Network Decomposition

Today, we study the concept of network decompositions, which is a fundamental tool in distributed symmetry breaking problems such as graph coloring, MIS, etc, and specially in the case of deterministic algorithms, they are the main building block in the best known solutions.

DEF: Given an undirected graph $G=(V,E)$, a (C,D)-network decomposition decomposes V into disjoint clusters $A_1, A_2, \dots$ such that

(1) Each cluster A_i has diameter at most D.

(2) The cluster-graph can be colored with C colors.

In the above, the cluster graph is the graph obtained by contracting each cluster into a new node. Furthermore, we typically want the decomposition to provide the c-coloring of the clusters (merely existence of a c-coloring is not enough).

Application:

Let's see why the above definition is useful. Take the MIS problem for instance. We can get a CD-round algorithm, using a (C,D)-network decomposition, as follows:

Pick the clusters of color 1. Solve MIS in each of these clusters, all in parallel, in D rounds, simply by learning the whole cluster. Note that different color-1 clusters do not interfere as they are not adjacent.

Now, we go to color 2. First remove all nodes in clusters of color-2 that have an MIS neighbor in color 1. Then, solve MIS in the remainder of the color-2 clusters.

Similarly, repeat this for all colors 3 to C. Each time, we spend about D+1 rounds, and there are C colors, so the algorithm takes $O(CD)$ rounds.

To see some numbers, there is a deterministic $(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})})$-network decomposition that runs in $2^{O(\sqrt{\log n})}$ rounds. This gives MIS algorithm that runs in $2^{O(\sqrt{\log n})} \times 2^{O(\sqrt{\log n})} + 2^{O(\sqrt{\log n})} = 2^{O(\sqrt{\log n})}$ rounds. To this day, this remains the fastest known deterministic MIS algorithm (compare, e.g., to the $O(\log n)$ rounds randomized algorithm of Luby).

What's known?

All graphs admit $(O(\log n), O(\log n))$-network decompositions & as argued by Linial & Saks [91], this is nearly best possible in general graphs.

Obtaining a network decomposition sequentially is relatively easy, and can be done also in $\tilde{O}(m)$ time.

This can also be obtained distributedly, but the fastest known algorithm takes $2^{O(\sqrt{\log n})}$ rounds [Panconesi, Srinivasan 92]. This algorithm first finds a $\left(2^{O(\sqrt{\log n})}, 2^{O(\sqrt{\log n})}\right)$-network decomposition & then turns it into a $(O(\log n), O(\log n))$-decomposition.

The first part is an improvement over the algorithm of Awerbuch, Goldberg, Luby & Plotkin [89] which finds a $\left(2^{\sqrt{\log n \log\log n}}, 2^{\sqrt{\log n \log\log n}}\right)$-decomp.

A (C,D)-network decomposition in T rounds for $C, D, T \in 2^{O(\sqrt{\log n \cdot \log\log n})}$:

A tool which we will use frequently is ruling sets:

DEF: Given graph $G=(V,E)$, & set $V' \subseteq V$, $S \subseteq V'$ is an (α,β)-ruling set of V' if:

(1) $\forall u,v \in S$, $dist(v,u) \geq \alpha$

(2) $\forall v \in V'$, $\exists w \in S$ s.t. $dist(v,w) \leq \beta$

Note that an MIS is a $(2,1)$-ruling set.

We first explain a deterministic algorithm that computes a $(k, k\log n)$-ruling set in $O(k\log n)$ rounds:

$(k, k\log n)$-ruling set for V' in $G=(V,E)$:

Let V_0 be the subset of nodes in V that their id has $MSB=0$ & $V_1 = V \setminus V_0$. Let $V'_0 = V' \cap V_0$ & $V'_1 = V' \cap V_1$.

Find a $(k, k\log \frac{n}{2})$-ruling set S_0 for V_0' & a $(k, k\log \frac{n}{2})$-ruling set S_1 for V_1', both in parallel.

