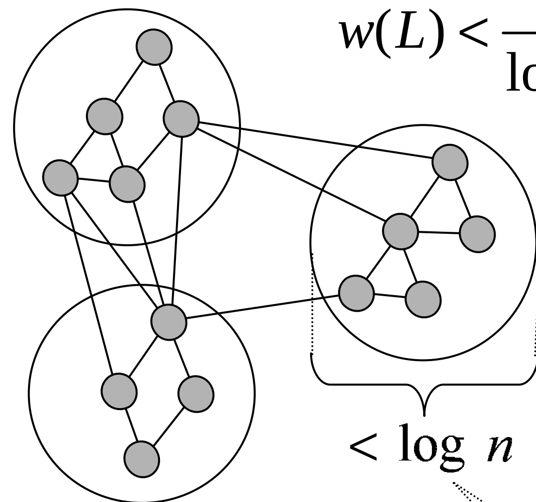


$> \log n, < \log^2 n$

clusters in a planar graph

$$w(L) < \frac{1}{\log n} w(G)$$

"global" property



$< \log n$

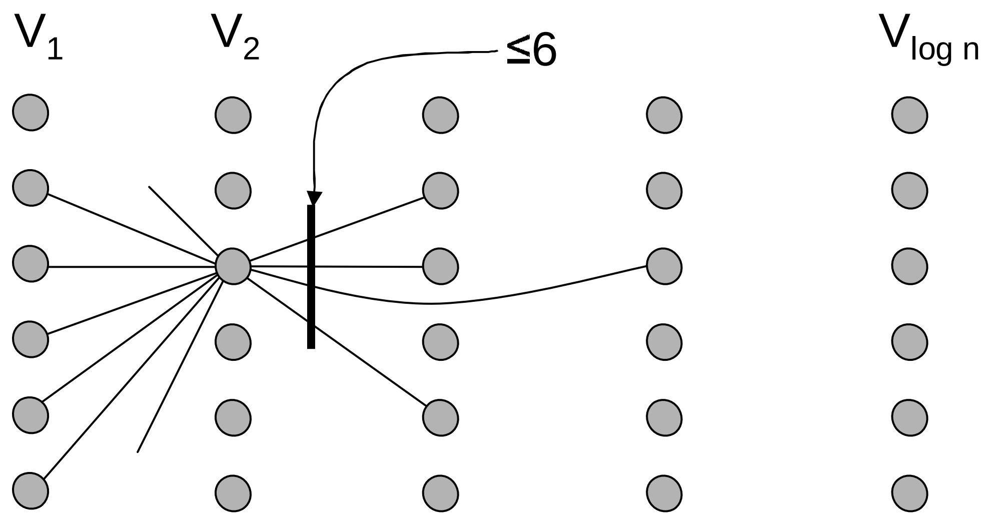
only upper bound !!!

L – set of connectors
C – set of clusters
G – input planar graph
w(S) – weight of a subgraph S

MWIS in planar graphs

- useful property of a planar graphs G:
  - it easy to compute *low-degree decomposition* in distributed model of computation ...
    def: *low-degree decomposition* is a partition of V(G) into disjoint independent sets V_1, V_2, ..., $V_{\log n}$ such that:

$$\forall_i \forall_{v \in V_i} \deg_{\bigcup_{j>i} V_j}(v) \le 6$$



MWIS in planar graphs

- *small problem:*

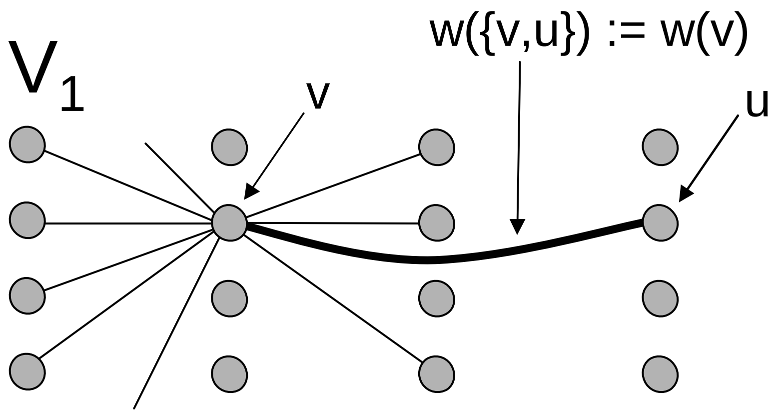
  in MWIS problem vertices (not edges) have weights !?!?