Define the ruling set S of V' as

$$S = S_0 + \{ v \mid v \in S_1, \nexists\, u \in S_0, \text{dist}(u,v) \leq k \}$$

By definition, each two nodes of S are k hops apart, which establishes property (1). For property (2), note that nodes of V_0' are satisfied because we include all of S_0 in S. For each node v in V_1', say w was a node in S_1 that had distance at most $k\log(\frac{n}{2})$ from v. If $w \in S$, we are done. Otherwise, there is $u \in S_0$ s.t. $\text{dist}(w,u) \leq k$. Hence, $\text{dist}(v,u) \leq \text{dist}(v,w) + \text{dist}(w,u) \leq$

$$k\log(\tfrac{n}{2}) + k = k\log n.$$

As for the time complexity, we have the following recursion:

$$T(n,k) = T(\tfrac{n}{2}, k) + k$$

$$\Rightarrow T(n,k) = O(k \log n).$$

We define a (α,β)-spanning forest of set V' in graph $G=(V,E)$ to be a collection $\mathcal{F}$ of trees such that the roots of these trees form an (α,β)-ruling set of V', each node of V' is in one of the trees & each tree has radius at most β. The trees can include nodes of $V \setminus V'$. Note that given an (α,β)-ruling set, an (α,β)-spanning forest can be computed in β rounds by growing BFS trees rooted at the ruling set for β rounds, and breaking ties arbitrarily.

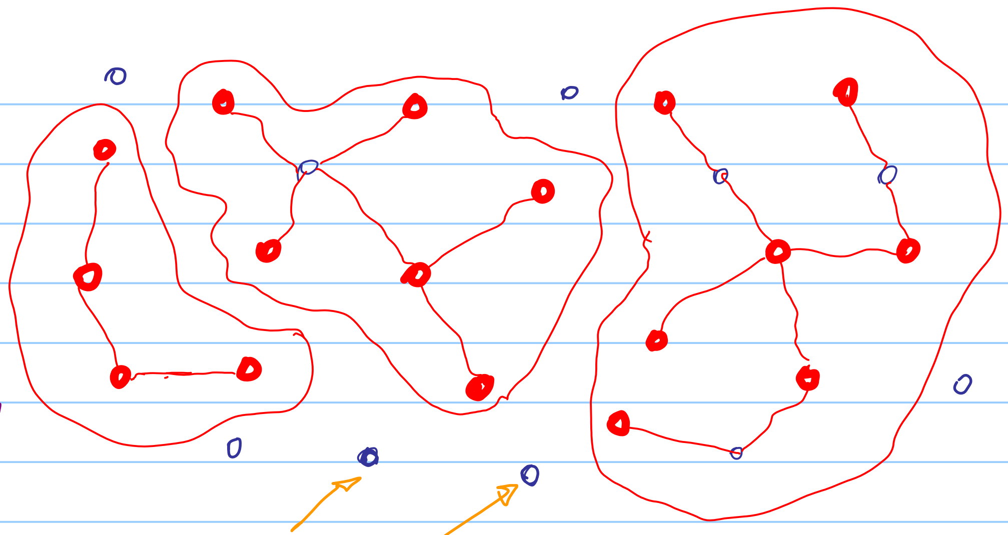
Next, we explain how to cluster nodes into a (C,D)-network decomposition for $C,D \in 2^{O(\sqrt{\log n \cdot \log\log n})}$

Later we explain how to color these clusters.

We use a parameter Δ^* during the construction. At the end, we see how to fix the value of Δ^* to obtain the best tradeoff.

Clustering: The construction runs in iterations. Let's first see the first level:

Call a node red if it's degree is $\geq \Delta^*$ & blue otherwise. Find a $(3, 3\log n)$-spanning forest of the red nodes. Blue nodes remaining out of these trees give the first-level clusters. Each of trees is contracted to a new "big" node for the next iteration of the construction.



blue nodes left outside trees
become first-level clusters
& they are removed for the rest of construction

trees are contracted to new nodes for
the next iteration

The second level of clustering is the same as the first level, but we work on the contracted graph where each red-tree of first level is a single node.

Generally, nodes of i-th level of the construction are trees of the $(i-1)$-th level. We find $(3, 3\log n)$-spanning forest for the red nodes (those with degree $\geq \Delta^*$), and nodes left out of these trees give i-th level clusters, and they are forgotten for the rest of the construction.

Lem1: Level-i nodes contain at least $(\Delta^*)^{i-1}$ many nodes of the graph $G=(V,E)$.

Lem2: Each level-i node has radius at most $(9\log n)^i$, in terms of distances in G.