  therefore we must define:

  $w(\{v,u\}) := w(v)$ if $v \in V_i$ and $v \in V_j$ and $i<j$

  in some low-degree decomposition $V_1, V_2, ..., V_{\log n}$

  (that decomposition must be computed ...)



weights of edges satisfy:

$$\sum_{e \in E} w(e) \leq 6 \sum_{v \in V} w(v)$$

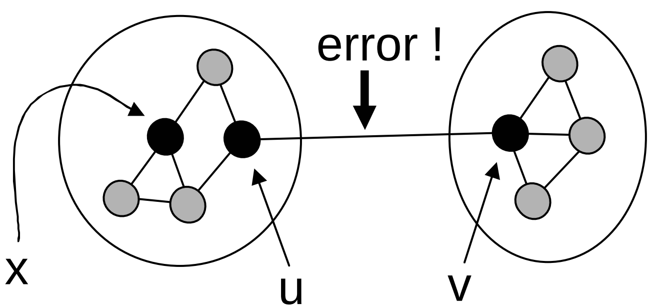
MWIS in planar graphs

- how to approximate MWIS if we have planar clusters with property: $$w(L) < \frac{1}{\log n} w(E(G))$$

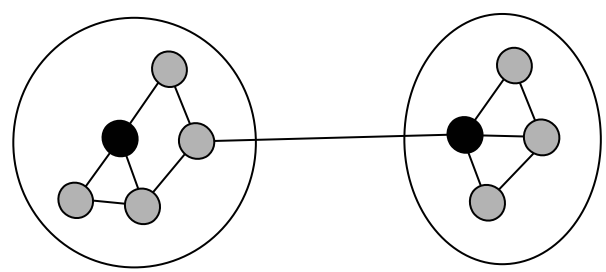
  where L is a set of connectors, and G is an input planar graph

  procedure MWIS_InTree

  1. partition set of vertices into clusters
     (do it using weights of edges !; we must compute low-deg decomp)
  2. in each cluster, in parallel, compute (optimal) MWIS ← I
     (errors on connectors may appear !)
  3. in the case of error on a connector, ← I'
     remove vertex with smaller weight



if w(u)<w(v) then
remove u form
independent set;
I={x,u,v}
I'={x,v}



MWIS in planar graphs

- how to approximate MWIS if we have planar clusters with property: $w(L) < \frac{1}{\log n} w(E(G))$
- lets denote:

  I = sum of independent sets computed in clusters in step 2

  I' = I after removing of errors on connectors (step 3)

  L = set of connectors

  wis(G) = weight of the heaviest independent set in G

- facts:

  F1 w(I) ≥ wis(G)

  F2 w(I') ≥ w(I) – w(L)

  edge-weights of connectors !

  F3 w(L) ≤ 1/log n * w(E(G)) ≤ 1/log n * 6 w(V(G))

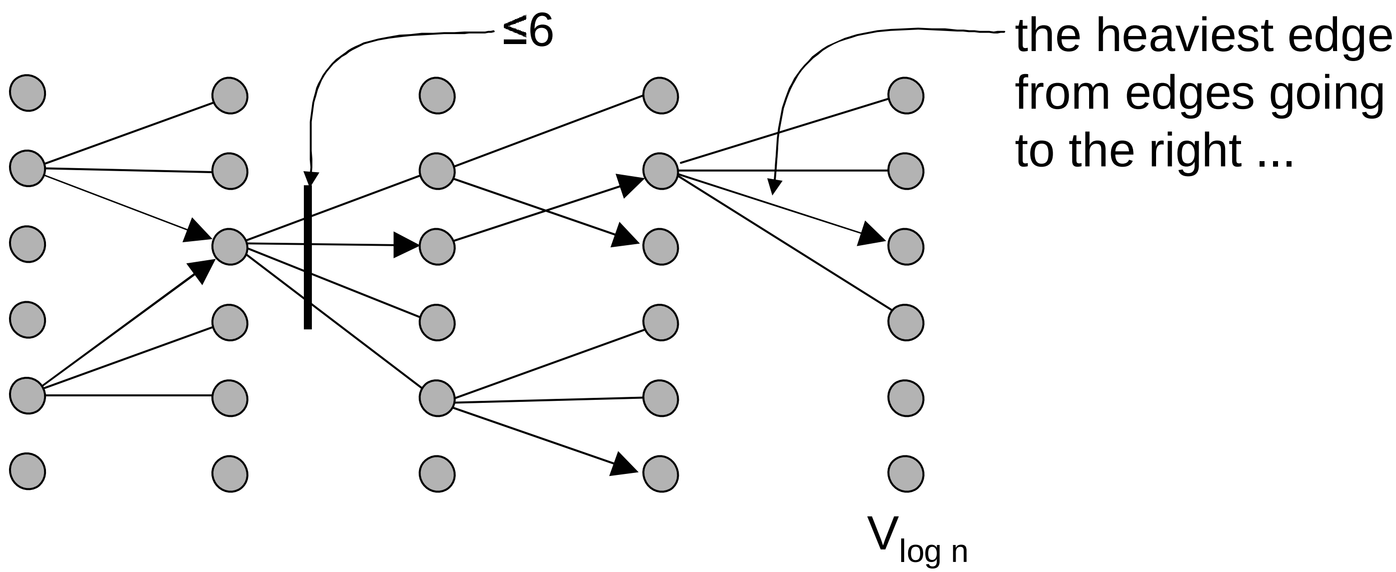
therefore:

$$\frac{w(I')}{wis(G)} \geq \frac{w(I) - \frac{6}{\log n} w(V(G))}{wis(G)} \geq 1 - \frac{6}{\log n} \frac{w(V(G))}{wis(G)}$$

≤4

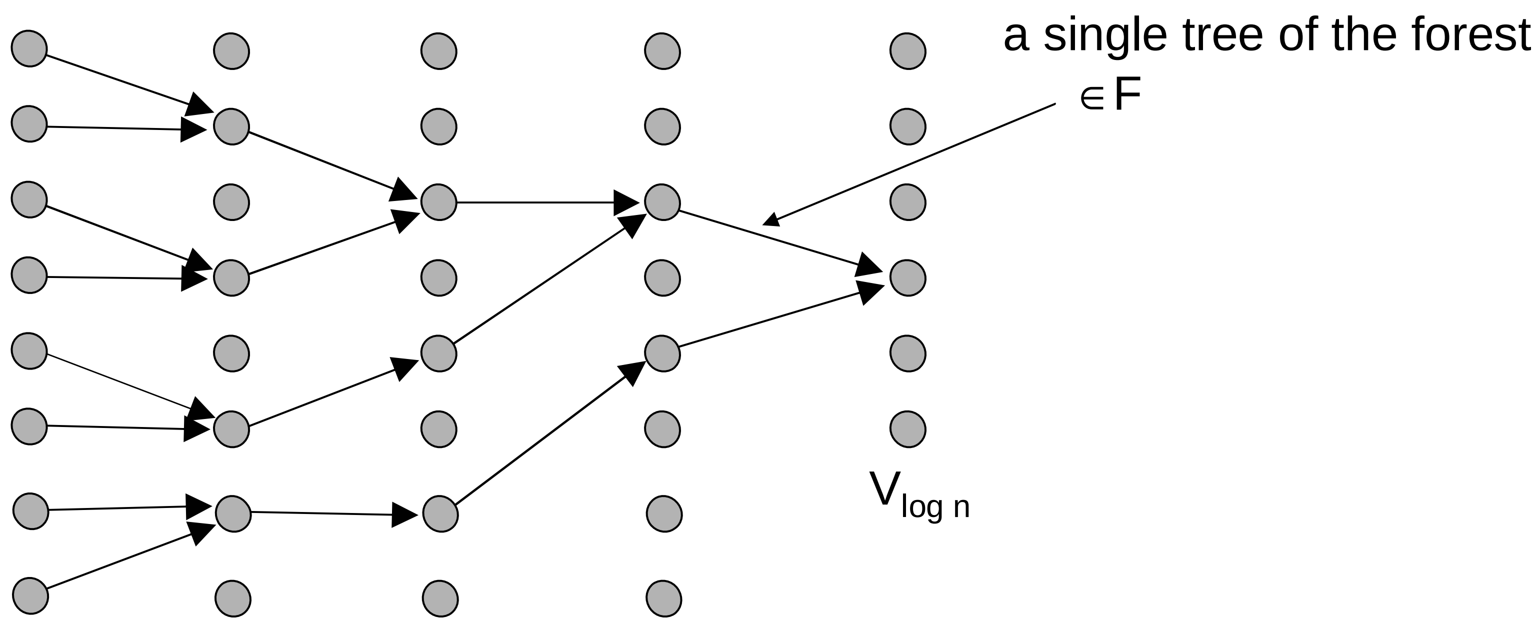
MWIS in planar graphs

- how to compute clusters in planar graphs ?
  1. compute low-degree decompositon
  2. let each vertex draw an arrow to the right side, choosing the heaviest edge ...
  3. we get forest F of oriented trees with diameter ≤ log n
  4. in each tree find the set of disjoint, spanning stars of the maximum weight (set of all stars in all trees is denoted by S)