From Lem1, we get that there are at most $\log_{\Delta^*} n$ many levels. Hence, using Lem2, the largest possible cluster radius is $(9 \log n)^{\log_{\Delta^*} n}$.

We will see that we can color these clusters with Δ^* many colors. Hence, the best tradeoff is when $\Delta^* = (9 \log n)^{\frac{\lg n}{\lg \Delta^*}}$

$$\Rightarrow \log \Delta^* = \frac{\log n}{\lg \Delta^*} \cdot \lg\lg n$$

$$\Rightarrow \Delta^* = 2^{\sqrt{\lg n \cdot \lg\lg n}}$$

which gives a $\left(2^{O(\sqrt{\lg n \cdot \lg\lg n})}, 2^{O(\sqrt{\lg n \cdot \lg\lg n})}\right)$-decomp.

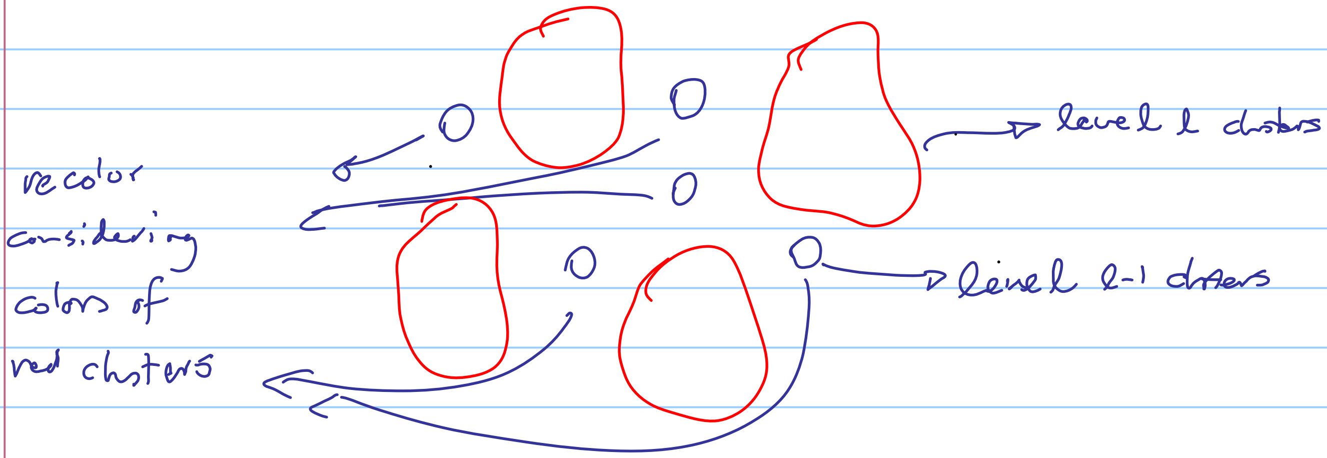
Also, note that the round complexity is $O(\log n)$ times the radius of the nodes in the highest level, which is same as $2^{O(\sqrt{\lg n \cdot \lg\lg n})}$.

Coloring: We have $\log n / \log \Delta^*$ levels of clusters. View each of these clusters as one node. Find a Δ^*-coloring tempColor of the clustergraph of each of the levels independently, and all in parallel. We use this tempColor to find the final coloring of the clusters.

For coloring, we go walk over levels $L = \frac{\log n}{\log \Delta^*}$ to 1. In level L, the coloring of clusters is the same as tempColor.

For level $\ell = L-1$, we do as follows

```
For j = 1 to $\Delta^*$
    color clusters that have tempColor = j
    considering their neighboring clusters in
    level L, and using one of colors 1 to $\Delta^*$.
```



Similarly,
```
for each level $\ell = L$ to $1$
    For each $j = 1$ to $\Delta^*$
        color clusters of level $\ell$ that have
        tempColor$=j$, considering the coloring of their
        neighboring clusters in levels $\ell$ to $L$.
```

Always, since each cluster in level ℓ has at most Δ^* neighboring clusters in levels ℓ to L, we can always find a color in 1 to Δ^*.

The process of finding the tempColor takes $O(\Delta^*) + O(\log^* n)$ rounds, in terms of nodes of the respective level, which means $2^{O(\sqrt{\lg n \cdot \lg\lg n})}$ rounds. The process of coloring over all levels $\ell = L$ to 1 also takes similar asymptotic time.