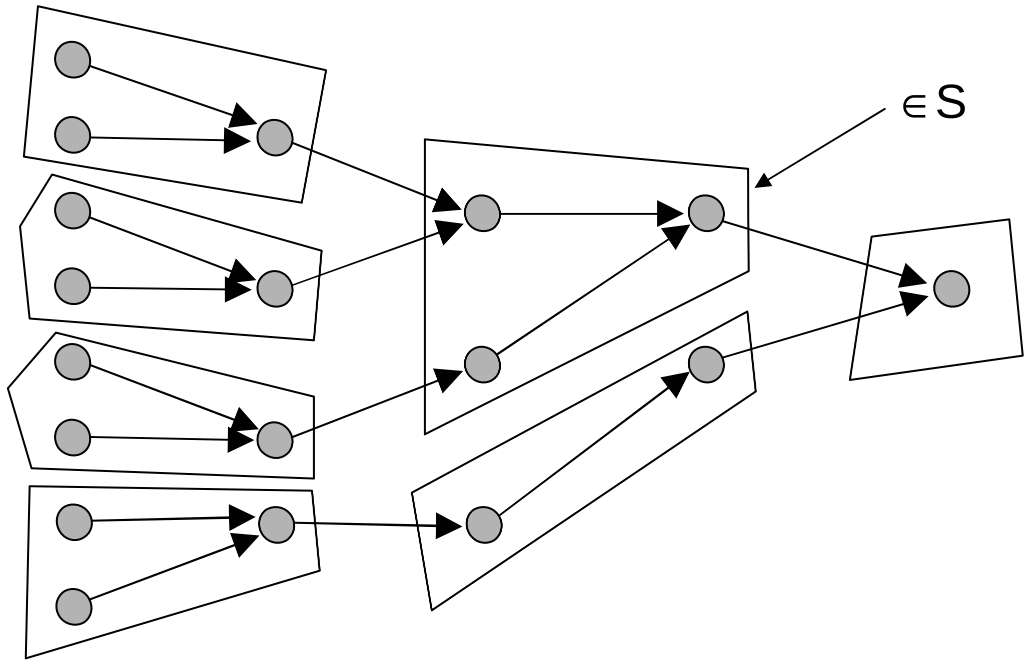
MWIS in planar graphs

- how to compute clusters in planar graphs ?
  - heavy forest in G
  - w(F) ≥ 1/6 w(G)



MWIS in planar graphs

- how to compute clusters in planar graphs ?
  - heavy spanning stars in a tree
  - w(S) ≥ ½ w(F)



MWIS in planar graphs

- how to compute clusters in planar graphs ?
  - we have clusters of diameter ≤2 (stars)

    such that

    $w(S) \geq 1/12\ w(G)$

    therefore

    $w(L) \leq (1 - 1/12)\ w(G)$

  - repeat that process of (log log n) –times

    (do it on cluster graphs !; cluster graph is also planar)

    in each iteration:

    diam := 3 diam + 2

    $w(L) := (1 – 1/12)\ w(L)$

    therefore after log log n – iterations:

    $\text{diam} \leq \log n$

    $w(L) \leq 1/\log n\ w(G)$

open problems

MWM in planar graphs

- we can not use the technique from MWIS to MWM because $w(G)/wm(G)$ is not bounded by a constant !
  (see page 25)
- **hypotesis**: it is possible to do *preprocessing* (removing of some edges: $G \to G^*$)
  in such a way that:
  1. $wm(G^*) > (1 - 1/\log n)\, wm(G)$
  2. $wm(G^*) > 1/\log n \; w(G^*)$

  where

  $wm(H)$ is a weight of the heaviest matching in H, and
  $w(H)$ is a weight of H

it is sufficient
to approximate
MWM
in planar graphs

$w(G^*)/wm(G) < \log n$

- we know how to do it in unweighted case !!!